DEEP LEARNING FOR BIOMEDICINE APPLICATIONS

Disusun oleh:
I Gede Susrama Mas Diyasa
Adhi Dharma Wibawa
Khairul Anam
Anindita Septiarini
Yessy Andriani Fauziah

PT GHANI PRESS GROUP

DEEP LEARNING FOR BIOMEDICINE APPLICATIONS

Penulis:
I Gede Susrama Mas Diyasa, Adhi Dharma Wibawa, Khairul Anam, Anindita Septiarini, Yessy Andriani Fauziah

Desain Cover:
Umar Khasan

Layouter:
Rizki Baehtiar Afandi

Editor Naskah:
Prof. Dr. Basuki Rahmat, S.Si, MT

Anggota IKAPI: No. 468/JTI/2026
Diterbitkan oleh: PT GHANI PRESS GROUP

Perpustakaan Nasional RI: Katalog Dalam Terbitan (KDT)

Diyasa dkk.,
DEEP LEARNING FOR BIOMEDICINE APPLICATIONS 2026 / penulis, Diyasa dkk., Lamongan: PT. Ghani Press Group, 2026.
V + 469 hlm; 15,5 x 23 cm

ISBN 978-634-97317-4-4
1. Kedokteran - Kecerdasan buatan I. Judul
610.285 63 [23]

Sanksi Pelanggaran Pasal 113
Undang-Undang No. 28 Tahun 2014 Tentang Hak Cipta

a) Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp 100.000.000 (seratus juta rupiah).
b) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp 500.000.000,00 (lima ratus juta rupiah).
c) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp 1.000.000.000,00 (satu miliar rupiah).
d) Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp 4.000.000.000,00 (empat miliar rupiah).

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat, karunia, dan petunjuk-Nya sehingga buku ajar berjudul "*Deep Learning for Biomedical Applications"* ini dapat diselesaikan dengan baik.

Perkembangan kecerdasan artifisial, khususnya *deep learning*, telah mengubah praktik di bidang biomedis secara mendasar. Kemampuan model pembelajaran mendalam dalam mengenali pola kompleks pada citra medis, sinyal biologis, maupun data klinis berskala besar membuka peluang baru bagi peningkatan akurasi diagnosis, percepatan proses skrining, serta pengembangan sistem pendukung keputusan klinis. Di sisi lain, penerapan teknologi ini pada domain kesehatan menuntut pemahaman yang utuh, tidak hanya pada aspek algoritmik, tetapi juga pada karakteristik data biomedis, kaidah validasi, serta pertimbangan etis dan keselamatan pasien.

Buku ini disusun untuk menjembatani kebutuhan tersebut. Pembahasan dimulai dari konsep dasar *deep learning* dan arsitektur jaringan saraf tiruan, dilanjutkan dengan metode pengolahan serta analisis data biomedis, penerapan pada bidang pencitraan dan diagnostik, hingga contoh studi kasus nyata pemanfaatan kecerdasan artifisial dalam layanan kesehatan. Penyajian materi diupayakan berimbang antara landasan teoretis dan penerapan praktis, sehingga dapat dimanfaatkan oleh mahasiswa program sarjana maupun pascasarjana, dosen, peneliti, serta praktisi yang menaruh minat pada bidang *biomedical engineering* dan informatika kesehatan.

Penulis menyadari sepenuhnya bahwa penyusunan buku ini tidak terlepas dari dukungan berbagai pihak. Untuk itu, penulis menyampaikan penghargaan dan ucapan terima kasih yang sebesar-besarnya kepada Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) Universitas Pembangunan Nasional "Veteran" Jawa Timur atas dukungan pendanaan melalui Hibah Jatim Melaju

Tahun 2026. Dukungan tersebut memiliki peran yang sangat penting dalam terwujudnya buku referensi ini, tidak hanya dalam memungkinkan penyelesaian naskah, tetapi juga dalam memperkuat ekosistem riset, penulisan, dan publikasi ilmiah di lingkungan perguruan tinggi.

Ucapan terima kasih juga penulis sampaikan kepada Pimpinan Universitas Pembangunan Nasional "Veteran" Jawa Timur beserta seluruh jajarannya atas dukungan kebijakan dan fasilitasi yang diberikan selama proses penulisan berlangsung. Penghargaan yang tulus penulis sampaikan pula kepada rekan-rekan sejawat dan sivitas akademika dari institusi asal masing-masing penulis atas diskusi ilmiah, masukan, dan kolaborasi yang telah memperkaya isi buku ini, serta kepada para mahasiswa bimbingan yang turut membantu penyiapan materi, pelaksanaan eksperimen, dan penyajian visualisasi hasil.

Secara khusus, penulis mengucapkan terima kasih kepada keluarga tercinta atas doa, pengertian, dan dukungan moral yang tiada henti sepanjang proses penyusunan buku ini. Kepada semua pihak yang tidak dapat kami sebutkan satu per satu, namun telah memberikan kontribusi baik secara langsung maupun tidak langsung, penulis menyampaikan penghargaan dan terima kasih yang setulus-tulusnya.

Semoga buku ini memberikan manfaat nyata bagi pengembangan ilmu pengetahuan dan berkontribusi pada kemajuan layanan kesehatan melalui pemanfaatan kecerdasan artifisial.

Surabaya, September 2026

Penulis

DAFTAR ISI

KATA PENGANTAR v

DAFTAR ISI vii

BAB 1
KLASIFIKASI *PNEUMONIA* PADA CITRA *CHEST X-RAY* MENGGUNAKAN *RESNET-50*, *EFFICIENTNET-B0*, DAN *EXPLAINABLE AI* 1

BAB 2
SEGMENTASI TUMOR OTAK PADA CITRA MRI MENGGUNAKAN *U-NET* DAN ESTIMASI TINGKAT KEPARAHAN BERDASARKAN LUAS AREA TUMOR 33

BAB 3
KLASIFIKASI MORFOLOGI *SPERMATOZOA* PADA CITRA MIKROSKOPIS MENGGUNAKAN OPERASI MORFOLOGI DAN *EFFICIENTNET-V2* 63

BAB 4
KLASIFIKASI DELAPAN KELAS PENYAKIT RETINA PADA CITRA OCT: PERBANDINGAN *RESNET-50* DAN *EFFICIENTNET-B0* DENGAN *GRAD-CAM++* 93

BAB 5
DETEKSI DAN ESTIMASI KEPARAHAN STROKE ISKEMIK PADA CITRA MRI OTAK: *PIPELINE* DUA TAHAP *DENSENET121-CBAM* DAN *NVAUTONET* 121

BAB 6
PREDIKSI KOMPLIKASI MEDIS PASIEN HEMODIALISIS MENGGUNAKAN *XGBOOST* DENGAN OPTIMALISASI *ARTIFICIAL BEE COLONY* 179

BAB 7
KLASIFIKASI EMPAT KELAS TUMOR OTAK PADA CITRA MRI: PERBANDINGAN CNN, *VISION TRANSFORMER*, DAN MODEL HIBRIDA CNN-ViT 233

BAB 8
KLASIFIKASI JENIS KELAMIN PADA CITRA PANORAMIK GIGI MENGGUNAKAN *DEEP LEARNING* BERBASIS *DENSENET169* 283

BAB 9
SISTEM ANALISIS PROPORSI MORFOLOGI SEL DARAH MERAH PADA APUSAN DARAH TEPI MENGGUNAKAN *MOBILENETV2* UNTUK PERINGATAN DINI TALASEMIA..... 351

BAB 10
KLASIFIKASI PENYAKIT PARU-PARU PADA CITRA *CHEST X-RAY* MENGGUNAKAN *FINE TUNING VGG16* DAN *EXPLAINABLE ARTIFICIAL INTELLIGENCE* 421

BIODATA PENULIS..... 467

BAB 1
KLASIFIKASI *PNEUMONIA* PADA CITRA *CHEST X-RAY* MENGGUNAKAN *RESNET-50*, *EFFICIENTNET-B0*, DAN *EXPLAINABLE AI*

Setelah mempelajari bab ini, pembaca diharapkan mampu:

a. Menjelaskan alasan penggunaan *Convolutional Neural Network* pada permasalahan klasifikasi citra medis.
b. Menguraikan prinsip kerja arsitektur *ResNet-50* dan *EfficientNet-B0* beserta perbedaan filosofi perancangannya.
c. Menerapkan *transfer learning*, *preprocessing*, dan augmentasi data pada citra *Chest X-ray*.
d. Melakukan penalaan *hyperparameter* serta menganalisis pengaruh *learning rate* terhadap performa model.
e. Mengevaluasi model menggunakan *accuracy*, *precision*, *recall*, *F1-score*, *ROC-AUC*, dan *confusion matrix*, serta memahami mengapa *recall* bersifat kritis pada konteks medis.
f. Menginterpretasikan keputusan model menggunakan *Grad-CAM* dan mengenali gejala *shortcut learning*.

➢ Pendahuluan

Pneumonia merupakan penyakit infeksi pada paru-paru yang hingga kini masih menjadi persoalan kesehatan serius di banyak negara, termasuk Indonesia. Peradangan pada alveoli yang ditimbulkannya mengganggu proses pertukaran oksigen di dalam tubuh sehingga keterlambatan penanganan dapat berakibat fatal. *World Health Organization* mencatat sekitar 740.180 kematian anak berusia di bawah lima tahun akibat *pneumonia* pada tahun 2019, atau sekitar 14% dari seluruh kematian pada kelompok usia tersebut. Angka tersebut menempatkan deteksi dini sebagai kunci utama penurunan angka kematian, dan pemeriksaan citra *Chest X-ray* menjadi pilihan yang paling banyak digunakan karena biayanya relatif terjangkau serta ketersediaannya yang luas di berbagai fasilitas kesehatan.

Persoalannya, interpretasi citra *Chest X-ray* masih dilakukan secara manual oleh dokter radiologi. Proses ini memerlukan waktu dan sangat bergantung pada pengalaman serta ketelitian pembacanya. Ketika jumlah pasien yang harus diperiksa meningkat, beban kerja yang tinggi membuka peluang terjadinya *human error* yang berpotensi memengaruhi hasil diagnosis. Kondisi inilah yang mendorong kebutuhan akan sistem pendukung diagnosis otomatis yang mampu bekerja cepat dan konsisten, bukan untuk menggantikan peran klinisi, melainkan untuk membantu proses penapisan awal dan menandai kasus yang memerlukan perhatian lebih.

Perkembangan *deep learning* menawarkan jawaban atas kebutuhan tersebut. Di antara berbagai pendekatan yang tersedia, *Convolutional Neural Network* (CNN) paling banyak digunakan untuk pengolahan citra karena kemampuannya mengekstraksi fitur secara otomatis tanpa perancangan fitur secara manual. Dua arsitektur CNN yang menjadi fokus bab ini adalah *ResNet-50* dan *EfficientNet-B0*. *ResNet-50* mengandalkan *residual connection* yang memungkinkan jaringan dibangun jauh lebih dalam tanpa terjebak pada masalah *vanishing gradient*, sedangkan *EfficientNet-B0* menerapkan *compound scaling* sehingga jauh lebih hemat parameter dan sumber daya komputasi. Keduanya mewakili dua filosofi perancangan yang berbeda, sehingga membandingkan keduanya secara langsung pada satu kasus medis yang sama memberikan gambaran nyata mengenai konsekuensi dari setiap pilihan arsitektur.

Akurasi yang tinggi, bagaimanapun, tidak dengan sendirinya membuat sebuah model layak digunakan di lingkungan klinis. Model *deep learning* kerap disebut sebagai *black box* karena sulit menjelaskan dasar dari prediksi yang dihasilkannya, sedangkan dalam praktik medis setiap keputusan menuntut alasan yang dapat ditelusuri. Pendekatan *Explainable Artificial Intelligence* (XAI) hadir untuk menutup celah tersebut, dan salah satu metodenya yang paling banyak digunakan adalah *Gradient-weighted Class Activation Mapping* (*Grad-CAM*). Metode ini memvisualisasikan area citra yang paling berkontribusi terhadap keputusan model dalam bentuk *heatmap*, sehingga dapat diperiksa apakah model benar-benar memperhatikan area paru-paru atau justru bertumpu pada informasi yang tidak relevan secara klinis.

Bab ini menuntun pembaca membangun sistem klasifikasi *pneumonia* dari citra *Chest X-ray* secara utuh, mulai dari penyiapan dan augmentasi data, penerapan *transfer learning* pada *ResNet-50* dan *EfficientNet-B0*, penalaan *hyperparameter* untuk memperoleh konfigurasi terbaik, evaluasi performa menggunakan berbagai metrik, hingga interpretasi hasil dengan *Grad-CAM*. Pembahasan dibatasi pada klasifikasi biner antara kelas NORMAL dan *PNEUMONIA* menggunakan dataset publik, tanpa validasi eksternal maupun uji klinis. Setelah menuntaskan bab ini, pembaca diharapkan tidak hanya mampu mereproduksi seluruh alur kerjanya, tetapi juga mampu menilai secara kritis apakah sebuah model layak dipercaya keterampilan yang akan terus digunakan pada bab-bab berikutnya.

➢ Pengantar Teori

Bagian ini menguraikan landasan teori yang diperlukan untuk memahami seluruh tahapan pada bab ini. Pembahasan dimulai dari operasi dasar penyusun CNN, dilanjutkan dengan dua arsitektur yang digunakan, konsep *transfer learning*, proses optimasi, metode interpretasi *Grad-CAM*, dan diakhiri dengan metrik evaluasi.

Convolutional Neural Network

Convolutional Neural Network adalah jenis jaringan saraf tiruan yang dirancang khusus untuk data berstruktur grid seperti citra. Berbeda dengan jaringan terhubung penuh yang memperlakukan setiap piksel sebagai fitur independen, CNN memanfaatkan dua asumsi penting pada citra, yaitu *lokalitas* (piksel yang berdekatan cenderung saling berkaitan) dan *invariansi translasi* (pola yang sama dapat muncul di posisi mana pun). Kedua asumsi tersebut diwujudkan melalui operasi konvolusi, yaitu penggeseran kernel berukuran kecil ke seluruh permukaan citra untuk menghasilkan *feature map*. Secara matematis, operasi konvolusi dua dimensi dinyatakan sebagai berikut.

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) \cdot K(m,n) \quad (1.1)$$

Dengan I menyatakan citra masukan, K menyatakan kernel atau filter, serta i dan j menyatakan posisi piksel pada *feature map* keluaran. Nilai bobot pada kernel tidak ditentukan secara manual, melainkan dipelajari selama proses pelatihan. Pada lapisan awal, kernel umumnya belajar mengenali pola sederhana seperti tepi dan gradien intensitas, sedangkan pada lapisan yang lebih dalam kernel mampu menangkap pola yang lebih kompleks seperti tekstur jaringan paru dan pola konsolidasi yang menjadi ciri *pneumonia*.

Keluaran konvolusi selanjutnya dilewatkan pada fungsi aktivasi nonlinier. Fungsi yang paling umum digunakan adalah *Rectified Linear Unit* (ReLU) karena sederhana secara komputasi dan tidak menyebabkan gradien menghilang pada nilai masukan positif.

$$f(x) = max(0, x) \quad (1.2)$$

Setelah aktivasi, *feature map* biasanya diperkecil dimensinya melalui operasi *pooling*. Operasi ini mengambil nilai maksimum atau rata-rata pada setiap jendela kecil sehingga jumlah parameter berkurang, komputasi menjadi lebih ringan, dan representasi menjadi lebih tahan terhadap pergeseran kecil pada objek. Rangkaian konvolusi, aktivasi, dan *pooling* diulang beberapa kali hingga menghasilkan representasi fitur yang padat, yang kemudian diteruskan ke lapisan klasifikasi untuk menghasilkan prediksi akhir.

Arsitektur *ResNet-50*

Menambah kedalaman jaringan secara teoretis meningkatkan kemampuan representasi, tetapi pada praktiknya jaringan yang sangat dalam justru mengalami penurunan akurasi. Penyebab utamanya adalah *vanishing gradient*, yaitu mengecilnya nilai gradien ketika dipropagasikan mundur melewati banyak lapisan sehingga bobot pada lapisan awal nyaris tidak diperbarui. *ResNet* mengatasi persoalan ini melalui *residual connection*, yaitu jalur pintas yang menambahkan masukan sebuah blok langsung ke keluarannya.

$$y = F(x, W_i) + x \quad (1.3)$$

Dengan x menyatakan masukan blok, F menyatakan fungsi residual yang dipelajari, dan y menyatakan keluaran blok. Dengan struktur ini, jaringan tidak lagi dipaksa mempelajari pemetaan

lengkap, melainkan cukup mempelajari selisih atau *residual* terhadap masukan. Apabila lapisan tambahan ternyata tidak bermanfaat, jaringan dapat dengan mudah mendorong F menuju nol sehingga blok tersebut berperilaku sebagai pemetaan identitas. Jalur pintas juga menyediakan lintasan langsung bagi gradien untuk mengalir ke lapisan awal.

ResNet-50 tersusun atas 50 lapisan berbobot yang dibangun dari blok *bottleneck*, yaitu kombinasi konvolusi 1×1, 3×3, dan 1×1. Konvolusi 1×1 pertama menurunkan jumlah kanal, konvolusi 3×3 melakukan ekstraksi fitur spasial, dan konvolusi 1×1 terakhir mengembalikan jumlah kanal ke dimensi semula. Rancangan ini menjaga biaya komputasi tetap terkendali meskipun jaringan sangat dalam, dan menghasilkan sekitar 25,6 juta parameter.

Arsitektur *EfficientNet-B0*

Pendekatan konvensional dalam memperbesar CNN dilakukan dengan menambah salah satu dimensi saja, yaitu memperdalam jaringan, memperlebar jumlah kanal, atau memperbesar resolusi citra masukan. *EfficientNet* memperkenalkan gagasan bahwa ketiga dimensi tersebut saling bergantung dan sebaiknya diperbesar secara serentak dengan proporsi yang seimbang. Gagasan ini dikenal sebagai *compound scaling* dan dirumuskan sebagai berikut.

$$d = \alpha^{\varphi},\ w = \beta^{\varphi},\ r = \gamma^{\varphi} \qquad (1.4)$$

Dengan d, w, dan r berturut-turut menyatakan faktor pengali untuk kedalaman, lebar, dan resolusi; φ adalah koefisien gabungan yang mengatur besarnya sumber daya tambahan yang tersedia; sedangkan α, β, dan γ adalah konstanta hasil pencarian yang memenuhi kendala $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ sehingga penambahan satu satuan φ kira-kira melipatgandakan biaya komputasi.

EfficientNet-B0 merupakan model dasar pada keluarga ini dan disusun dari blok MBConv (*Mobile Inverted Bottleneck Convolution*) yang memanfaatkan *depthwise separable convolution* serta modul *squeeze-and-excitation* untuk memberi bobot berbeda pada setiap kanal. Kombinasi tersebut membuat *EfficientNet-B0* hanya memerlukan sekitar 5,3 juta parameter, kurang dari seperempat parameter *ResNet-50*, namun tetap mampu memberikan akurasi yang kompetitif. Karakteristik ini menjadikannya menarik untuk penerapan pada perangkat dengan sumber daya terbatas, misalnya perangkat penapisan di fasilitas kesehatan primer.

Transfer Learning

Melatih CNN dari awal memerlukan data dalam jumlah sangat besar, sedangkan citra medis berlabel umumnya terbatas karena proses pelabelannya menuntut keahlian klinis. *Transfer learning* menjadi jalan keluar dengan memanfaatkan model yang telah dilatih pada dataset berskala besar seperti *ImageNet*, lalu menyesuaikannya untuk permasalahan baru. Asumsi yang mendasarinya adalah bahwa fitur tingkat rendah seperti tepi, sudut, dan tekstur bersifat umum dan tetap berguna meskipun domain citranya berbeda jauh.

Terdapat dua strategi utama. Pada *feature extraction*, seluruh bobot model dasar dibekukan dan hanya lapisan klasifikasi baru yang dilatih; strategi ini cepat dan cocok ketika data sangat sedikit. Pada *fine-tuning*, sebagian lapisan akhir model dasar ikut dilatih dengan *learning rate* kecil sehingga model dapat menyesuaikan diri dengan karakteristik khusus citra medis. Pada bab ini digunakan pendekatan *transfer learning* dengan bobot *ImageNet* sebagai titik awal bagi kedua arsitektur.

Fungsi Aktivasi Keluaran dan Fungsi Kerugian

Karena permasalahan yang dihadapi bersifat biner, lapisan keluaran hanya terdiri atas satu neuron dengan fungsi aktivasi sigmoid yang memetakan keluaran jaringan ke rentang nol sampai satu sehingga dapat ditafsirkan sebagai probabilitas kelas *PNEUMONIA*.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \qquad (1.5)$$

Selisih antara probabilitas prediksi dan label sebenarnya diukur menggunakan fungsi kerugian *binary cross-entropy*, yang memberikan penalti besar ketika model sangat yakin namun keliru.

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \, log(\hat{y}_i) + (1 - y_i) \, log(1 - \hat{y}_i)] \qquad (1.6)$$

Dengan N menyatakan jumlah sampel, y menyatakan label sebenarnya bernilai 0 untuk NORMAL dan 1 untuk *PNEUMONIA*, serta ŷ menyatakan probabilitas hasil prediksi model.

Optimasi dan Peran *Learning Rate*

Proses pelatihan pada dasarnya adalah pencarian nilai bobot yang meminimumkan fungsi kerugian. Pembaruan bobot dilakukan berdasarkan arah negatif gradien fungsi kerugian terhadap bobot tersebut.

$$\theta \leftarrow \theta - \eta \frac{\partial L}{\partial \theta} \qquad (1.7)$$

Dengan θ menyatakan parameter model dan η menyatakan *learning rate*. Nilai η menentukan besar langkah pada setiap pembaruan dan merupakan *hyperparameter* yang paling berpengaruh terhadap keberhasilan pelatihan. Nilai yang terlalu besar membuat proses optimasi melompati titik minimum sehingga pelatihan berosilasi atau bahkan divergen, sedangkan nilai yang terlalu kecil membuat pelatihan berjalan sangat lambat dan berisiko terjebak pada minimum lokal. Fakta bahwa kedua ujung ekstrem tersebut sama-sama merugikan menjadi alasan utama dilakukannya penalaan *learning rate* secara sistematis pada bab ini.

Explainable AI dan *Grad-CAM*

Explainable Artificial Intelligence merupakan sekumpulan pendekatan yang bertujuan membuat keputusan model dapat dipahami manusia. Pada domain medis, kebutuhan ini bukan sekadar pelengkap, melainkan syarat kelayakan: seorang klinisi tidak dapat mempertanggungjawabkan diagnosis yang didasarkan pada keluaran yang tidak dapat dijelaskan. *Grad-CAM* merupakan salah satu metode XAI yang paling banyak digunakan pada CNN karena tidak memerlukan perubahan arsitektur maupun pelatihan ulang.

Grad-CAM bekerja dengan menghitung gradien skor kelas terhadap *feature map* pada lapisan konvolusi terakhir. Rata-rata gradien pada setiap kanal digunakan sebagai bobot yang menyatakan seberapa penting kanal tersebut bagi kelas yang diprediksi.

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \qquad (1.8)$$

Bobot tersebut kemudian dikalikan dengan *feature map* yang bersesuaian dan dijumlahkan, lalu dilewatkan pada fungsi ReLU agar hanya kontribusi positif terhadap kelas yang bersangkutan yang dipertahankan.

$$L^c_{Grad-CAM} = ReLU(\sum_k \alpha^c_k A^k) \qquad (1.9)$$

Peta hasil perhitungan tersebut berukuran sama dengan *feature map* terakhir, yaitu jauh lebih kecil daripada citra asli, sehingga perlu diperbesar kembali ke ukuran 224 × 224 dan ditumpangkan pada citra masukan dalam bentuk *heatmap*. Area berwarna merah menandakan kontribusi paling besar terhadap keputusan model, sedangkan area berwarna biru menandakan kontribusi yang kecil.

Metrik Evaluasi

Seluruh metrik evaluasi pada klasifikasi biner diturunkan dari *confusion matrix*, yaitu tabel silang antara label sebenarnya dan label hasil prediksi sebagaimana ditunjukkan pada Tabel 1.1.

Tabel 1.1 Struktur *Confusion Matrix* pada Klasifikasi Biner

	Prediksi NORMAL	**Prediksi *PNEUMONIA***
Aktual NORMAL	True Negative (TN)	False Positive (FP)
Aktual *PNEUMONIA*	False Negative (FN)	True Positive (TP)

Accuracy menyatakan proporsi seluruh prediksi yang benar terhadap total data. Metrik ini mudah dipahami, tetapi dapat menyesatkan apabila distribusi kelas tidak seimbang.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (1.10)$$

Precision menyatakan seberapa besar proporsi kasus yang diprediksi *pneumonia* yang memang benar-benar *pneumonia*. Nilai yang tinggi berarti model jarang menghasilkan alarm palsu.

$$Precision = \frac{TP}{TP + FP} \quad (1.11)$$

Recall atau sensitivitas menyatakan seberapa besar proporsi kasus *pneumonia* yang berhasil ditemukan oleh model. Pada konteks penapisan penyakit, metrik inilah yang paling kritis: sebuah *false positive* hanya berujung pada pemeriksaan lanjutan yang tidak perlu, sedangkan sebuah *false negative* berarti pasien yang benar-benar sakit dipulangkan tanpa penanganan.

$$Recall = \frac{TP}{TP + FN} \quad (1.12)$$

F1-score merupakan rata-rata harmonik antara *precision* dan *recall* sehingga memberikan gambaran keseimbangan di antara keduanya. Nilai F1 hanya akan tinggi apabila kedua komponennya sama-sama tinggi.

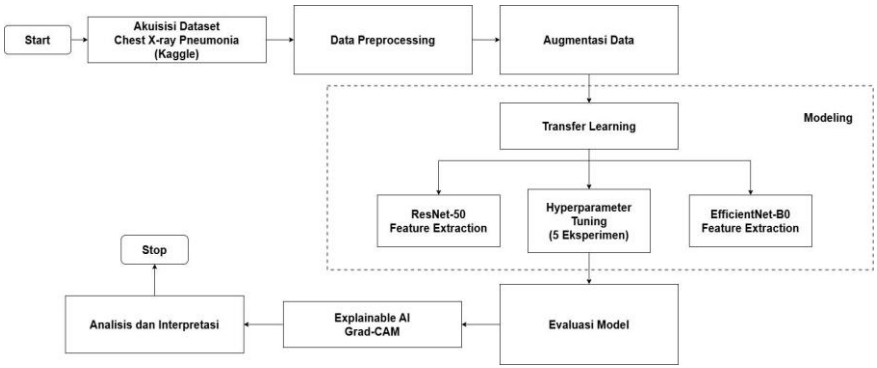
$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (1.13)$$

Metrik terakhir yang digunakan adalah *ROC-AUC*, yaitu luas area di bawah kurva *Receiver Operating Characteristic* yang memetakan hubungan antara *true positive rate* dan *false positive rate* pada berbagai nilai ambang. Berbeda dengan metrik sebelumnya yang bergantung pada satu ambang tertentu, *ROC-AUC* menilai kemampuan pemisahan model secara menyeluruh. Nilai 1,0 menandakan pemisahan sempurna, sedangkan nilai 0,5 setara dengan tebakan acak.

➢ Metodologi

Alur Metodologi

Keseluruhan proses yang dibahas pada bab ini disusun sebagai satu alur kerja yang berurutan, dimulai dari akuisisi data hingga interpretasi hasil. Alur tersebut ditunjukkan pada Gambar 1.1.



Gambar 1.1 Arsitektur dan Alur Metodologi Klasifikasi *Pneumonia*

Proses diawali dengan akuisisi dataset *Chest X-ray Pneumonia* dari Kaggle. Dataset kemudian melalui tahap *preprocessing* yang meliputi pengubahan ukuran citra, normalisasi nilai piksel, pelabelan

kelas, serta pembagian data menjadi subset pelatihan, validasi, dan pengujian. Data pelatihan selanjutnya diperkaya melalui augmentasi. Tahap pemodelan menggunakan pendekatan *transfer learning* dengan *ResNet-50* dan *EfficientNet-B0* sebagai pengekstraksi fitur, disertai lima skenario penalaan *hyperparameter*. Model yang dihasilkan dievaluasi menggunakan berbagai metrik, dan model terbaik dianalisis lebih lanjut menggunakan *Grad-CAM*.

Dataset

Dataset yang digunakan adalah *Chest X-Ray Pneumonia Balanced Dataset* yang diperoleh dari Kaggle. Dataset ini berisi citra rontgen dada yang dikelompokkan ke dalam dua kelas, yaitu NORMAL dan *PNEUMONIA*. Secara keseluruhan dataset asli memuat 8.530 citra dengan komposisi seimbang, yaitu 4.265 citra untuk masing-masing kelas. Keseimbangan ini merupakan keunggulan tersendiri karena dataset *pneumonia* yang beredar luas umumnya memiliki proporsi kelas yang timpang sehingga menuntut penanganan tambahan.

Dengan pertimbangan efisiensi waktu pelatihan dan keterbatasan sumber daya komputasi, tidak seluruh data digunakan. Sebanyak 4.000 citra dipilih sebagai sampel dan dibagi menjadi tiga subset dengan proporsi 70:15:15 sebagaimana disajikan pada Tabel 1.2.

Tabel 1.2 Distribusi Dataset yang Digunakan

Subset	Jumlah Citra	Proporsi	Fungsi
Training	2.800	70%	Membangun model klasifikasi
Validation	600	15%	Memantau performa dan mencegah *overfitting*
Testing	600	15%	Mengukur performa pada data baru
Total	4.000	100%	—

Pemisahan yang tegas antara ketiga subset merupakan hal yang wajib dijaga. Data pengujian tidak boleh disentuh sama sekali selama proses pelatihan maupun penalaan *hyperparameter*, karena satu kali saja data tersebut ikut memengaruhi keputusan perancangan, angka evaluasi yang dihasilkan tidak lagi merepresentasikan performa model pada data yang benar-benar baru.

Preprocessing

Tahap *preprocessing* bertujuan menghasilkan data yang seragam dan sesuai dengan kebutuhan arsitektur yang digunakan. Tahapan yang dilakukan meliputi empat langkah berikut:

a. *Pembacaan dan pelabelan data.* Seluruh citra dibaca dari direktori dataset dan diberi label sesuai kelasnya, yaitu NORMAL atau *PNEUMONIA*.
b. *Pembagian dataset.* Data dibagi menjadi subset pelatihan, validasi, dan pengujian sesuai proporsi pada Tabel 1.2.
c. *Pengubahan ukuran citra.* Seluruh citra diubah menjadi berukuran 224 × 224 piksel agar sesuai dengan ukuran masukan yang diharapkan oleh *ResNet-50* dan *EfficientNet-B0*.
d. *Normalisasi nilai piksel.* Nilai piksel disesuaikan menggunakan fungsi *preprocessing* bawaan masing-masing arsitektur agar distribusi masukan sesuai dengan kondisi ketika model dilatih pada *ImageNet*. Bentuk paling sederhana dari normalisasi adalah penskalaan rentang nilai piksel sebagaimana ditunjukkan pada Persamaan dibawah ini.

$$x' = \frac{x}{255} \qquad (1.14)$$

Perlu ditekankan bahwa setiap arsitektur memiliki skema normalisasi yang berbeda. *ResNet-50* pada Keras mengharapkan masukan dalam urutan kanal BGR dengan pengurangan rerata *ImageNet*, sedangkan *EfficientNet* memiliki lapisan penskalaan tersendiri. Menggunakan skema normalisasi yang keliru merupakan kesalahan yang sering terjadi dan dampaknya berupa penurunan akurasi yang sulit dilacak karena tidak menimbulkan pesan galat apa pun. Kode 1.1 memperlihatkan pembentukan *data generator* untuk kedua model.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.resnet50 import preprocess_input as resnet_pre
train_gen= ImageDataGenerator(preprocessing_function=resnet_pre)
train_data = train_gen.flow_from_directory(
   "dataset/train", target_size=(224, 224),
   batch_size=32, class_mode="binary")
```

Kode 1.1 Pembentukan Data Generator dengan Normalisasi Sesuai Arsitektur

Augmentasi Data

Augmentasi data merupakan teknik memperbanyak variasi data pelatihan dengan menghasilkan citra baru dari citra asli melalui berbagai transformasi tanpa mengubah informasi utama yang dikandungnya. Teknik ini berfungsi sebagai bentuk regularisasi: model tidak pernah melihat citra yang benar-benar identik dua kali sehingga lebih sulit menghafal data pelatihan dan terdorong untuk mempelajari pola yang lebih umum. Meskipun dataset yang digunakan telah seimbang, augmentasi tetap diperlukan agar model tahan terhadap variasi posisi pasien, orientasi, dan kualitas pencahayaan antarperangkat rontgen. Parameter augmentasi yang diterapkan dirangkum pada Tabel 1.3.

Tabel 1.3 Teknik Augmentasi yang Diterapkan pada Data Pelatihan

Teknik	Tujuan Klinis
Rotation	Mengakomodasi variasi posisi pasien saat pengambilan citra
Horizontal flip	Menambah variasi orientasi tanpa menghilangkan ciri paru-paru
Zoom	Melatih model mengenali pola *pneumonia* pada berbagai skala
Width shift dan height shift	Meningkatkan ketahanan terhadap pergeseran posisi objek
Brightness adjustment	Mengakomodasi perbedaan pencahayaan antarperangkat medis

Catatan: Augmentasi yang Aman secara Klinis
Tidak semua transformasi yang lazim digunakan pada citra umum dapat diterapkan pada citra medis. *Vertical flip* sengaja tidak digunakan pada bab ini karena membalik citra toraks secara vertikal menghasilkan anatomi yang mustahil ditemui pada praktik nyata, sehingga model justru dilatih mengenali kondisi yang tidak pernah ada.
Rotasi pun perlu dibatasi pada sudut kecil, umumnya di bawah 15 derajat. Demikian pula *horizontal flip* perlu dipertimbangkan secara hati-hati, sebab pembalikan horizontal menukar posisi

jantung dan menyerupai kondisi *situs inversus* yang sangat jarang terjadi. Prinsip utamanya sederhana: augmentasi harus menghasilkan citra yang masih mungkin muncul di ruang radiologi.

Perancangan Model

Kedua arsitektur dibangun menggunakan pendekatan *transfer learning*. Model dasar dimuat beserta bobot hasil pelatihan pada *ImageNet*, sedangkan lapisan klasifikasi bawaannya dilepas dan digantikan dengan kepala klasifikasi baru yang sesuai dengan permasalahan biner. Kepala klasifikasi tersebut terdiri atas lapisan *global average pooling* untuk meringkas *feature map* menjadi vektor, lapisan *dropout* sebagai regularisasi, dan satu neuron keluaran dengan aktivasi sigmoid. Struktur ini ditunjukkan pada Kode 1.2.

```
from tensorflow.keras.applications import ResNet50, EfficientNetB0
from tensorflow.keras import layers, models
base = ResNet50(weights="imagenet", include_top=False,
            input_shape=(224, 224, 3))
base.trainable = False
model = models.Sequential([
    base,
    layers.GlobalAveragePooling2D(),
    layers.Dropout(0.3),
    layers.Dense(1, activation="sigmoid")
])
model.compile(optimizer="adam", loss="binary_crossentropy",
          metrics=["accuracy"])
```

Kode 1.2 Pembentukan Model *Transfer Learning* dengan Kepala Klasifikasi Biner

Skenario Penalaan *Hyperparameter*

Penalaan *hyperparameter* dilakukan untuk memperoleh konfigurasi yang menghasilkan performa terbaik pada masing-masing arsitektur. Pengujian dilakukan melalui lima skenario eksperimen dengan nilai *learning rate* yang diturunkan secara bertahap dari 10^{-3} hingga 10^{-5}.

Nilai *batch size* ditetapkan tetap sebesar 32 dan jumlah *epoch* maksimum ditetapkan sebesar 35 pada seluruh skenario. Penetapan kedua nilai tersebut secara tetap merupakan keputusan metodologis yang disengaja: dengan hanya satu variabel yang berubah, perbedaan performa yang teramati dapat diatribusikan secara langsung kepada *learning rate*. Rancangan eksperimen tersebut disajikan pada Tabel 1.4.

Tabel 1.4 Skenario Penalaan *Hyperparameter*

Eksperimen	***Learning Rate***	***Batch Size***	***Epoch* Maksimum**
E1	0,001	32	35
E2	0,0005	32	35
E3	0,0001	32	35
E4	0,00005	32	35
E5	0,00001	32	35

Strategi Pelatihan dan *Callback*

Selain pengaturan *hyperparameter*, proses pelatihan dilengkapi dua mekanisme *callback*. *EarlyStopping* menghentikan pelatihan secara otomatis apabila nilai *validation loss* tidak membaik selama lima *epoch* berturut-turut, sekaligus mengembalikan bobot terbaik yang pernah dicapai. Mekanisme ini mencegah *overfitting* sekaligus menghemat waktu komputasi. Sementara itu, *ReduceLROnPlateau* menurunkan nilai *learning rate* secara adaptif apabila *validation loss* tidak membaik selama tiga *epoch* berturut-turut, sehingga model dapat melakukan penyesuaian yang lebih halus ketika mendekati titik optimum.

Konsekuensi penting dari penggunaan *EarlyStopping* adalah bahwa jumlah *epoch* pada Tabel 1.4 merupakan batas maksimum, bukan jumlah yang pasti dijalankan. Pada praktiknya sebagian besar eksperimen berhenti jauh sebelum *epoch* ke-35, dan hal ini akan terlihat jelas pada kurva pelatihan yang dibahas pada subbab berikutnya.

Prosedur Evaluasi dan Interpretasi

Evaluasi dilakukan menggunakan 600 citra pada subset pengujian yang belum pernah dilihat model selama pelatihan. Performa diukur menggunakan *accuracy*, *precision*, *recall*, *F1-score*, dan *ROC-AUC*, serta dilengkapi *confusion matrix* untuk memperlihatkan pola

kesalahan yang terjadi. Setelah konfigurasi terbaik diperoleh, dilakukan interpretasi menggunakan *Grad-CAM* terhadap sejumlah sampel yang terdiri atas kasus dengan prediksi benar dan kasus dengan prediksi salah. Pemilihan kedua jenis kasus tersebut bersifat disengaja, karena visualisasi pada kasus yang salah justru paling informatif untuk mengungkap dasar keputusan model yang keliru. Inti perhitungan *Grad-CAM* ditunjukkan pada Kode 1.3.

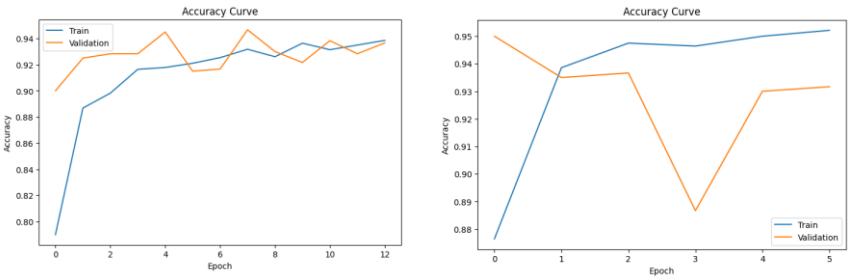
```
import tensorflow as tf
grad_model = tf.keras.models.Model(
    model.inputs,
    [model.get_layer(last_conv).output, model.output])
with tf.GradientTape() as tape:
    conv_out, preds = grad_model(img_array)
    loss = preds[:, 0]
grads = tape.gradient(loss, conv_out)
weights = tf.reduce_mean(grads, axis=(0, 1, 2))
heatmap = tf.reduce_sum(weights * conv_out[0], axis=-1)
heatmap = tf.maximum(heatmap, 0) / tf.reduce_max(heatmap)
```

Kode 1.3 Inti Perhitungan *Grad-CAM*

➢ Hasil dan Ujicoba

Hasil Pelatihan

Pengamatan terhadap kurva pelatihan merupakan langkah pertama dalam menilai kesehatan sebuah model, dan sebaiknya dilakukan sebelum melihat angka evaluasi apa pun. Kurva memperlihatkan bagaimana model belajar dari waktu ke waktu, sesuatu yang tidak dapat ditangkap oleh satu angka akurasi akhir. Perkembangan nilai *accuracy* pada kedua model selama pelatihan ditunjukkan pada Gambar 1.2.



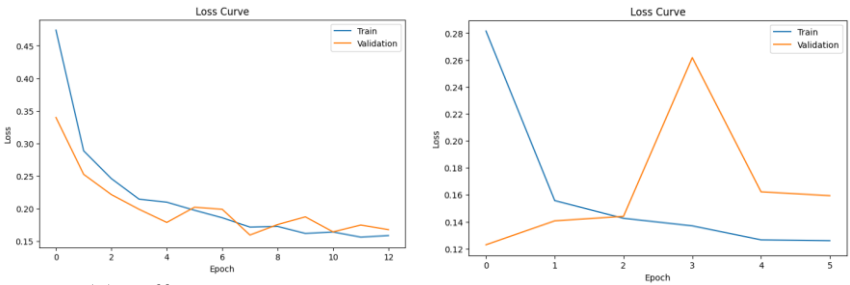
(a) *EfficientNet-B0* (b) *ResNet-50*

Gambar 1.2 Kurva *Accuracy* Pelatihan Kedua Model

Pada Gambar 1.2(a), nilai *training accuracy* dan *validation accuracy* model *EfficientNet-B0* sama-sama menunjukkan tren peningkatan. Meskipun terdapat beberapa fluktuasi pada kurva validasi, kedua kurva tetap berada pada rentang yang berdekatan hingga akhir pelatihan. Kondisi ini menunjukkan bahwa model mampu mempelajari pola pada citra *Chest X-ray* secara efektif dengan indikasi *overfitting* yang relatif rendah. Yang menarik, pada beberapa *epoch* awal nilai validasi justru berada di atas nilai pelatihan gejala yang lazim terjadi ketika *dropout* aktif selama pelatihan namun dinonaktifkan saat validasi.

Pada Gambar 1.2(b), *training accuracy* model *ResNet-50* meningkat secara konsisten hingga akhir pelatihan, sedangkan *validation accuracy* mencapai nilai tinggi meskipun disertai fluktuasi pada beberapa *epoch*. Perbedaan panjang sumbu horizontal antara kedua grafik juga patut diperhatikan: pelatihan *ResNet-50* berhenti pada *epoch* keenam sementara *EfficientNet-B0* berlanjut hingga *epoch* ketiga belas. Perbedaan ini bukan kekeliruan, melainkan konsekuensi langsung dari mekanisme *EarlyStopping* yang menghentikan pelatihan ketika perbaikan *validation loss* tidak lagi terjadi.

Sisi kedua dari proses pelatihan tercermin pada kurva *loss* yang ditunjukkan pada Gambar 1.3. Kurva ini sering kali lebih informatif daripada kurva akurasi karena mampu mendeteksi awal terjadinya *overfitting* lebih dini.



(a) *EfficientNet-B0* (b) *ResNet-50*

Gambar 1.3 Kurva *Loss* Pelatihan Kedua Model

Pada Gambar 1.3(a), nilai *training loss* dan *validation loss EfficientNet-B0* menurun secara berdampingan dengan selisih yang relatif kecil pada sebagian besar *epoch*. Pola ini menandakan proses optimasi berjalan stabil dan model berhasil mencapai konvergensi tanpa mengalami *overfitting* yang berarti. Pada Gambar 1.3(b), *training loss ResNet-50* terus menurun hingga akhir pelatihan, sedangkan *validation loss* menunjukkan fluktuasi yang lebih besar terutama pada awal pelatihan sebelum kembali menurun dan bertahan pada tingkat yang rendah.

Secara umum, kedua model berhasil menurunkan nilai *loss* dengan baik. *EfficientNet-B0* memperlihatkan pola pembelajaran yang lebih halus, sedangkan *ResNet-50* lebih bergejolak namun tetap mencapai tingkat *loss* akhir yang rendah. Perlu digarisbawahi bahwa kestabilan kurva pelatihan tidak identik dengan performa akhir yang lebih baik; sebagaimana akan terlihat pada subbab evaluasi, model dengan kurva yang lebih bergejolak justru menghasilkan angka evaluasi yang lebih unggul.

➢ Hasil Penalaan *Hyperparameter*

Hasil lengkap kelima skenario eksperimen pada model *ResNet-50* disajikan pada Tabel 1.5, dengan konfigurasi terbaik ditandai dengan huruf tebal.

Tabel 1.5 Hasil Penalaan *Hyperparameter ResNet-50*

Eks.	*Learning Rate*	*Accuracy*	*Precision*	*Recall*	*F1-Score*	*ROC-AUC*
E1	0,001	0,9417	0,9783	0,9033	0,9393	0,9908
E2	0,0005	0,9433	0,9555	0,9300	0,9426	0,9854
E3	0,0001	0,9333	0,9815	0,8833	0,9298	0,9865
E4	0,00005	0,9383	0,9647	0,9100	0,9365	0,9836
E5	0,00001	0,9133	0,9844	0,8400	0,9065	0,9811

Berdasarkan Tabel 1.5, *ResNet-50* menghasilkan performa terbaik pada eksperimen E2 dengan *learning rate* sebesar 0,0005. Konfigurasi ini menghasilkan *accuracy* sebesar 94,33%, *precision* sebesar 95,55%, *recall* sebesar 93,00%, *F1-score* sebesar 94,26%, dan *ROC-AUC* sebesar 98,54%. Eksperimen E1 sesungguhnya menghasilkan *precision* dan *ROC-AUC* yang sedikit lebih tinggi, namun E2 dipilih karena memberikan keseimbangan yang lebih baik pada seluruh metrik, khususnya pada *recall* dan *F1-score*. Pemilihan ini mencerminkan pertimbangan klinis yang telah dibahas pada Subbab 1.2.8: pada penapisan *pneumonia*, kemampuan menemukan kasus yang sakit lebih berharga daripada kemampuan menghindari alarm palsu.

Pola yang lebih menarik terlihat pada kolom *precision* dan *recall* secara bersamaan. Seiring mengecilnya *learning rate*, nilai *precision* justru meningkat hingga mencapai 98,44% pada E5, sedangkan *recall* menurun tajam dari 90,33% pada E1 menjadi 84,00% pada E5. Ini bukan kebetulan, melainkan gejala model yang menjadi semakin konservatif: dengan langkah pembaruan yang sangat kecil, model tidak sempat bergerak jauh dari kondisi awalnya dan cenderung hanya memberikan prediksi *pneumonia* ketika sangat yakin. Akibatnya, 48 dari 300 kasus *pneumonia* luput terdeteksi pada E5. Menurunkan *learning rate* tidak selalu memperbaiki model.

Hasil penalaan pada model *EfficientNet-B0* disajikan pada Tabel 1.6.

Tabel 1.6 Hasil Penalaan *Hyperparameter EfficientNet-B0*

Eks.	***Learning Rate***	***Accuracy***	***Precision***	***Recall***	***F1-Score***	***ROC-AUC***
E1	0,001	0,9350	0,9611	0,9067	0,9331	0,9821
E2	0,0005	0,9350	0,9547	0,9133	0,9336	0,9811

E3	**0,0001**	**0,9367**	**0,9613**	**0,9100**	**0,9349**	**0,9811**
E4	0,00005	0,9217	0,9634	0,8767	0,9180	0,9794
E5	0,00001	0,9233	0,9704	0,8733	0,9193	0,9745

EfficientNet-B0 mencapai performa terbaik pada eksperimen E3 dengan *learning rate* sebesar 0,0001, menghasilkan *accuracy* sebesar 93,67%, *precision* sebesar 96,13%, *recall* sebesar 91,00%, *F1-score* sebesar 93,49%, dan *ROC-AUC* sebesar 98,11%. Nilai *accuracy* pada E1, E2, dan E3 sesungguhnya sangat berdekatan dengan rentang perbedaan kurang dari 0,2%, sehingga E3 dipilih karena memberikan kombinasi metrik yang paling seimbang. Pola penurunan performa pada E4 dan E5 yang sama dengan *ResNet-50*, yaitu bahwa *learning rate* yang terlalu kecil menghambat konvergensi.

Rangkuman konfigurasi terbaik kedua model disajikan pada Tabel 1.7.

Tabel 1.7 Konfigurasi *Hyperparameter* Terbaik pada Kedua Model

Model	**Eksperimen Terbaik**	***Learning Rate***	***Batch Size***
ResNet-50	E2	0,0005	32
EfficientNet-B0	E3	0,0001	32

Perbedaan nilai *learning rate* optimum di antara kedua arsitektur yang patut digarisbawahi. *ResNet-50* mencapai kondisi terbaik pada nilai lima kali lebih besar dibandingkan *EfficientNet-B0*. Hal ini menunjukkan bahwa nilai *learning rate* tidak dapat dipindahkan begitu saja dari satu arsitektur ke arsitektur lain, meskipun keduanya dilatih pada dataset yang sama persis. Setiap arsitektur memiliki lanskap optimasi yang berbeda, sehingga penalaan harus diulang setiap kali arsitektur diganti.

Evaluasi Model *ResNet-50*

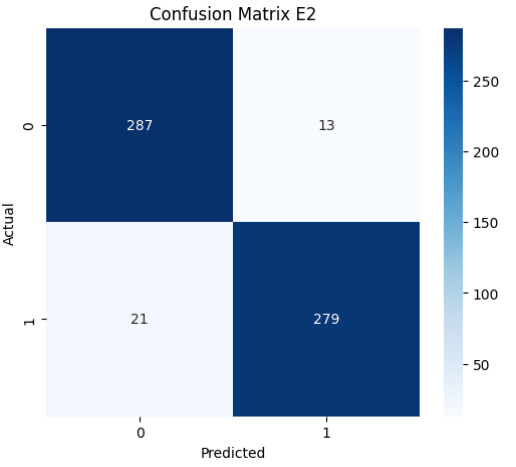
Evaluasi dilakukan menggunakan 600 citra pada subset pengujian dengan konfigurasi E2. Hasil pengukuran seluruh metrik disajikan pada Tabel 1.8.

Tabel 1.8 Hasil Evaluasi Model *ResNet-50*

Metrik	**Nilai**
Accuracy	94,33%
Precision	95,54%
Recall	93,00%
F1-Score	94,25%
ROC-AUC	98,53%

Nilai *accuracy* sebesar 94,33% menunjukkan bahwa sebagian besar citra berhasil diklasifikasikan dengan benar. Nilai *precision* sebesar 95,54% menandakan kemampuan yang baik dalam menekan prediksi positif yang keliru, sedangkan *recall* sebesar 93,00% menunjukkan bahwa sebagian besar kasus *pneumonia* berhasil terdeteksi. Nilai *F1-score* sebesar 94,25% mencerminkan keseimbangan yang baik di antara kedua metrik tersebut. Nilai *ROC-AUC* sebesar 98,53% patut mendapat perhatian khusus karena jauh lebih tinggi daripada *accuracy*; hal ini menandakan bahwa kemampuan pemisahan model sesungguhnya sangat baik dan sebagian kesalahan yang terjadi berpotensi diperbaiki hanya dengan menyesuaikan nilai ambang keputusan, tanpa perlu melatih ulang model.

Pola kesalahan yang terjadi dapat ditelusuri lebih rinci melalui *confusion matrix* pada Gambar 1.4.



Gambar 1.4 *Confusion Matrix* Model *ResNet-50* pada Konfigurasi E2

Berdasarkan Gambar 1.4, dari 300 citra NORMAL sebanyak 287 citra berhasil diklasifikasikan dengan benar dan 13 citra keliru diprediksi sebagai *PNEUMONIA*. Pada kelas *PNEUMONIA*, sebanyak 279 citra dikenali dengan benar sementara 21 citra keliru diprediksi sebagai NORMAL. Susunan angka tersebut memperlihatkan sesuatu yang tidak tampak pada nilai *accuracy*: kesalahan model tidak terdistribusi merata. Jumlah *false negative* sebanyak 21 kasus lebih banyak daripada *false positive* sebanyak 13 kasus, padahal *false negative* justru merupakan jenis kesalahan yang paling berbahaya pada konteks penapisan *pneumonia*. Bagi penerapan nyata, hal ini merupakan alasan kuat untuk menurunkan nilai ambang keputusan dari 0,5 agar model menjadi lebih sensitif, dengan konsekuensi bertambahnya alarm palsu yang masih dapat ditoleransi.

Evaluasi Model *EfficientNet-B0*

Evaluasi pada model *EfficientNet-B0* dilakukan dengan prosedur yang sama menggunakan konfigurasi E3. Hasilnya disajikan pada Tabel 1.9.

Tabel 1.9 Hasil Evaluasi Model *EfficientNet-B0*

Metrik	**Nilai**
Accuracy	93,66%
Precision	96,12%
Recall	91,00%
F1-Score	93,49%
ROC-AUC	98,10%

EfficientNet-B0 memperoleh *accuracy* sebesar 93,66% dengan *precision* sebesar 96,12%. Nilai *precision* yang tinggi menunjukkan bahwa model sangat berhati-hati dalam menyatakan sebuah citra sebagai *pneumonia*. Namun nilai *recall* sebesar 91,00% lebih rendah dibandingkan *ResNet-50*, yang berarti terdapat 27 kasus *pneumonia* yang tidak berhasil terdeteksi. Nilai *F1-score* sebesar 93,49% dan *ROC-AUC* sebesar 98,10% menunjukkan bahwa performa keseluruhan model tetap tergolong sangat baik.

Capaian tersebut menjadi jauh lebih berkesan apabila dikaitkan dengan jumlah parameter kedua model. *EfficientNet-B0* hanya menggunakan sekitar 5,3 juta parameter, sedangkan *ResNet-50* menggunakan sekitar 25,6 juta parameter. Dengan kata lain, selisih *accuracy* yang hanya sebesar 0,67% diperoleh dengan biaya parameter hampir lima kali lipat.

Perbandingan Performa Kedua Model

Perbandingan langsung performa kedua model disajikan pada Tabel 1.10.

Tabel 1.10 Perbandingan Performa *ResNet-50* dan *EfficientNet-B0*

Model	***Accuracy***	***Precision***	***Recall***	***F1-Score***	***ROC-AUC***
ResNet-50	94,33%	95,54%	93,00%	94,25%	98,53%
EfficientNet-B0	93,66%	96,12%	91,00%	93,49%	98,10%

Kedua model menunjukkan performa yang sangat baik dengan selisih yang tergolong kecil. *ResNet-50* unggul pada *accuracy*, *recall*, *F1-score*, dan *ROC-AUC*, sedangkan *EfficientNet-B0* unggul tipis pada *precision*. Pola ini merupakan wujud nyata dari *trade-off* antara *precision* dan *recall* yang telah dibahas pada landasan teori: *EfficientNet-B0* berperilaku lebih konservatif sehingga jarang salah ketika menyatakan *pneumonia*, tetapi konsekuensinya lebih banyak kasus *pneumonia* yang terlewat.

Pilihan model terbaik karena itu tidak dapat dipisahkan dari konteks penggunaannya. Untuk keperluan penapisan awal, yaitu skenario yang menjadi fokus bab ini, kasus *pneumonia* yang terlewat merupakan kegagalan yang jauh lebih mahal daripada rujukan yang tidak perlu, sehingga *ResNet-50* dengan *recall* 93,00% menjadi pilihan yang lebih tepat dan digunakan pada tahap interpretasi *Grad-CAM*. Namun apabila sistem hendak diterapkan pada perangkat bergerak atau fasilitas dengan sumber daya komputasi terbatas, efisiensi *EfficientNet-B0* dapat menjadi pertimbangan yang lebih menentukan daripada selisih *recall* sebesar dua persen.

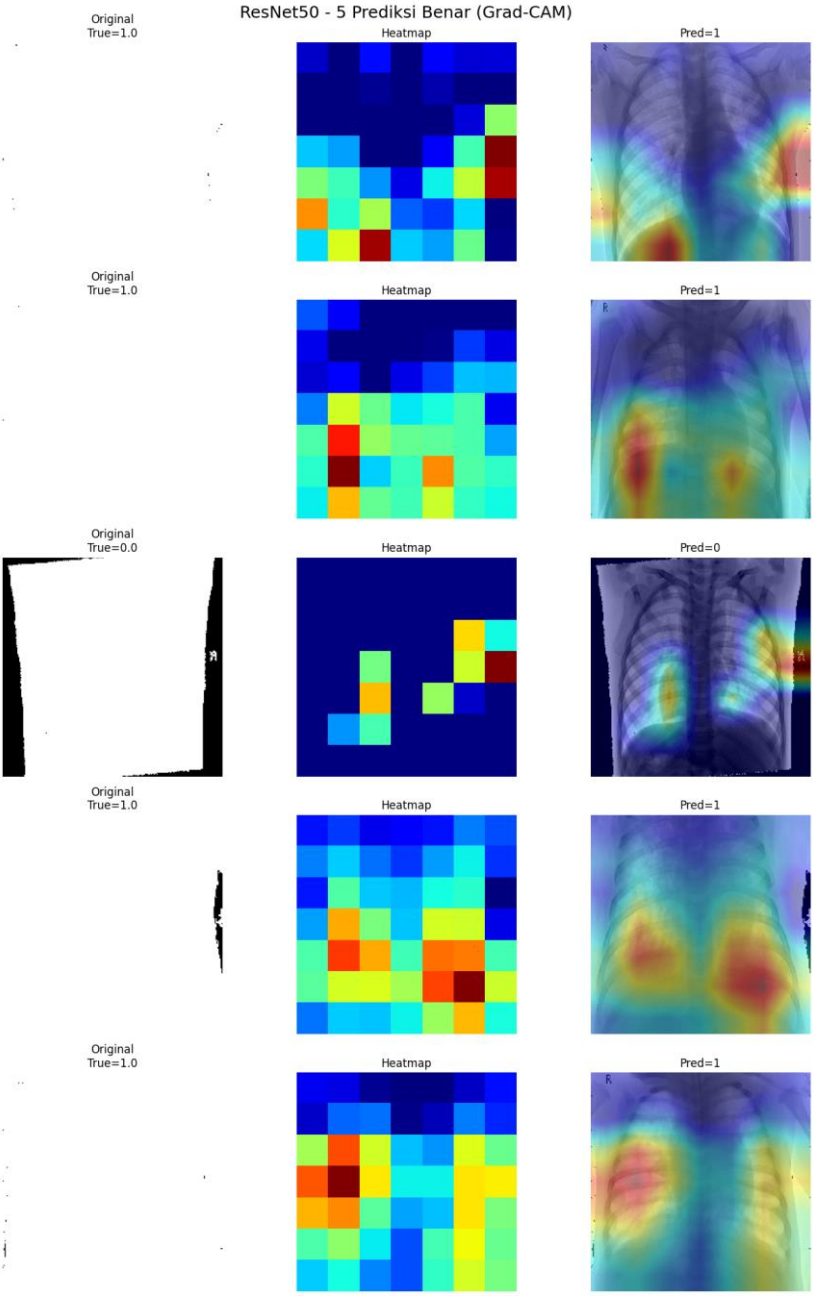
Interpretasi Model dengan *Grad-CAM*

Seluruh angka evaluasi yang telah dibahas hanya menjawab pertanyaan mengenai seberapa sering model benar, tetapi tidak menjawab pertanyaan yang jauh lebih penting bagi penerapan klinis, yaitu apakah model benar karena alasan yang tepat. *Grad-CAM* diterapkan untuk menjawab pertanyaan tersebut. Visualisasi pada model *ResNet-50* untuk kasus dengan prediksi benar ditunjukkan pada Gambar 1.5.

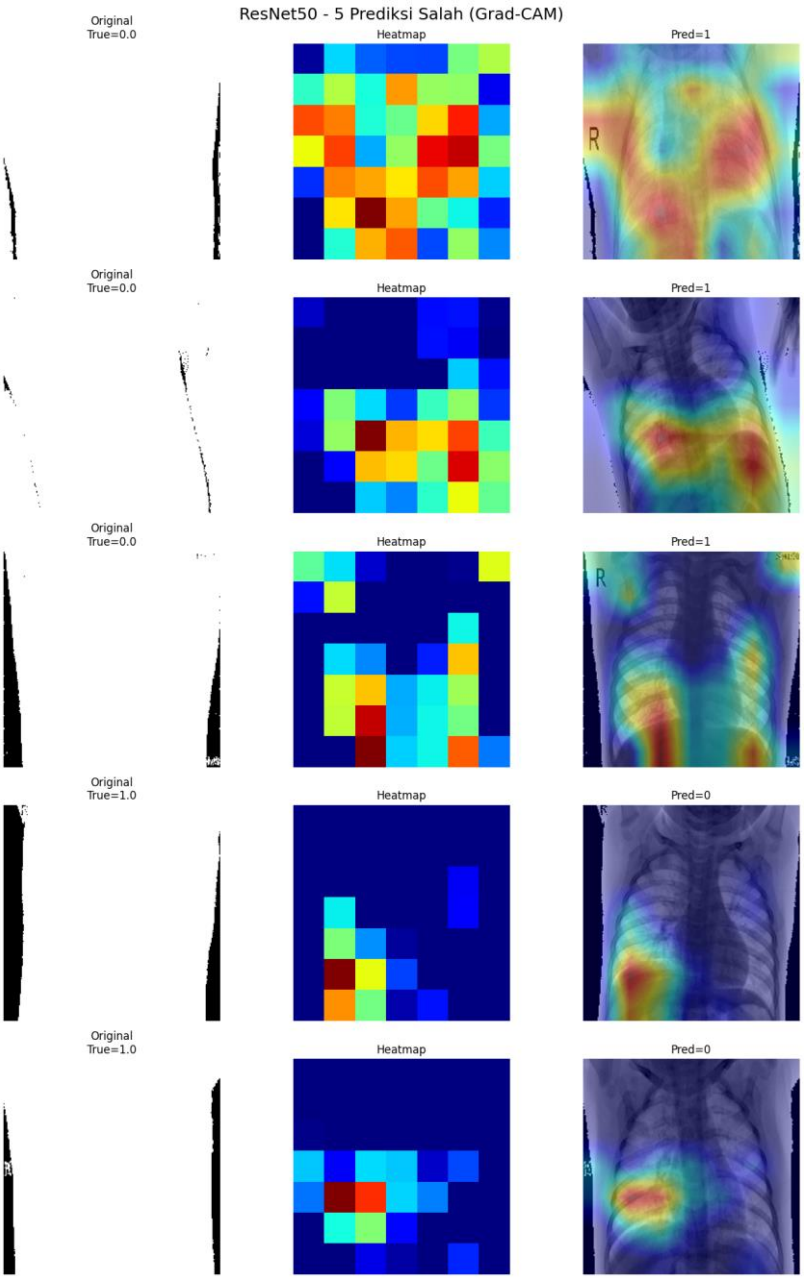
Setiap baris pada gambar tersebut memuat citra asli, *heatmap Grad-CAM* dengan resolusi asli lapisan konvolusi terakhir, dan hasil penumpangan *heatmap* pada citra masukan. Terlihat bahwa area dengan kontribusi terbesar, yang ditandai warna kuning hingga merah, terpusat pada wilayah paru-paru. Pola ini menunjukkan bahwa keputusan model didasarkan pada informasi yang relevan secara klinis. Sebagai pembanding, visualisasi pada kasus dengan prediksi salah ditunjukkan pada Gambar 1.6.

Pada kasus yang keliru, aktivasi model terlihat jauh lebih menyebar dan sebagian berada di luar wilayah paru-paru, seperti pada tepi citra dan penanda orientasi. Kondisi ini menunjukkan bahwa perhatian model tidak sepenuhnya terfokus pada area yang relevan sehingga memengaruhi hasil prediksi. Visualisasi yang sama pada model *EfficientNet-B0* disajikan pada Gambar 1.7 dan Gambar 1.8.

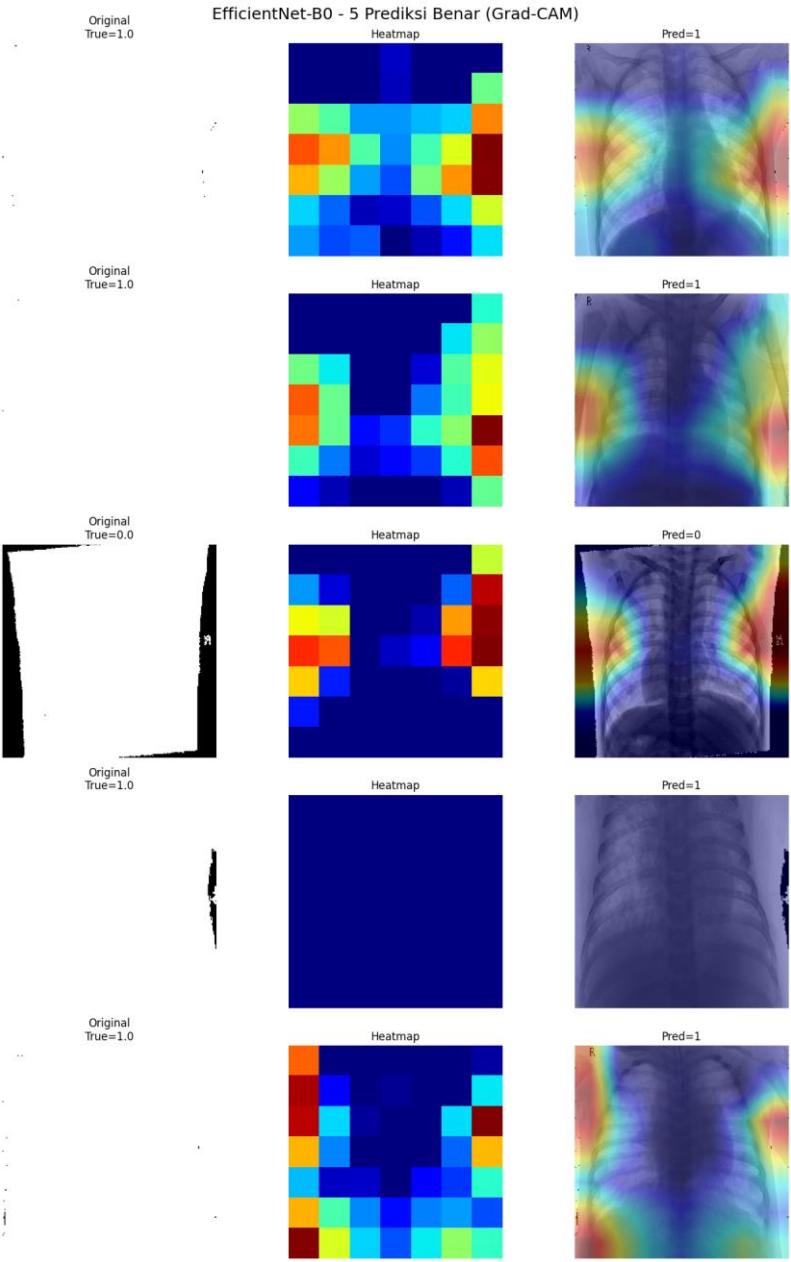
EfficientNet-B0 juga mampu mengidentifikasi area paru-paru sebagai sumber informasi utama pada kasus dengan prediksi benar. Namun pada beberapa kasus yang keliru, aktivasi kembali terlihat pada area nonklinis dan bagian tepi citra. Secara keseluruhan, pola aktivasi *ResNet-50* terlihat lebih konsisten terpusat pada wilayah paru-paru dibandingkan *EfficientNet-B0*, dan konsistensi ini sejalan dengan hasil evaluasi kuantitatif yang menempatkan *ResNet-50* sebagai model dengan performa lebih baik.



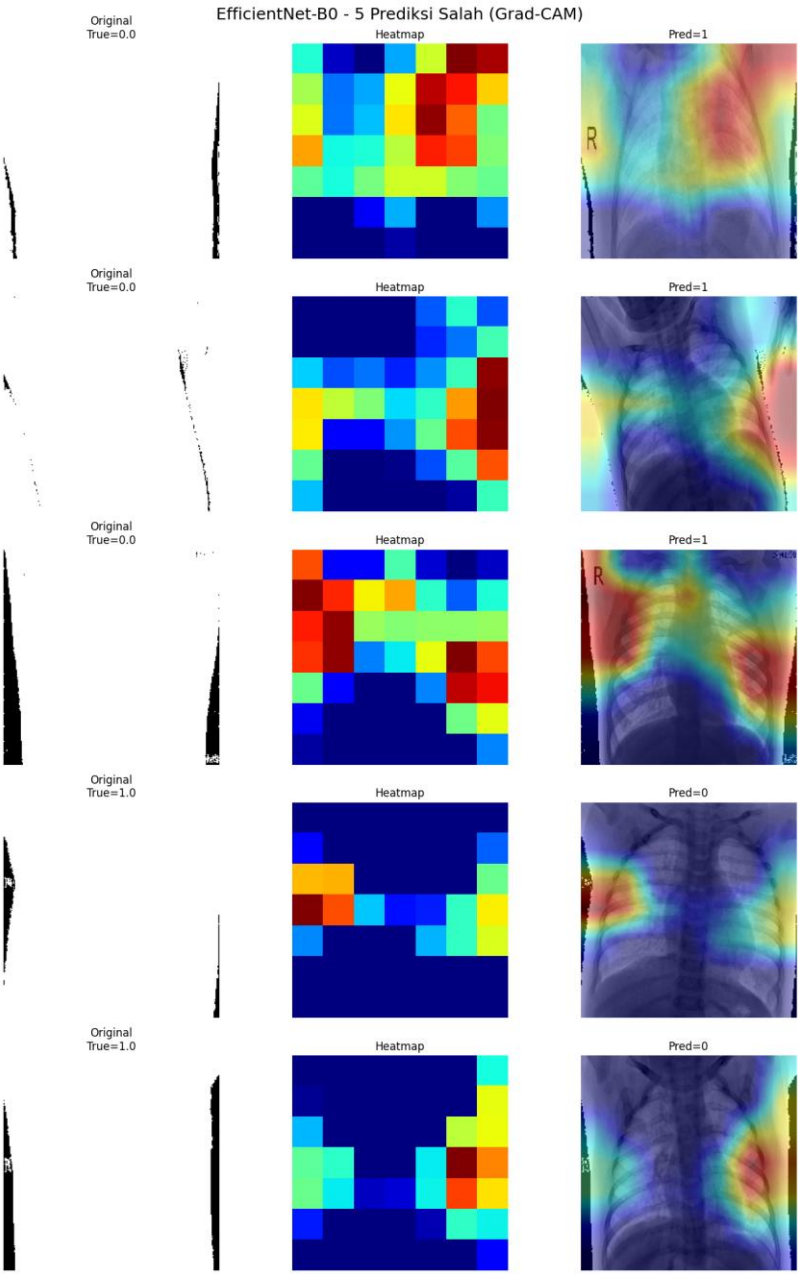
Gambar 1.5 Visualisasi *Grad-CAM* pada Prediksi Benar Model *ResNet-50*



Gambar 1.6 Visualisasi *Grad-CAM* pada Prediksi Salah Model *ResNet-50*



Gambar 1.7 Visualisasi *Grad-CAM* pada Prediksi Benar Model *EfficientNet-B0*



Gambar 1.8 Visualisasi *Grad-CAM* pada Prediksi Salah Model *EfficientNet-B0*

Peringatan: Gejala *Shortcut Learning*
Model mengaktifkan penanda orientasi dan tepi citra merupakan hasil terpenting pada bab ini, sekaligus yang paling mudah terabaikan. Gejala ini dikenal sebagai *shortcut learning*, yaitu kondisi ketika model memperoleh akurasi tinggi bukan dengan memahami patologi, melainkan dengan mengeksploitasi korelasi yang kebetulan ada pada data.
Penanda orientasi yang ditempelkan teknisi radiologi merupakan contoh yang khas. Apabila citra pasien *pneumonia* pada sebuah rumah sakit kebetulan lebih sering diambil menggunakan perangkat portabel dengan gaya penandaan tertentu, model dapat mempelajari penanda tersebut sebagai penentu kelas. Model semacam ini akan tampak sangat akurat pada data pengujian yang berasal dari sumber yang sama, lalu gagal total ketika diterapkan di rumah sakit lain.
Inilah sebabnya evaluasi kuantitatif saja tidak pernah memadai untuk sistem medis. Tanpa *Grad-CAM*, angka *accuracy* sebesar 94,33% akan terbaca sebagai keberhasilan tanpa cacat.

Pembahasan dan Keterbatasan

Hasil yang diperoleh pada bab ini perlu dibaca bersama sejumlah keterbatasan yang melekat pada rancangan eksperimennya. Keterbatasan tersebut tidak mengurangi nilai temuan, tetapi menentukan sejauh mana kesimpulan dapat digeneralisasikan.

a. *Sumber data tunggal.* Seluruh citra berasal dari satu dataset publik sehingga model belum diuji terhadap *domain shift*, yaitu perbedaan karakteristik citra yang timbul dari perbedaan merek perangkat rontgen, protokol pengambilan, maupun populasi pasien. Performa pada data dari rumah sakit lain hampir pasti akan lebih rendah.
b. *Penggunaan sebagian data.* Hanya 4.000 dari 8.530 citra yang digunakan karena keterbatasan sumber daya komputasi. Pemanfaatan seluruh data berpeluang meningkatkan kemampuan generalisasi model.
c. *Klasifikasi biner.* Model hanya membedakan NORMAL dan *PNEUMONIA*, sedangkan pada praktik nyata radiolog menghadapi berbagai kelainan lain seperti efusi pleura,

tuberkulosis, dan edema paru. Model ini akan memaksakan salah satu dari dua label pada kondisi apa pun yang ditemuinya.

d. *Ruang penalaan yang sempit.* Penalaan hanya dilakukan terhadap *learning rate*, sedangkan *batch size*, laju *dropout*, dan strategi *fine-tuning* tidak ikut dieksplorasi.
e. *Belum ada validasi klinis.* Sistem belum diuji pada alur kerja nyata dan belum dibandingkan dengan pembacaan radiolog, sehingga belum dapat dinyatakan layak pakai.
f. *Indikasi shortcut learning.* Aktivasi pada penanda orientasi menunjukkan bahwa sebagian keputusan model berpotensi bertumpu pada informasi nonklinis, dan hal ini merupakan persoalan yang harus diselesaikan sebelum penerapan lebih lanjut.

Sejumlah arah pengembangan dapat ditempuh untuk mengatasi keterbatasan tersebut. Penerapan segmentasi paru-paru pada tahap *preprocessing* akan memaksa model mengabaikan area di luar paru dan merupakan langkah paling langsung untuk menekan *shortcut learning*. Penggunaan dataset yang lebih besar dan lebih beragam, disertai validasi eksternal pada data dari institusi berbeda, akan memberikan gambaran performa yang lebih jujur. Eksplorasi arsitektur lain seperti DenseNet, ConvNeXt, atau *Vision Transformer* dapat memperluas perbandingan, sementara penggunaan metode XAI pembanding seperti LIME, Score-CAM, atau SHAP dapat memperkuat keandalan interpretasi karena kesimpulan yang hanya bertumpu pada satu metode interpretasi selalu rentan terhadap kelemahan metode tersebut.

➢ Ringkasan

Bab ini telah menguraikan pembangunan sistem klasifikasi *pneumonia* pada citra *Chest X-ray* secara menyeluruh menggunakan pendekatan *transfer learning* dengan arsitektur *ResNet-50* dan *EfficientNet-B0*, dilengkapi interpretasi berbasis *Grad-CAM*. Beberapa hal penting dapat dirangkum sebagai berikut.

a. Konfigurasi terbaik *ResNet-50* diperoleh pada *learning rate* 0,0005, sedangkan *EfficientNet-B0* pada *learning rate* 0,0001. Perbedaan ini menegaskan bahwa penalaan harus diulang setiap kali arsitektur diganti.

b. *ResNet-50* memperoleh *accuracy* 94,33% dan *recall* 93,00%, sedangkan *EfficientNet-B0* memperoleh *accuracy* 93,66% dan *recall* 91,00%. *ResNet-50* dipilih sebagai model terbaik karena keunggulannya pada *recall*, yang merupakan metrik paling kritis pada penapisan medis.
c. Menurunkan *learning rate* tidak selalu memperbaiki model. Nilai yang terlalu kecil membuat model menjadi terlalu konservatif sehingga *precision* meningkat namun *recall* menurun tajam.
d. *EfficientNet-B0* mencapai performa yang hampir setara dengan hanya sekitar seperlima jumlah parameter *ResNet-50*, sehingga tetap menarik untuk penerapan dengan sumber daya terbatas.
e. *Grad-CAM* menunjukkan bahwa kedua model umumnya berfokus pada area paru-paru pada prediksi yang benar, tetapi memperlihatkan aktivasi pada penanda orientasi dan tepi citra pada prediksi yang salah. Akurasi tinggi belum menjamin model dapat dipercaya.

Gagasan yang paling penting untuk dibawa ke bab berikutnya adalah bahwa evaluasi sebuah model medis tidak berhenti pada angka. Sebuah model yang benar karena alasan yang keliru akan gagal pada saat yang paling merugikan, yaitu ketika diterapkan pada pasien nyata di lingkungan yang berbeda dari data pelatihannya.

➢ Daftar Pustaka

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Dalam Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), hlm. 770–778.

Islam, M. R., dkk. (2022). *Transfer Learning* Based *Pneumonia* Detection from *Chest X-ray* Images Using EfficientNet. Journal of Medical Imaging and Health Informatics.

Kermany, D. S., dkk. (2018). Identifying Medical Diagnoses and Treatable Diseases by Image-Based *Deep Learning*. Cell, 172(5), 1122–1131.

Richardo, dkk. (2025). Penerapan *Grad-CAM* untuk Interpretasi Model *Deep Learning* pada Klasifikasi Citra *Chest X-ray*.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). *Grad-CAM*: Visual Explanations from Deep Networks via Gradient-Based Localization. Dalam Proceedings of the IEEE International Conference on Computer Vision (ICCV), hlm. 618–626.

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for *Convolutional Neural Network*s. Dalam Proceedings of the 36th International Conference on Machine Learning (ICML), hlm. 6105–6114.

World Health Organization. (2022). *Pneumonia* in Children. Jenewa: WHO.

Zech, J. R., dkk. (2018). Variable Generalization Performance of a *Deep Learning* Model to Detect *Pneumonia* in Chest Radiographs: A Cross-Sectional Study. PLOS Medicine, 15(11).

BAB 2
SEGMENTASI TUMOR OTAK PADA CITRA MRI MENGGUNAKAN *U-NET* DAN ESTIMASI TINGKAT KEPARAHAN BERDASARKAN LUAS AREA TUMOR

Setelah mempelajari bab ini, pembaca diharapkan mampu:

a. Menjelaskan perbedaan mendasar antara klasifikasi citra dan segmentasi citra beserta konsekuensinya terhadap perancangan model dan metrik evaluasi.
b. Menguraikan prinsip kerja arsitektur *U-Net* serta empat variannya, yaitu Attention *U-Net*, Residual *U-Net*, *U-Net*++, dan *Pretrained U-Net*.
c. Menerapkan *preprocessing* dan augmentasi berpasangan pada citra MRI beserta mask segmentasinya.
d. Menghitung dan menafsirkan *Dice Score*, *Intersection over Union*, *precision*, dan *recall* pada tingkat piksel.
e. Menganalisis pengaruh ketidakseimbangan kelas dan pemilihan ambang keputusan terhadap hasil segmentasi.
f. Memanfaatkan hasil segmentasi untuk menghitung luas area tumor sebagai indikator kuantitatif tingkat keparahan.

➢ Pendahuluan

Tumor otak merupakan penyakit yang secara langsung mengganggu fungsi sistem saraf pusat dan berpotensi menimbulkan komplikasi serius apabila tidak terdeteksi sejak dini. Berbeda dengan banyak penyakit lain, dampak sebuah tumor otak tidak hanya ditentukan oleh keberadaannya, melainkan juga oleh lokasi dan ukurannya. Sebuah lesi berukuran kecil pada area yang mengatur fungsi vital dapat jauh lebih berbahaya daripada lesi berukuran besar pada area yang relatif tidak kritis. Karena itu, penentuan lokasi dan ukuran tumor secara tepat merupakan langkah yang menentukan dalam proses diagnosis maupun perencanaan tindakan medis.

Magnetic Resonance Imaging (MRI) merupakan modalitas pencitraan yang paling banyak digunakan untuk keperluan tersebut karena kemampuannya menghasilkan citra jaringan lunak dengan tingkat detail yang tinggi. Melalui citra MRI, area tumor dapat dibedakan dari jaringan sehat berdasarkan perbedaan karakteristik intensitas dan teksturnya. Namun proses penandaan batas tumor secara manual, yang dikenal sebagai *delineasi*, merupakan pekerjaan yang sangat menyita waktu. Seorang radiolog harus menelusuri puluhan irisan citra untuk satu orang pasien, dan hasil penandaan antara dua radiolog yang berbeda pun tidak selalu identik. Variabilitas antarpengamat inilah yang menjadi persoalan tersendiri ketika ukuran tumor hendak dijadikan dasar pengambilan keputusan klinis.

Bab sebelumnya telah memperlihatkan bagaimana *Convolutional Neural Network* dapat menentukan apakah sebuah citra mengandung kelainan atau tidak. Bab ini melangkah satu tingkat lebih jauh dengan mengajukan pertanyaan yang lebih menuntut: bukan sekadar *apakah* terdapat tumor, melainkan *di mana persisnya* tumor itu berada dan *seberapa luas* areanya. Pergeseran ini mengubah keluaran model dari sebuah angka probabilitas tunggal menjadi sebuah peta keputusan yang harus dihasilkan untuk setiap piksel pada citra. Perubahan tersebut membawa konsekuensi menyeluruh, mulai dari rancangan arsitektur, pemilihan fungsi kerugian, hingga metrik yang digunakan untuk menilai keberhasilan model.

Arsitektur yang paling banyak digunakan untuk tugas semacam ini adalah *U-Net*. Rancangannya menggabungkan jalur *encoder* yang memampatkan citra menjadi representasi semantik dengan jalur *decoder* yang mengembalikan representasi tersebut ke resolusi semula, serta menghubungkan keduanya melalui *skip connection* agar informasi spasial yang hilang selama pemampatan dapat dipulihkan. Sejak diperkenalkan, *U-Net* telah melahirkan banyak varian yang masing-masing berupaya menyempurnakan satu aspek tertentu, misalnya dengan menambahkan mekanisme perhatian, koneksi residual, jalur *skip* yang lebih rapat, atau memanfaatkan *encoder* yang telah dilatih sebelumnya. Bab ini membandingkan lima varian tersebut secara langsung pada satu dataset yang sama.

Hasil segmentasi memiliki nilai yang melampaui sekadar penentuan lokasi. Setelah batas tumor diketahui, luas areanya dapat dihitung secara langsung dari jumlah piksel yang terkandung di dalamnya, dan angka ini dapat dimanfaatkan sebagai indikator kuantitatif tingkat keparahan. Inilah keunggulan utama pendekatan segmentasi dibandingkan klasifikasi: keluarannya bukan sekadar label, melainkan besaran terukur yang dapat dipantau perkembangannya dari waktu ke waktu, misalnya untuk menilai respons pasien terhadap terapi.

Bab ini menuntun pembaca membangun sistem segmentasi tumor otak secara utuh menggunakan *Brain MRI Segmentation Dataset* yang memuat citra pasien *Lower Grade Glioma*. Pembahasan mencakup penyiapan data berpasangan antara citra dan *mask*, penerapan augmentasi yang menjaga keselarasan keduanya, pelatihan lima varian *U-Net*, penalaan *hyperparameter* pada model terbaik, interpretasi hasil menggunakan *Grad-CAM++*, hingga perhitungan luas area tumor. Sepanjang pembahasan, satu tema akan muncul berulang kali, yaitu bahwa ketidakseimbangan antara area tumor dan area latar belakang merupakan tantangan utama pada segmentasi medis dan memengaruhi hampir setiap keputusan perancangan yang diambil.

➢ Pengantar Teori

Bagian ini menguraikan landasan teori yang diperlukan pada bab ini. Karena sebagian konsep dasar seperti operasi konvolusi, fungsi aktivasi, dan prinsip *transfer learning* telah dibahas pada Bab 1, pembahasan di sini difokuskan pada hal-hal yang khas bagi permasalahan segmentasi.

Dari Klasifikasi menuju Segmentasi

Pada permasalahan klasifikasi, sebuah citra berukuran 224 × 224 piksel dipetakan menjadi satu keluaran tunggal. Pada permasalahan segmentasi, citra berukuran 240 × 240 piksel harus dipetakan menjadi 57.600 keputusan, satu untuk setiap piksel. Segmentasi karena itu dapat dipandang sebagai klasifikasi biner yang dilakukan secara serentak pada seluruh piksel, dengan syarat tambahan bahwa keputusan pada piksel yang bertetangga harus saling konsisten

sehingga menghasilkan area yang utuh, bukan bercak-bercak yang terpisah.

Perubahan ini menimbulkan dua persoalan teknis yang tidak ditemui pada klasifikasi. Persoalan pertama adalah pemulihan resolusi. Rangkaian *pooling* yang membuat CNN efisien justru menghancurkan informasi posisi yang sangat dibutuhkan pada segmentasi; sebuah jaringan yang telah memampatkan citra 240 × 240 menjadi peta 15 × 15 tidak lagi memiliki informasi yang cukup untuk menandai batas tumor secara presisi. Persoalan kedua adalah ketidakseimbangan kelas yang jauh lebih ekstrem. Pada dataset yang digunakan, area tumor umumnya hanya menempati beberapa persen dari keseluruhan luas citra, sehingga sebuah model yang memprediksi seluruh piksel sebagai latar belakang akan tetap memperoleh akurasi piksel di atas 95%. Kenyataan inilah yang membuat *accuracy* sama sekali tidak berguna sebagai metrik pada segmentasi medis.

Arsitektur *U-Net*

U-Net menjawab persoalan pemulihan resolusi melalui rancangan berbentuk huruf U yang terdiri atas tiga bagian. Jalur *encoder* atau jalur kontraksi bekerja seperti CNN biasa: rangkaian konvolusi dan *pooling* secara bertahap memperkecil dimensi spasial sambil memperbanyak jumlah kanal, sehingga menghasilkan representasi yang semakin kaya secara semantik namun semakin miskin secara spasial. Bagian *bottleneck* di dasar huruf U memuat representasi paling padat. Jalur *decoder* atau jalur ekspansi kemudian mengembalikan dimensi spasial secara bertahap melalui operasi *transposed convolution*.

Kunci keberhasilan *U-Net* terletak pada *skip connection* yang menghubungkan setiap tingkat *encoder* dengan tingkat *decoder* yang bersesuaian. Melalui jalur ini, *feature map* beresolusi tinggi dari *encoder* digabungkan langsung dengan *feature map* hasil *upsampling* pada *decoder*. Gagasannya sederhana namun sangat efektif: *decoder* memperoleh informasi semantik mengenai *apa* objek yang sedang disegmentasi dari jalur utama, sekaligus memperoleh informasi spasial mengenai *di mana* letak batasnya dari jalur pintas. Tanpa mekanisme ini, keluaran segmentasi akan tampak kabur dan kehilangan detail tepi. Rancangan tersebut pula yang

membuat *U-Net* mampu bekerja baik meskipun jumlah data pelatihan terbatas, sebuah keunggulan yang sangat berarti pada domain medis.

Varian Arsitektur *U-Net*

Empat varian *U-Net* dibahas pada bab ini, masing-masing menyempurnakan satu aspek tertentu dari rancangan aslinya. *Attention U-Net* menyisipkan sebuah *attention gate* pada setiap *skip connection*. Persoalan yang hendak diselesaikan adalah bahwa *skip connection* konvensional meneruskan seluruh isi *feature map encoder* tanpa membedakan mana yang relevan dan mana yang sekadar latar belakang. *Attention gate* menghitung sebuah peta bobot berdasarkan gabungan sinyal dari *encoder* dan sinyal pemandu dari *decoder*, kemudian mengalikan peta bobot tersebut dengan *feature map* yang diteruskan. Secara ringkas, mekanisme ini dapat dinyatakan sebagai berikut.

$$\alpha = \sigma_2(\psi(\sigma_1(W_x x + W_g g + b))),\ x_{out} = \alpha \cdot x \qquad (2.1)$$

Dengan x menyatakan *feature map* dari *encoder*, g menyatakan sinyal pemandu dari *decoder*, σ_1 menyatakan fungsi ReLU, σ_2 menyatakan fungsi sigmoid, dan x dengan indeks *out* menyatakan *feature map* terbobot yang diteruskan ke *decoder*. Karena nilai α berada pada rentang nol sampai satu, area yang dinilai tidak relevan akan ditekan mendekati nol sebelum mencapai *decoder*.

Residual U-Net mengganti blok konvolusi biasa dengan blok residual sebagaimana telah dibahas pada Bab 1 melalui Persamaan 1.3. Manfaatnya sama seperti pada *ResNet*, yaitu memperlancar aliran gradien sehingga jaringan dapat dilatih lebih dalam dan mempelajari fitur yang lebih kompleks.

U-Net++ atau *Nested U-Net* berangkat dari pengamatan bahwa *skip connection* pada *U-Net* asli menggabungkan dua *feature map* yang secara semantik berbeda jauh: keluaran *encoder* masih bersifat mentah sedangkan keluaran *decoder* sudah sangat terolah. Kesenjangan semantik ini diperkecil dengan menyisipkan rangkaian blok konvolusi perantara pada jalur *skip*, membentuk struktur bersarang yang rapat. *U-Net++* juga menerapkan *deep supervision*, yaitu penghitungan kerugian pada beberapa tingkat keluaran sekaligus.

Pretrained U-Net mengganti seluruh jalur *encoder* dengan arsitektur yang telah dilatih pada *ImageNet*, dalam hal ini *ResNet34* dan *ResNet50*. Pendekatan ini merupakan penerapan *transfer learning* pada tugas segmentasi. Karena *encoder* semacam ini dirancang untuk citra berwarna, citra MRI yang bersifat *grayscale* perlu direplikasi menjadi tiga kanal terlebih dahulu.

Metrik Evaluasi Segmentasi

Metrik utama pada segmentasi medis adalah *Dice Score*, yang mengukur derajat tumpang tindih antara area prediksi dan area *ground truth*.

$$Dice = \frac{2\,|A \cap B|}{|A| + |B|} \qquad (2.2)$$

Dengan A menyatakan himpunan piksel yang diprediksi sebagai tumor dan B menyatakan himpunan piksel tumor sebenarnya. Apabila dinyatakan dalam istilah *confusion matrix* tingkat piksel, *Dice* memiliki bentuk yang setara dengan *F1-score*.

$$Dice = \frac{2\,TP}{2\,TP + FP + FN} \quad (2.3)$$

Perhatikan bahwa jumlah *true negative* tidak muncul sama sekali pada rumus tersebut. Inilah alasan *Dice* menjadi metrik pilihan pada segmentasi medis: piksel latar belakang yang jumlahnya sangat dominan tidak diperhitungkan sehingga metrik ini tidak dapat dikelabui oleh model yang memprediksi seluruh citra sebagai latar belakang.

Metrik kedua adalah *Intersection over Union* (IoU) atau indeks Jaccard, yang mengukur rasio antara irisan dan gabungan kedua himpunan.

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN} \quad (2.4)$$

Dice dan IoU mengukur hal yang pada dasarnya sama dan terhubung melalui hubungan monoton berikut.

$$IoU = \frac{Dice}{2 - Dice} \qquad (2.5)$$

Karena hubungan tersebut bersifat monoton, urutan peringkat model tidak akan berubah apabila salah satu metrik diganti dengan metrik lainnya. Nilai IoU selalu lebih kecil daripada *Dice* untuk

segmentasi yang tidak sempurna, sehingga IoU dapat dipandang sebagai ukuran yang lebih ketat. Selain kedua metrik tersebut, *precision* dan *recall* tetap digunakan dengan definisi yang sama seperti pada Bab 1, hanya saja perhitungannya dilakukan pada tingkat piksel. *Precision* yang rendah menandakan model menandai area yang lebih luas daripada tumor sebenarnya, sedangkan *recall* yang rendah menandakan sebagian area tumor tidak tertandai.

Fungsi Kerugian untuk Segmentasi

Pilihan fungsi kerugian pada segmentasi tidak sesederhana pada klasifikasi. *Binary cross-entropy* yang dihitung untuk setiap piksel merupakan pilihan yang paling langsung, namun memiliki kelemahan serius pada data yang sangat tidak seimbang. Karena kerugian dihitung sebagai rata-rata seluruh piksel, kontribusi dari puluhan ribu piksel latar belakang akan menenggelamkan kontribusi dari beberapa ratus piksel tumor.

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i\, log(p_i) + (1 - y_i)\, log(1 - p_i)] \quad (2.6)$$

Alternatifnya adalah *Dice loss*, yaitu bentuk *Dice* yang dibuat dapat diturunkan dengan menggunakan probabilitas prediksi secara langsung tanpa proses *thresholding*.

$$L_{Dice} = 1 - \frac{2\sum_i p_i y_i + \varepsilon}{\sum_i p_i + \sum_i y_i + \varepsilon} \quad (2.7)$$

Dengan ε menyatakan konstanta kecil yang mencegah pembagian dengan nol pada irisan citra yang tidak mengandung tumor sama sekali. *Dice loss* secara bawaan tahan terhadap ketidakseimbangan kelas, tetapi gradiennya cenderung tidak stabil pada awal pelatihan. Karena itu praktik yang lazim adalah menggabungkan keduanya sehingga *binary cross-entropy* menyediakan gradien yang stabil sementara *Dice loss* mengarahkan optimasi kepada metrik yang sesungguhnya ingin dimaksimalkan.

$$L_{total} = \lambda\, L_{Dice} + (1 - \lambda)\, L_{BCE} \quad (2.8)$$

Pada bab ini digunakan nilai λ sebesar 0,8 sehingga komponen *Dice* mendominasi proses optimasi.

Estimasi Luas Area dan Tingkat Keparahan

Setelah *mask* segmentasi diperoleh, luas area tumor dalam satuan piksel dihitung dengan menjumlahkan seluruh piksel yang bernilai satu.

$$A_{piksel} = \sum_i \sum_j M(i,j) \quad (2.9)$$

Dengan M menyatakan *mask* biner hasil segmentasi. Nilai tersebut kemudian dikonversi ke satuan fisik menggunakan informasi ukuran piksel yang tercatat pada metadata citra MRI.

$$A_{mm^2} = A_{piksel} \times s_x \times s_y \quad (2.10)$$

Dengan s menyatakan jarak antarpiksel pada masing-masing sumbu dalam satuan milimeter. Perlu diperhatikan bahwa proses pengubahan ukuran citra menjadi 240 × 240 piksel pada tahap *preprocessing* turut mengubah nilai s tersebut, sehingga faktor skala harus disesuaikan agar luas yang dihitung tetap merepresentasikan ukuran sebenarnya. Kelalaian pada langkah ini merupakan sumber kesalahan yang sering terjadi dan sepenuhnya tidak terlihat pada metrik evaluasi mana pun.

Ukuran yang lebih tahan terhadap perbedaan skala adalah rasio antara luas tumor dan luas penampang otak pada irisan yang sama.

$$R = \frac{A_{tumor}}{A_{otak}} \times 100\% \quad (2.11)$$

Rasio ini bersifat nirdimensi sehingga dapat dibandingkan antarpasien maupun antarperangkat pencitraan. Nilai R selanjutnya dapat dipetakan ke dalam kategori tingkat keparahan sebagai indikator pendukung. Perlu ditegaskan bahwa ambang kategori semacam ini bersifat konvensional dan bukan merupakan standar klinis; penentuannya harus dilakukan bersama klinisi dan divalidasi terhadap data hasil pengukuran manual.

Grad-CAM++

Interpretasi model pada bab ini menggunakan *Grad-CAM++*, yaitu pengembangan dari *Grad-CAM* yang telah dibahas pada Bab 1. *Grad-CAM* asli menghitung bobot setiap kanal melalui rata-rata sederhana atas gradien, sehingga cenderung menonjolkan satu area dominan saja. *Grad-CAM++* menggantikan rata-rata tersebut

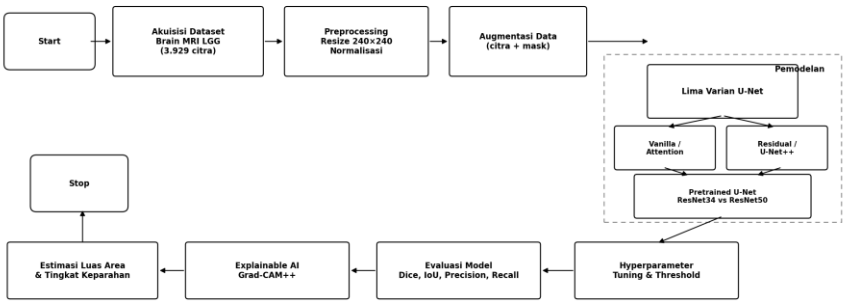
dengan rata-rata terbobot yang memanfaatkan turunan tingkat kedua dan ketiga, sehingga mampu menangani citra yang memuat beberapa area penting sekaligus dan menghasilkan cakupan yang lebih merata pada objek berukuran besar.

Pada konteks segmentasi, penggunaan metode interpretasi semacam ini sekilas tampak berlebihan karena keluaran model sudah berupa peta spasial yang dapat langsung diamati. Manfaatnya terletak pada perbedaan yang diungkapkannya: *mask* keluaran memperlihatkan keputusan akhir setelah proses *thresholding*, sedangkan *Grad-CAM++* memperlihatkan area yang memengaruhi keputusan tersebut pada tingkat representasi internal. Ketika keduanya tidak sejalan, misalnya ketika model memberikan perhatian pada jaringan sehat namun tetap menghasilkan *mask* yang tepat, hal itu merupakan pertanda bahwa keberhasilan yang teramati mungkin bersifat kebetulan.

➢ Metodologi

Alur Metodologi

Keseluruhan proses yang dibahas pada bab ini disusun sebagai satu alur kerja berurutan sebagaimana ditunjukkan pada Gambar 2.1.



Gambar 2.1 Alur Metodologi Segmentasi Tumor Otak pada Citra MRI

Proses diawali dengan akuisisi *Brain MRI Segmentation Dataset*, dilanjutkan dengan *preprocessing* dan augmentasi yang diterapkan secara berpasangan pada citra dan *mask*. Tahap pemodelan menguji lima varian *U-Net* untuk memperoleh arsitektur dengan performa terbaik, yang kemudian dikembangkan lebih lanjut

melalui perbandingan *encoder ResNet34* dan *ResNet50* disertai penalaan *hyperparameter*. Model final dievaluasi menggunakan *Dice*, IoU, *precision*, dan *recall*, diinterpretasikan menggunakan *Grad-CAM++*, dan hasil segmentasinya dimanfaatkan untuk mengestimasi luas area tumor.

Dataset

Dataset yang digunakan adalah *Brain MRI Segmentation Dataset* yang memuat 3.929 citra MRI otak dari 110 pasien penderita *Lower Grade Glioma* (LGG) beserta *mask* segmentasi tumor sebagai *ground truth*. Citra MRI disimpan dalam format TIFF tiga kanal yang merepresentasikan urutan *pre-contrast*, FLAIR, dan *post-contrast*, sedangkan *mask* segmentasi berupa citra biner satu kanal dengan nilai 0 untuk latar belakang dan nilai 1 untuk area tumor.

Hasil eksplorasi data memperlihatkan ketidakseimbangan distribusi yang signifikan sebagaimana disajikan pada Tabel 2.1.

Tabel 2.1 Distribusi Irisan Citra pada Dataset

Kategori Irisan	**Jumlah Citra**	**Proporsi**
Mengandung tumor	1.373	34,9%
Tidak mengandung tumor	2.556	65,1%
Total	**3.929**	**100%**

Ketidakseimbangan ini merupakan hal yang wajar pada dataset segmentasi medis, karena sebuah tumor hanya tampak pada sebagian irisan dari keseluruhan volume MRI seorang pasien. Namun kewajaran tersebut tidak membuatnya menjadi tidak berbahaya. Perlu pula dicatat bahwa ketidakseimbangan terjadi pada dua tingkat sekaligus, yaitu pada tingkat irisan sebagaimana ditunjukkan tabel di atas, dan pada tingkat piksel di dalam setiap irisan yang mengandung tumor. Ketidakseimbangan tingkat piksel justru jauh lebih ekstrem dan menjadi alasan utama pemilihan *Dice* sebagai metrik serta *DiceBCELoss* sebagai fungsi kerugian.

Dataset dibagi menjadi data pelatihan, validasi, dan pengujian dengan rasio 80:10:10 menggunakan nilai *seed* 42 agar pembagiannya dapat direproduksi. Pembagian tersebut ditunjukkan pada Tabel 2.2.

Tabel 2.2 Pembagian Dataset

Subset	**Proporsi**	**Perkiraan Jumlah Citra**	**Fungsi**
Training	80%	± 3.143	Melatih parameter model
Validation	10%	± 393	Memilih model terbaik dan ambang optimal
Testing	10%	± 393	Mengukur performa akhir

Preprocessing

Tahap *preprocessing* bertujuan menyeragamkan format data sebelum digunakan dalam pelatihan. Tahapan yang dilakukan meliputi langkah-langkah berikut.

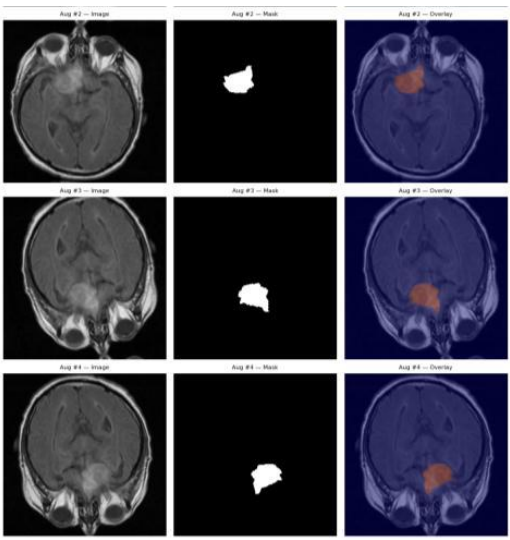
a. *Pembacaan data berpasangan.* Setiap citra MRI dibaca bersama *mask* segmentasinya sebagai satu pasangan yang tidak boleh terpisah pada seluruh tahap berikutnya.
b. *Pengubahan ukuran.* Seluruh citra dan *mask* diubah menjadi berukuran 240 × 240 piksel. Interpolasi pada *mask* harus menggunakan metode *nearest neighbour*, sebab interpolasi bilinier akan menghasilkan nilai antara yang merusak sifat biner label.
c. *Normalisasi nilai piksel.* Nilai intensitas citra dinormalisasi untuk menstabilkan proses pelatihan. Pada model *Pretrained U-Net*, citra dikonversi menjadi tiga kanal dan dinormalisasi menggunakan statistik rerata serta simpangan baku *ImageNet* agar sesuai dengan kondisi pelatihan *encoder*.
d. *Binerisasi mask. Mask* dibinerisasi menggunakan ambang 0,5 sehingga menghasilkan label 0 untuk latar belakang dan 1 untuk tumor.
e. *Pembagian dataset.* Data dibagi dengan rasio 80 : 10 : 10 menggunakan *seed* 42.

Catatan: Kebocoran Data pada Dataset Pasien
Pembagian data pada dataset medis menyimpan jebakan yang tidak tampak pada dataset citra umum. Dataset ini memuat 3.929

irisan yang berasal dari hanya 110 pasien, sehingga rata-rata setiap pasien menyumbang sekitar 36 irisan yang saling berdekatan dan sangat mirip satu sama lain.
Apabila pembagian dilakukan secara acak pada tingkat irisan, irisan ke-20 dan ke-21 milik pasien yang sama dapat terpisah ke data pelatihan dan data pengujian. Model kemudian dinilai pada citra yang nyaris identik dengan citra yang telah dipelajarinya, dan hasilnya adalah angka evaluasi yang terlalu optimistis. Praktik yang benar adalah melakukan pembagian pada tingkat pasien, yaitu memastikan seluruh irisan milik satu pasien berada pada subset yang sama.

Augmentasi Data

Augmentasi pada tugas segmentasi memiliki satu syarat yang tidak berlaku pada klasifikasi, yaitu bahwa setiap transformasi geometris harus diterapkan secara serentak dan identik pada citra maupun *mask*-nya. Apabila citra diputar sementara *mask* tidak, maka pasangan data yang dihasilkan menjadi salah dan model akan dilatih menggunakan label yang tidak sesuai. Contoh hasil augmentasi berpasangan ditunjukkan pada Gambar 2.2.



Gambar 2.2 Contoh Hasil Augmentasi Berpasangan antara Citra MRI dan Mask

Teknik augmentasi yang diterapkan beserta alasannya dirangkum pada Tabel 2.3.

Tabel 2.3 Teknik Augmentasi yang Diterapkan

Teknik	**Parameter**	**Tujuan**
Horizontal Flip	p = 0,5	Menambah variasi orientasi lateral
Vertical Flip	p = 0,5	Menambah variasi orientasi kraniokaudal
ShiftScaleRotate	batas 0,05	Mensimulasikan variasi posisi dan ukuran tumor
Border reflection	—	Mengisi area kosong akibat transformasi geometris

Augmentasi hanya diterapkan pada data pelatihan. Data validasi dan pengujian hanya melalui proses pengubahan ukuran dan normalisasi sehingga evaluasi dilakukan pada kondisi yang merepresentasikan data asli. Menerapkan augmentasi pada data evaluasi akan membuat angka yang dihasilkan kehilangan makna sebagai penduga performa nyata.

Perlu dicatat bahwa *vertical flip* digunakan pada bab ini, berbeda dengan Bab 1 yang justru menghindarinya. Perbedaan ini bukan ketidakkonsistenan, melainkan konsekuensi dari perbedaan modalitas. Citra rontgen dada memiliki orientasi atas-bawah yang bermakna secara anatomis sehingga pembalikan vertikal menghasilkan citra yang mustahil, sedangkan irisan MRI aksial otak jauh lebih simetris sehingga pembalikan vertikal masih menghasilkan citra yang masuk akal. Pelajaran umumnya adalah bahwa kelayakan sebuah teknik augmentasi selalu bergantung pada modalitas dan bidang irisan yang digunakan, bukan pada daftar teknik yang berlaku universal.

Perbandingan Kelima Arsitektur

Lima arsitektur yang diuji pada bab ini memiliki kesamaan pada struktur dasar berupa *encoder*, *bottleneck*, *decoder*, dan *skip connection*, serta menghasilkan keluaran berupa *mask* satu kanal. Perbedaan rincinya disajikan pada Tabel 2.4.

Tabel 2.4 Perbandingan Rinci Kelima Arsitektur *U-Net*

Komponen	**Vanilla *U-Net***	**Attention *U-Net***	***Pretrained ResNet34***	**Residual *U-Net***	***U-Net++***
Masukan	MRI 1 kanal	MRI 1 kanal	MRI 3 kanal	MRI 1 kanal	MRI 1 kanal
Encoder	CNN standar	CNN standar	*ResNet34*	CNN residual	CNN *nested*
Bobot *encoder*	Acak	Acak	*ImageNet*	Acak	Acak
Transfer learning	Tidak	Tidak	Ya	Tidak	Tidak
Filter encoder	64–512	64–512	*Stage ResNet34*	64–512	64–512
Bottleneck	1024	1024	Fitur *ResNet34*	1024	1024
Skip connection	*Concat*	*Attention + Concat*	*Encoder–Decoder*	*Concat*	Dense/*Nested*
Mekanisme tambahan	—	*Attention Gate*	*Transfer Learning*	Residual	*Dense Skip*
Deep supervision	Tidak	Tidak	Tidak	Tidak	Ya
Normalisasi	*BatchNorm*	*BatchNorm*	*BatchNorm*	*BatchNorm*	*BatchNorm*
Pooling	Max 2×2	Max 2×2	Encoder *ResNet*	Max 2×2	Max 2×2
Upsampling	*Transposed* 2×2	*Transposed* 2×2	*Decoder Upsample*	*Transposed* 2×2	*Transposed* 2×2
Lapisan keluaran	Conv 1×1	Conv 1×1	*Segmentation Head*	Conv 1×1	Conv 1×1
Aktivasi keluaran	Sigmoid	Sigmoid	Logits + Sigmoid	Sigmoid	Sigmoid

Vanilla *U-Net* berperan sebagai model dasar atau *baseline* karena rancangannya paling sederhana. Attention *U-Net* menambahkan empat *attention gate*, satu pada setiap *skip connection*. Residual *U-Net* menerapkan koneksi residual pada setiap blok konvolusi. *Pretrained U-Net ResNet34* memanfaatkan *transfer learning* dan karenanya memerlukan masukan tiga kanal. *U-Net++* menerapkan jalur *skip* bersarang disertai *deep supervision*.

Konfigurasi Pelatihan dan *Hyperparameter*

Penalaan *hyperparameter* pada model utama tidak dilakukan dengan menguji satu variabel secara terpisah seperti pada Bab 1, melainkan dengan menyusun satu konfigurasi menyeluruh yang setiap komponennya dipilih untuk mengatasi kendala tertentu pada permasalahan ini. Konfigurasi lengkapnya disajikan pada Tabel 2.5.

Tabel 2.5 Konfigurasi *Hyperparameter* Model Utama *Pretrained U-Net ResNet50*

Komponen	**Konfigurasi**	**Alasan Pemilihan**
Arsitektur	*Pretrained U-Net*, encoder *ResNet50*	Representasi fitur *ImageNet*
Ukuran citra	240 × 240 piksel	Menyeragamkan dimensi masukan
Batch size	4	Menyesuaikan keterbatasan memori GPU
Maksimum *epoch*	100	Batas atas, dibatasi *early stopping*
Freeze encoder	5 *epoch* pertama	Mencegah bobot *pretrained* rusak di awal
Optimizer	AdamW	Pemisahan weight decay dari gradien
Learning rate	Encoder 1e-5, *Decoder* 5e-5	Encoder cukup disesuaikan, *decoder* dilatih penuh
Regularisasi	Weight decay 1e-4	Menekan *overfitting*
Fungsi kerugian	*Dice*BCE*Loss*	Menggabungkan kestabilan dan ketahanan kelas

Bobot kerugian	*Dice* 0,8 dan BCE 0,2	Mengutamakan optimasi terhadap *Dice*
Augmentasi	HFlip, VFlip, ShiftScaleRotate	Meningkatkan generalisasi
Penyeimbangan data	Empty ratio 1,0	Menyeimbangkan irisan tumor dan non-tumor
Scheduler	ReduceLROnPlateau, factor 0,5, patience 5	Penyesuaian halus saat mendekati optimum
Early stopping	Patience 15, min delta 1e-4	Mencegah *overfitting*
Pencarian ambang	Rentang 0,35–0,50	Mengoptimalkan *Dice* pada data validasi

Dua komponen pada tabel tersebut layak memperoleh perhatian khusus. Yang pertama adalah penggunaan *learning rate* yang berbeda antara *encoder* dan *decoder*. *Encoder* telah memuat representasi visual yang berguna sehingga hanya perlu disesuaikan secara halus, sedangkan *decoder* dimulai dari bobot acak sehingga memerlukan langkah pembaruan yang lebih besar. Menerapkan satu nilai yang sama bagi keduanya akan merusak representasi *pretrained* apabila nilainya terlalu besar, atau memperlambat pembelajaran *decoder* apabila nilainya terlalu kecil. Pembekuan *encoder* selama lima *epoch* pertama memperkuat strategi yang sama dengan memberi kesempatan *decoder* mencapai kondisi yang wajar sebelum gradien merambat ke *encoder*.

Yang kedua adalah *empty ratio* sebesar 1,0, yaitu penyeimbangan jumlah irisan tanpa tumor terhadap irisan bertumor pada setiap *epoch* pelatihan. Tanpa mekanisme ini, dua pertiga data pelatihan tidak memuat tumor sama sekali dan model akan condong memilih prediksi kosong. Perlu ditegaskan bahwa penyeimbangan hanya diterapkan pada data pelatihan; data pengujian tetap mempertahankan distribusi aslinya sehingga hasil evaluasi tetap merepresentasikan kondisi nyata.

Prosedur Evaluasi

Evaluasi dilakukan menggunakan dua cara pembacaan yang berbeda dan keduanya perlu dilaporkan bersama. *Dice all slices* dihitung pada seluruh irisan data pengujian, termasuk irisan yang tidak mengandung tumor sama sekali, sedangkan *Dice tumor-only* dihitung hanya pada irisan yang benar-benar mengandung tumor. Perbedaan keduanya bersifat mendasar: metrik pertama menilai kemampuan model menahan diri dari prediksi palsu pada irisan kosong, sedangkan metrik kedua menilai ketepatan penandaan batas tumor.

Selain itu dilakukan pencarian ambang optimal pada rentang 0,35 sampai 0,50 menggunakan data validasi. Nilai ambang 0,5 yang menjadi bawaan tidak selalu merupakan pilihan terbaik pada segmentasi dengan kelas yang tidak seimbang, karena menurunkan ambang akan memperluas area prediksi sehingga menaikkan *recall* dengan mengorbankan *precision*. Pencarian ambang harus dilakukan pada data validasi dan tidak boleh menggunakan data pengujian, sebab menyetel ambang berdasarkan data pengujian sama artinya dengan membocorkan informasi evaluasi ke dalam proses perancangan.

➢ Hasil dan Ujicoba

Perbandingan Kelima Varian *U-Net*

Hasil pelatihan kelima varian arsitektur disajikan pada Tabel 2.6.

Tabel 2.6 Hasil Eksperimen Kelima Varian *U-Net*

Metrik	Vanilla *U-Net*	Attention *U-Net*	*Pretrained ResNet34*	Residual *U-Net*	*U-Net++*
Best *epoch*	35	39	33	25	36
Validation Dice (tumor)	0,3244	0,7555	0,7923	0,7779	0,7517
Test *Dice all slices*	0,4020	0,4952	0,4898	0,5262	0,4328
Test *Dice* tumor-*only*	0,8039	0,7738	0,7804	0,7759	0,7751
Test IoU tumor-*only*	0,6721	0,6761	0,6919	0,6823	0,6827
Test *Precision* tumor-*only*	0,8827	0,7939	0,8485	0,8003	0,8204

Test *Recall* tumor-*only*	0,7380	0,8247	0,8041	0,8220	0,7938

Pembacaan tabel tersebut menuntut kehati-hatian karena tidak terdapat satu model pun yang unggul pada seluruh metrik. Residual *U-Net* memperoleh nilai tertinggi pada *Test Dice all slices* sebesar 0,5262, yang menunjukkan kemampuan relatif lebih baik ketika evaluasi mencakup seluruh irisan. Koneksi residual tampaknya membantu aliran informasi sehingga model lebih konsisten dalam membedakan irisan bertumor dan irisan kosong.

Hal yang paling mencolok pada tabel tersebut adalah selisih besar antara *Dice all slices* dan *Dice tumor-only* pada seluruh model, misalnya 0,4020 berbanding 0,8039 pada Vanilla *U-Net*. Selisih ini bukan sekadar keanehan angka, melainkan pesan yang cukup jelas: ketika model menemukan tumor, penandaan batasnya cukup akurat; persoalan sesungguhnya terletak pada irisan yang tidak mengandung tumor, tempat model masih menghasilkan bercak prediksi palsu. Pada irisan kosong, satu piksel prediksi yang keliru sudah cukup membuat nilai *Dice* pada irisan tersebut menjadi nol, sehingga rata-ratanya turun tajam. Dengan demikian, hambatan utama sistem ini bukanlah kualitas segmentasi, melainkan kemampuan menahan diri untuk tidak memprediksi apa pun ketika memang tidak ada yang perlu diprediksi.

Vanilla *U-Net* memperoleh *Dice tumor-only* tertinggi sebesar 0,8039 dan *precision* tertinggi sebesar 0,8827, tetapi sekaligus *recall* terendah sebesar 0,7380. Pola ini menandakan model yang bersifat konservatif: prediksi yang dihasilkannya cenderung benar, namun sebagian area tumor tidak tertandai. Pada konteks perencanaan tindakan medis, area tumor yang terlewat merupakan kegagalan yang lebih merugikan daripada area yang tertandai berlebihan, sehingga *precision* yang tinggi di sini tidak dapat langsung dibaca sebagai keunggulan.

Perhatian: Anomali pada *Validation Dice* Vanilla *U-Net*
Nilai *validation Dice* Vanilla *U-Net* sebesar 0,3244 jauh berada di bawah keempat model lainnya yang berkisar pada 0,75, namun *Dice tumor-only* pada data pengujian justru tertinggi sebesar 0,8039. Ketidakselarasan sebesar ini tidak wajar dan tidak dapat dijelaskan hanya oleh perbedaan arsitektur.
Penjelasan yang paling mungkin adalah bahwa kedua angka tersebut dihitung pada populasi irisan yang berbeda, yaitu seluruh irisan pada data validasi dan hanya irisan bertumor pada data pengujian. Kemungkinan lain adalah pengaruh ambang keputusan yang belum dioptimalkan pada tahap ini.
Ketika sebuah angka menyimpang sejauh ini dari pola keseluruhan, langkah yang benar adalah menelusuri asal-usulnya terlebih dahulu, bukan langsung mencari pembenaran teoretis atas keunggulan yang tampak. Pembaca dianjurkan memverifikasi ulang protokol perhitungan sebelum menarik kesimpulan dari perbandingan semacam ini.

Attention *U-Net* memperoleh *recall* tertinggi sebesar 0,8247 dan Residual *U-Net* mengikuti dengan 0,8220, yang menunjukkan bahwa keduanya lebih sensitif dalam menandai area tumor. Namun *recall* yang tinggi pada kedua model tersebut tidak disertai *precision* yang tinggi, sehingga terdapat kecenderungan menandai area yang lebih luas daripada tumor sebenarnya. *Pretrained U-Net ResNet34* memperlihatkan profil yang paling seimbang dengan *validation Dice* tertinggi sebesar 0,7923, IoU tertinggi sebesar 0,6919, serta *precision* 0,8485 dan *recall* 0,8041 yang berdekatan. Atas dasar kestabilan inilah arsitektur tersebut dipilih untuk dikembangkan lebih lanjut.

Pengembangan Model Utama: *ResNet34* dan *ResNet50*

Tahap berikutnya membandingkan penggunaan *encoder ResNet34* dan *ResNet50* pada arsitektur *Pretrained U-Net* untuk mengetahui apakah penambahan kedalaman jaringan memberikan peningkatan performa. Pelatihan dilakukan dengan batas 100 *epoch* disertai *early stopping* dan pencarian ambang optimal. Hasilnya disajikan pada Tabel 2.7.

Tabel 2.7 Perbandingan Encoder *ResNet34* dan *ResNet50* pada *Pretrained U-Net*

Metrik	*Pretrained U-Net ResNet34*	*Pretrained U-Net ResNet50*
Best *epoch*	69	57
Best *Validation Dice* (tumor)	0,8041	0,8070
Ambang optimal	0,40	0,50
Test *Dice all slices*	0,7496	0,7445
Test *Dice* tumor-*only*	0,8023	0,8099
Test IoU tumor-*only*	0,7131	0,7183
Test *Precision* tumor-*only*	0,8445	0,8366
Test *Recall* tumor-*only*	0,8241	0,8375

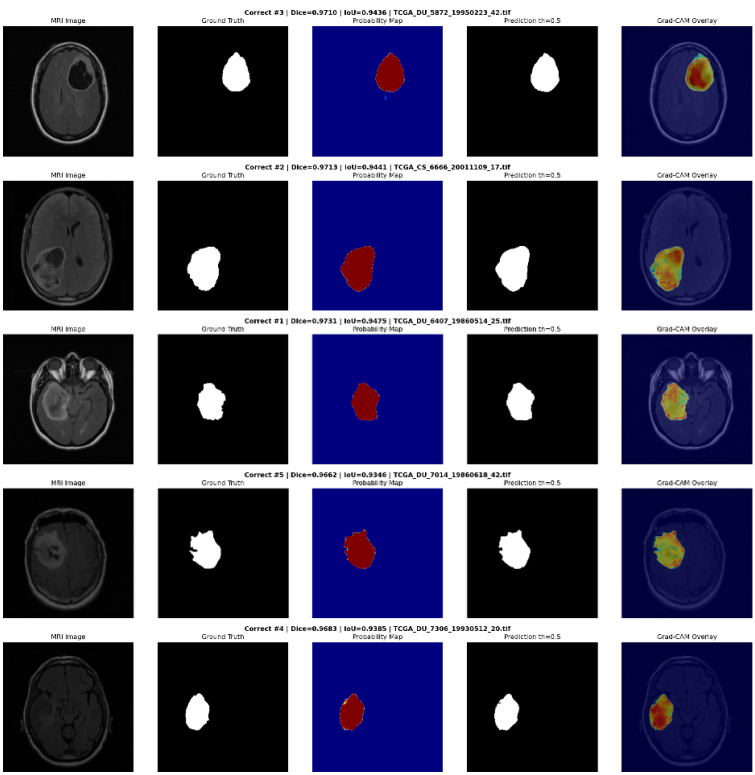
Peningkatan yang paling menonjol pada tabel tersebut justru bukan perbandingan antara kedua *encoder*, melainkan lonjakan nilai *Dice all slices* dari kisaran 0,49 pada Tabel 2.6 menjadi kisaran 0,75 pada Tabel 2.7. Peningkatan sebesar lima puluh persen ini terjadi pada metrik yang sebelumnya menjadi kelemahan utama seluruh model, dan penyebabnya adalah kombinasi *empty ratio*, *DiceBCELoss*, dan pencarian ambang optimal yang diterapkan pada tahap ini. Dengan kata lain, persoalan bercak prediksi palsu pada irisan kosong sebagian besar teratasi bukan melalui pergantian arsitektur, melainkan melalui perbaikan strategi pelatihan. Pelajaran yang dapat ditarik adalah bahwa ketika sebuah model mengalami hambatan, penalaan proses pelatihan sering kali memberikan hasil yang jauh lebih besar daripada penggantian arsitektur.

Membandingkan kedua *encoder* secara langsung, *ResNet50* unggul pada *validation Dice* (0,8070 berbanding 0,8041), *Dice tumor-only* (0,8099 berbanding 0,8023), IoU (0,7183 berbanding 0,7131), dan *recall* (0,8375 berbanding 0,8241). Sebaliknya, *ResNet34* unggul pada *Dice all slices* (0,7496 berbanding 0,7445) dan *precision* (0,8445 berbanding 0,8366). Seluruh selisih tersebut berada pada kisaran satu persen atau kurang, yaitu besaran yang sangat mungkin bersumber dari keacakan inisialisasi maupun pembagian data, sehingga tidak dapat dinyatakan bermakna tanpa pengulangan eksperimen menggunakan beberapa nilai *seed* yang berbeda.

Perbedaan ambang optimal antara kedua model, yaitu 0,40 untuk *ResNet34* dan 0,50 untuk *ResNet50*, merupakan hal yang menarik dan konsisten dengan pola metriknya. *ResNet34* memerlukan ambang yang lebih rendah agar area prediksinya melebar, dan hal ini sejalan dengan sifatnya yang lebih presisi namun kurang sensitif. Atas dasar keunggulannya pada metrik yang berkaitan langsung dengan kualitas penandaan tumor, *Pretrained U-Net ResNet50* dipilih sebagai model final pada bab ini.

Visualisasi Hasil Prediksi yang Benar

Angka evaluasi perlu dilengkapi pengamatan langsung terhadap keluaran model. Gambar 2.3 menampilkan lima sampel dengan hasil segmentasi terbaik, masing-masing memuat citra MRI asli, *ground truth*, *probability map*, hasil prediksi setelah *thresholding*, dan *overlay Grad-CAM*.



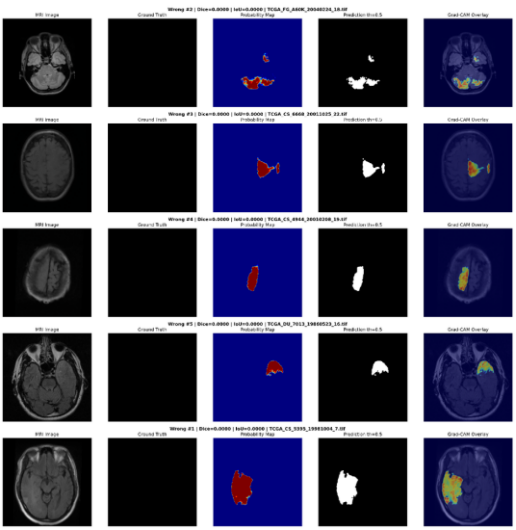
Gambar 2.3 Lima Sampel dengan Hasil Segmentasi Terbaik

Kelima sampel tersebut memperoleh nilai *Dice* antara 0,9662 dan 0,9731 serta IoU antara 0,9346 dan 0,9475, yaitu tingkat kesesuaian yang mendekati sempurna. Bentuk dan lokasi area prediksi hampir seluruhnya berimpit dengan *ground truth*, dengan perbedaan yang hanya tampak pada batas tepi berupa prediksi yang sedikit lebih kecil atau lebih besar dari *mask* sebenarnya. Visualisasi *Grad-CAM* turut memperkuat hasil tersebut karena area aktivasi model terkonsentrasi pada *region* tumor dan bukan pada area lain yang tidak relevan.

Satu hal yang patut diamati pada kolom *probability map* adalah bahwa area tumor tampak berwarna merah pekat dengan batas yang tegas terhadap latar belakang berwarna biru. Ketegasan batas ini menunjukkan bahwa model memberikan probabilitas mendekati satu di dalam area tumor dan mendekati nol di luarnya, sehingga hasil akhirnya tidak sensitif terhadap pemilihan nilai ambang. Kondisi semacam inilah yang menandakan model benar-benar yakin, dan perlu diingat bahwa kelima sampel ini merupakan hasil terbaik, bukan gambaran umum performa model.

Visualisasi Hasil Prediksi yang Salah

Bagian yang justru paling informatif adalah pengamatan terhadap kasus yang gagal. Gambar 2.4 menampilkan lima sampel dengan hasil segmentasi terburuk.



Gambar 2.4 Lima Sampel dengan Hasil Segmentasi Terburuk

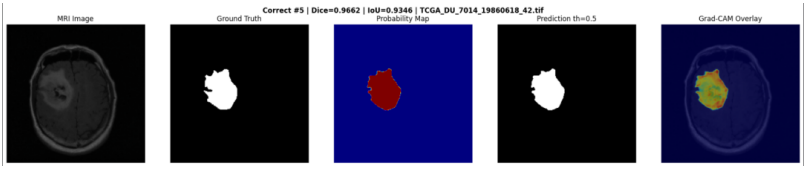
Seluruh sampel pada gambar tersebut memiliki nilai *Dice* dan IoU sebesar nol, dan kolom *ground truth*-nya sepenuhnya hitam. Ini yang sangat penting: kelima kegagalan terburuk model bukanlah kasus tumor yang salah ditandai batasnya, melainkan kasus *false positive* murni pada irisan yang sama sekali tidak mengandung tumor. Nilai *Dice* sebesar nol pada kasus semacam ini bersifat mutlak, karena setiap piksel yang diprediksi sebagai tumor pada irisan kosong langsung menghasilkan *Dice* nol tanpa memandang seberapa kecil areanya.

Pengamatan terhadap kolom citra MRI mengungkap pola yang konsisten: sebagian besar kesalahan terjadi pada irisan di bagian bawah otak yang memuat batang otak, serebelum, dan rongga sinus. Area tersebut memang memiliki intensitas dan tekstur yang menyerupai karakteristik lesi, sehingga model keliru mengenali struktur anatomi normal sebagai tumor. Visualisasi *Grad-CAM* memperlihatkan bahwa model memang memusatkan perhatian pada area tersebut, yang berarti kesalahan ini bukan sekadar derau melainkan keliru mengenali struktur yang secara visual memang ambigu.

Nilai *Dice all slices* tertahan pada kisaran 0,75 meskipun *Dice tumor-only* telah mencapai 0,81. Hambatan yang tersisa bukan terletak pada kemampuan menandai batas tumor, melainkan pada kemampuan mengenali bahwa sebuah irisan memang tidak memuat tumor sama sekali. Persoalan semacam ini sesungguhnya merupakan persoalan klasifikasi, bukan persoalan segmentasi, dan hal itu mengarahkan pada satu saran pengembangan yang akan dibahas pada bagian akhir bab.

Interpretasi Model dengan *Grad-CAM++*

Interpretasi lebih lanjut dilakukan menggunakan *Grad-CAM++* untuk memastikan bahwa keputusan model bertumpu pada informasi yang relevan. Contoh hasil visualisasi ditunjukkan pada Gambar 2.5.



Gambar 2.5 Visualisasi *Grad-CAM++* pada Sampel dengan *Dice* 0,9662

Pada sampel tersebut model menghasilkan *Dice* sebesar 0,9662 dan IoU sebesar 0,9346. *Probability map* memperlihatkan area tumor dengan keyakinan tinggi dan batas yang tegas, sedangkan *overlay Grad-CAM++* memperlihatkan konsentrasi aktivasi yang berimpit dengan lokasi tumor. Kesesuaian antara ketiganya menunjukkan bahwa model tidak hanya benar secara numerik, tetapi juga memusatkan perhatian pada objek yang tepat.

Berbeda dengan Bab 1 yang menemukan indikasi *shortcut learning* melalui aktivasi pada penanda orientasi, pemeriksaan pada bab ini tidak menemukan gejala serupa. Perbedaan tersebut sebagian dapat dijelaskan oleh sifat tugasnya: segmentasi menuntut keluaran yang benar pada setiap piksel sehingga jauh lebih sulit diselesaikan melalui jalan pintas dibandingkan klasifikasi yang hanya menuntut satu keputusan per citra. Meskipun demikian, kesimpulan ini hanya berlaku bagi sampel yang diperiksa dan tidak dapat digeneralisasikan tanpa pemeriksaan yang lebih sistematis.

Estimasi Luas Area dan Tingkat Keparahan

Tahap akhir memanfaatkan *mask* hasil segmentasi untuk menghitung luas area tumor menggunakan Persamaan 2.9 sampai 2.11. Perhitungan dilakukan dalam tiga langkah, yaitu penjumlahan piksel bernilai satu pada *mask*, konversi ke satuan milimeter persegi menggunakan informasi jarak antarpiksel, dan penghitungan rasio terhadap luas penampang otak pada irisan yang sama.

Keandalan angka luas yang dihasilkan sepenuhnya bergantung pada kualitas segmentasi yang mendasarinya, dan hubungan keduanya tidak bersifat linier. Nilai *Dice* sebesar 0,81 tidak berarti kesalahan luasnya sebesar 19%, sebab kesalahan pada dua arah dapat saling meniadakan: sebuah prediksi yang meleset ke satu sisi namun kurang pada sisi lain dapat menghasilkan luas total yang hampir tepat meskipun *Dice*-nya rendah. Sebaliknya, prediksi yang secara konsisten sedikit lebih luas dari *mask* sebenarnya akan menghasilkan taksiran yang bias meskipun *Dice*-nya tinggi. Karena itu penilaian estimasi luas harus dilakukan menggunakan metrik tersendiri, misalnya galat relatif terhadap luas *ground truth*, dan tidak dapat disimpulkan langsung dari nilai *Dice*.

Terdapat pula satu kekeliruan yang perlu dihindari sejak awal. Luas yang dihitung pada satu irisan merupakan besaran dua dimensi, sedangkan tingkat keparahan klinis berkaitan dengan volume tumor yang bersifat tiga dimensi. Konversi menuju volume memerlukan penjumlahan luas seluruh irisan dikalikan ketebalan irisan, dan hal ini menuntut segmentasi yang konsisten pada seluruh volume MRI seorang pasien. Pada tingkat pengembangan yang dibahas pada bab ini, luas per irisan sebaiknya diperlakukan sebagai indikator pendukung, bukan sebagai penentu tingkat keparahan.

Catatan: Ambang Kategori Keparahan
Pemetaan nilai luas atau rasio menjadi kategori seperti ringan, sedang, dan berat merupakan langkah yang tampak sederhana namun sesungguhnya menuntut kehati-hatian tertinggi pada keseluruhan alur kerja ini.
Ambang kategori tidak dapat ditetapkan berdasarkan sebaran data atau pertimbangan statistik semata, melainkan harus berasal dari kesepakatan klinis dan divalidasi terhadap luaran pasien yang sebenarnya. Sebuah sistem yang menyatakan tingkat keparahan menggunakan ambang yang dipilih sendiri oleh perancangnya memberikan kesan objektivitas yang tidak berdasar, dan justru berpotensi lebih berbahaya daripada sistem yang hanya melaporkan angka luas apa adanya.
Sepanjang belum tersedia validasi klinis, praktik yang lebih bertanggung jawab adalah melaporkan nilai luas beserta rentang ketidakpastiannya dan menyerahkan penafsiran tingkat keparahan kepada klinisi.

Pembahasan dan Keterbatasan

Hasil yang diperoleh pada bab ini perlu dibaca bersama sejumlah keterbatasan berikut.

a. Pembagian data pada tingkat irisan. Dataset memuat 3.929 irisan dari hanya 110 pasien, sehingga pembagian acak pada tingkat irisan berpotensi menempatkan irisan yang berdekatan dari pasien yang sama pada subset yang berbeda. Angka evaluasi yang dihasilkan berpeluang lebih optimistis daripada performa sebenarnya.

b. Selisih yang tidak diuji signifikansinya. Perbedaan antara *ResNet*34 dan *ResNet*50 berada pada kisaran satu persen dan hanya diperoleh dari satu kali pelatihan, sehingga belum dapat dibedakan dari keacakan eksperimen.
c. *False positive* pada irisan kosong. Kelima kegagalan terburuk seluruhnya berupa prediksi palsu pada irisan tanpa tumor, dan hal ini masih menjadi hambatan utama nilai *Dice all slices*.
d. Sumber data tunggal. Seluruh citra berasal dari dataset LGG sehingga model belum diuji terhadap variasi karakteristik tumor jenis lain maupun perbedaan protokol akuisisi antarinstitusi.
e. Estimasi luas belum divalidasi. Perhitungan luas belum dibandingkan dengan hasil pengukuran manual oleh radiolog, sehingga tingkat ketelitiannya belum diketahui.
f. Segmentasi dua dimensi. Model bekerja per irisan tanpa memanfaatkan keterkaitan antaririsan, padahal informasi tiga dimensi berpotensi meningkatkan konsistensi segmentasi sekaligus menekan prediksi palsu.

Beberapa arah pengembangan dapat ditempuh untuk mengatasi keterbatasan tersebut. Yang paling langsung menjawab persoalan utama adalah penerapan strategi dua tahap, yaitu menempatkan sebuah pengklasifikasi ringan untuk menentukan keberadaan tumor pada sebuah irisan sebelum model segmentasi dijalankan. Karena kegagalan terburuk seluruhnya berupa prediksi palsu pada irisan kosong, penyaringan semacam ini berpotensi meningkatkan *Dice all slices* secara berarti tanpa mengubah model segmentasinya sama sekali. Arah lain yang dapat ditempuh mencakup perluasan dataset dengan menggabungkan LGG dan BraTS, penggunaan fungsi kerugian yang lebih sensitif terhadap ketidakseimbangan kelas seperti *Focal Tversky loss*, eksplorasi *encoder* berbasis *EfficientNet* maupun arsitektur *Transformer* seperti Swin Transformer dan TransUNet, serta peralihan menuju segmentasi tiga dimensi menggunakan 3D *U-Net*.

➢ Ringkasan

Bab ini telah menguraikan pembangunan sistem segmentasi tumor otak pada citra MRI menggunakan lima varian arsitektur *U-Net*, disertai interpretasi berbasis *Grad-CAM++* dan pemanfaatan hasil segmentasi untuk mengestimasi luas area tumor. Beberapa hal penting dapat dirangkum sebagai berikut.

a. Tidak terdapat satu arsitektur yang unggul pada seluruh metrik. Vanilla *U-Net* paling presisi, Attention *U-Net* paling sensitif, sedangkan *Pretrained U-Net* menunjukkan profil yang paling seimbang sehingga dipilih untuk dikembangkan lebih lanjut.
b. Selisih besar antara *Dice all slices* dan *Dice tumor-only* mengungkap bahwa hambatan utama sistem bukan terletak pada ketepatan penandaan batas tumor, melainkan pada prediksi palsu di irisan yang tidak mengandung tumor.
c. Perbaikan strategi pelatihan berupa *empty ratio*, *DiceBCELoss*, dan pencarian ambang optimal meningkatkan *Dice all slices* dari kisaran 0,49 menjadi 0,75, yaitu peningkatan yang jauh melampaui pengaruh penggantian arsitektur.
d. Model final *Pretrained U-Net ResNet50* mencapai *Dice tumor-only* sebesar 0,8099, IoU sebesar 0,7183, dan *recall* sebesar 0,8375, dengan selisih terhadap *ResNet34* yang terlalu kecil untuk dinyatakan bermakna tanpa pengulangan eksperimen.
e. Kelima kegagalan terburuk seluruhnya berupa *false positive* pada irisan kosong, terutama pada area batang otak dan serebelum yang secara visual menyerupai lesi.
f. Estimasi luas area tumor dapat dihitung langsung dari *mask* segmentasi, tetapi keandalannya tidak dapat disimpulkan dari nilai *Dice* dan penetapan kategori keparahan menuntut validasi klinis tersendiri.

Gagasan yang paling penting untuk dibawa dari bab ini adalah bahwa memilih metrik yang tepat sama pentingnya dengan memilih arsitektur yang tepat. Sebuah model yang tampak baik pada satu cara pembacaan dapat memperlihatkan kelemahan mendasar pada cara pembacaan yang lain, dan hanya dengan melaporkan keduanya secara bersamaan gambaran performa yang sesungguhnya dapat diperoleh.

➢ Daftar Pustaka

Berghout, T. (2025). The Neural Frontier of Future Medical Imaging: A Review of *Deep Learning* for Brain Tumor Detection. Journal of Imaging, 11(1), 2.

Bouhafra, S., & El Bahi, H. (2025). *Deep Learning* Approaches for Brain Tumor Detection and Classification Using MRI Images (2020 to 2024): A Systematic Review. Journal of Imaging Informatics in Medicine, 38(3), 1403–1433.

Buda, M., Saha, A., & Mazurowski, M. A. (2019). Association of Genomic Subtypes of Lower-Grade Gliomas with Shape Features Automatically Extracted by a *Deep Learning* Algorithm. Computers in Biology and Medicine, 109, 218–225.

Chattopadhay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018). *Grad-CAM++*: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. Dalam IEEE Winter Conference on Applications of Computer Vision (WACV), hlm. 839–847.

Chen, X., Wang, X., Zhang, K., Fung, K. M., Thai, T. C., Moore, K., Mannel, R. S., Liu, H., Zheng, B., & Qiu, Y. (2021). Recent Advances and Clinical Applications of *Deep Learning* in Medical Image Analysis. Medical Image Analysis, 79, 102444.

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., & Dean, J. (2021). A Guide to *Deep Learning* in Healthcare. Nature Medicine, 27(1), 29–41.

Horbinski, C., Berger, T., Packer, R. J., & Wen, P. Y. (2022). Clinical Implications of the 2021 Edition of the WHO Classification of Central Nervous System Tumours. Nature Reviews Neurology, 18(9), 515–529.

Li, M., Jiang, Y., Zhang, Y., & Zhu, H. (2023). Medical Image Analysis Using *Deep Learning* Algorithms. Frontiers in Public Health, 11, 1273253.

Louis, D. N., Perry, A., Wesseling, P., Brat, D. J., Cree, I. A., Hawkins, C., Ng, H. K., Pfister, S. M., Reifenberger, G., Soffietti, R., von Deimling, A., & Ellison, D. W. (2021). The 2021 WHO Classification of Tumors of the Central Nervous System: A Summary. Neuro-Oncology, 23(8), 1231–1251.

Makwana, C. (2024). Automated Segmentation of Brain Tumors in MRI: A Comprehensive Review of *Deep Learning* Approaches. Journal of Electrical Systems, 20(7s), 1942–1948.

Milletari, F., Navab, N., & Ahmadi, S. A. (2016). V-Net: Fully *Convolutional Neural Network*s for Volumetric Medical Image Segmentation. Dalam Fourth International Conference on 3D Vision (3DV), hlm. 565–571.

Oktay, O., Schlemper, J., Le Folgoc, L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., Glocker, B., & Rueckert, D. (2018). Attention U-Net: Learning Where to Look for the Pancreas. Dalam Medical Imaging with *Deep Learning* (MIDL).

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Dalam Medical Image Computing and Computer-Assisted Intervention (MICCAI), hlm. 234–241.

Sistaninejhad, B., Rasi, H., & Nayeri, P. (2023). A Review Paper about *Deep Learning* for Medical Image Analysis. Computational and Mathematical Methods in Medicine, 2023, 7091301.

Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., & Liang, J. (2018). UNet++: A *Nested* U-Net Architecture for Medical Image Segmentation. Dalam *Deep Learning* in Medical Image Analysis (DLMIA), hlm. 3–11.

BAB 3
KLASIFIKASI MORFOLOGI *SPERMATOZOA* PADA CITRA MIKROSKOPIS MENGGUNAKAN OPERASI MORFOLOGI DAN *EFFICIENTNET-V2*

Setelah mempelajari bab ini, pembaca diharapkan mampu:

a. Menjelaskan kriteria morfologi *spermatozoa* menurut WHO dan implikasinya terhadap perancangan sistem klasifikasi otomatis.
b. Menerapkan operasi morfologi matematis ambang adaptif, erosi, *opening*, *closing*, dan analisis komponen terhubung untuk mengisolasi objek tunggal dari citra mikroskopis yang padat.
c. Membangun model klasifikasi biner menggunakan *transfer learning* dengan *EfficientNet-V2S*.
d. Menjelaskan cara kerja *Binary Focal Cross-Entropy* dan alasan penggunaannya pada data yang sangat tidak seimbang.
e. Mengevaluasi model secara kritis dengan membandingkannya terhadap baseline kelas mayoritas, serta menilai kecukupan ukuran data uji.
f. Mengenali situasi ketika ketidakseimbangan kelas merupakan sifat alamiah domain, bukan cacat pengumpulan data.

➢ Pendahuluan

Infertilitas didefinisikan sebagai ketidakmampuan pasangan mencapai kehamilan setelah dua belas bulan atau lebih melakukan hubungan seksual secara teratur tanpa kontrasepsi. Kondisi ini tidak hanya berdampak pada aspek medis, tetapi juga menyentuh sisi psikologis, sosial, dan kualitas hidup pasangan yang mengalaminya. World Health Organization melaporkan bahwa sekitar satu dari enam orang pada usia reproduktif di dunia pernah mengalami infertilitas sepanjang hidupnya, atau setara dengan 17,5% populasi dewasa. Faktor pria berkontribusi sekitar setengah dari seluruh kasus, dengan penyebab yang paling sering ditemukan berupa kelainan pada konsentrasi, motilitas, dan morfologi *spermatozoa*. Di Indonesia, Kementerian Kesehatan mencatat bahwa 10–15% pasangan usia subur mengalami kesulitan memperoleh keturunan, yang berarti terdapat empat hingga enam juta pasangan menghadapi persoalan ini.

Di antara ketiga parameter tersebut, bab ini memusatkan perhatian pada morfologi. Menurut WHO, penilaian morfologi mencakup bentuk kepala, bagian tengah, dan ekor *spermatozoa*. Sebuah sel dikategorikan normal apabila memiliki kepala oval dengan ukuran proporsional, bagian tengah yang ramping dan lurus, serta ekor tunggal yang panjang dan tidak melengkung. Peningkatan proporsi sel dengan bentuk menyimpang dikenal sebagai *teratozoospermia*, dan kondisi ini kerap ditemukan bersamaan dengan gangguan motilitas dalam bentuk *asthenoteratozoospermia*. Penilaian morfologi menggunakan kriteria ketat atau *strict criteria* yang dikenal pula sebagai kriteria Kruger, dengan nilai rujukan batas bawah untuk sel berbentuk normal sebesar 4%.

Angka 4% tersebut mengandung konsekuensi yang perlu dipahami sejak awal dan akan berulang kali muncul sepanjang bab ini. Pada sampel semen manusia yang sehat sekalipun, sebagian besar *spermatozoa* memiliki bentuk yang menyimpang dari kriteria ketat. Dengan demikian, ketidakseimbangan antara kelas normal dan abnormal bukanlah cacat pengumpulan data yang perlu diperbaiki, melainkan sifat alamiah dari domain permasalahannya. Setiap keputusan perancangan, mulai dari penanganan ketidakseimbangan kelas hingga pembacaan angka evaluasi, harus mempertimbangkan kenyataan ini.

Evaluasi morfologi di Indonesia hingga kini masih banyak mengandalkan pengamatan mikroskopis manual. Analis laboratorium menilai bentuk setiap sel secara langsung, sebuah proses yang menyita waktu dan hasilnya sangat dipengaruhi oleh pengalaman serta ketelitian pemeriksa, sehingga menimbulkan variabilitas yang tinggi antarpemeriksa. Teknologi *Computer-Aided Sperm Analysis* (CASA) menawarkan jalan keluar berupa analisis otomatis yang lebih cepat dan konsisten, namun penerapannya di Indonesia terkendala harga perangkat yang mahal serta sifat perangkat lunaknya yang tertutup dan tidak dapat disesuaikan dengan kebutuhan laboratorium setempat.

Dari sudut pandang teknis, permasalahan pada bab ini berbeda secara mendasar dari dua bab sebelumnya. Pada Bab 1, objek yang dianalisis adalah paru-paru yang menempati sebagian besar bidang citra. Pada Bab 2, objek berupa tumor yang perlu ditandai batasnya namun tetap berada pada satu citra utuh. Pada bab ini, objek yang

dianalisis adalah sel tunggal berukuran beberapa mikrometer yang berada di antara puluhan sel serupa, saling bertumpang tindih, dan bergerak. Sebelum klasifikasi dapat dilakukan, setiap sel harus terlebih dahulu ditemukan, dipotong dari bidang pandang, dan dipisahkan dari tetangganya. Tahap penyiapan citra karena itu bukan sekadar pelengkap, melainkan penentu keberhasilan keseluruhan sistem.

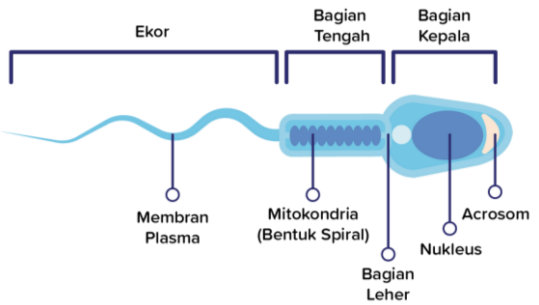
Bab ini menuntun pembaca membangun jalur klasifikasi morfologi *spermatozoa* secara utuh, mulai dari akuisisi video mikroskopis, standardisasi dan peningkatan kualitas citra, pemotongan *Region of Interest*, isolasi objek menggunakan rangkaian operasi morfologi matematis, hingga klasifikasi biner normal dan abnormal menggunakan *EfficientNet-V2S* melalui pendekatan *transfer learning*. Bab ini juga membahas integrasi hasil ke dalam antarmuka berbasis web serta, yang tidak kalah penting, cara membaca angka akurasi sebesar 97% secara kritis pada data uji yang sangat tidak seimbang.

➢ Pengantar Teori

Bagian ini menguraikan landasan teori yang diperlukan pada bab ini. Konsep dasar mengenai konvolusi, *transfer learning*, dan metrik evaluasi telah dibahas pada Bab 1, sehingga pembahasan berikut difokuskan pada hal-hal yang khas bagi permasalahan isolasi dan klasifikasi objek mikroskopis.

Morfologi *Spermatozoa* dan Kriteria WHO

Spermatozoa merupakan sel reproduktif jantan yang bertugas membuahi sel telur. Struktur selnya terdiri atas tiga bagian utama sebagaimana ditunjukkan pada Gambar 3.1.



Gambar 3.1 Struktur Bagian *Spermatozoa*

Bagian kepala berbentuk oval dengan panjang sekitar 3–5 µm dan lebar 2–3 µm, memuat inti sel serta struktur akrosom yang mengandung enzim untuk menembus zona pelusida sel telur. Bagian tengah atau *midpiece* mengandung mikrotubula dengan susunan 9+2 dan dikelilingi mitokondria yang menyediakan energi bagi pergerakan. Ekor atau flagelum berfungsi sebagai alat gerak menuju sel telur. Kelainan pada salah satu dari ketiga bagian tersebut dapat menurunkan kapasitas fertilisasi secara bermakna.

Ukuran kepala yang hanya beberapa mikrometer merupakan angka yang perlu diingat ketika membaca bagian metodologi. Pada perbesaran 40× dengan resolusi video 512 × 512 piksel, seluruh struktur kepala sepanjang 3–5 µm hanya terwakili oleh beberapa belas piksel. Artinya, ruang antara detail yang membedakan bentuk normal dari abnormal dengan tingkat derau sensor kamera sangatlah sempit, dan setiap operasi pengolahan citra yang diterapkan berpotensi menghapus informasi yang justru menjadi dasar penilaian.

Peningkatan Kontras

Citra mikroskopis lapangan terang umumnya memiliki rentang intensitas yang sempit karena *spermatozoa* hampir transparan dan hanya sedikit lebih gelap daripada mediumnya. *Contrast stretching* memperlebar rentang tersebut dengan memetakan persentil intensitas tertentu ke rentang penuh nol sampai 255.

$$I'(x,y) = clip\left(\frac{I(x,y) - p_2}{p_{98} - p_2} \times 255, 0, 255\right) \quad (3.1)$$

Dengan p_2 dan p_{98} berturut-turut menyatakan persentil kedua dan kesembilan puluh delapan dari distribusi intensitas citra. Penggunaan persentil, bukan nilai minimum dan maksimum, membuat proses ini tahan terhadap piksel pencilan akibat pantulan cahaya maupun titik derau. Fungsi *clip* memastikan hasilnya tetap berada pada rentang yang sah setelah pemetaan.

Ambang Adaptif

Pengubahan citra keabuan menjadi citra biner memerlukan penentuan nilai ambang. Penggunaan satu nilai ambang global tidak sesuai untuk citra mikroskopis karena pencahayaan pada bidang pandang mikroskop hampir tidak pernah merata; bagian tengah

umumnya lebih terang daripada tepi. Ambang adaptif mengatasi hal ini dengan menghitung ambang tersendiri bagi setiap piksel berdasarkan rerata terbobot lingkungan di sekitarnya.

$$T(x,y) = \sum_{(i,j) \in N(x,y)} G(i,j) \cdot I(i,j) - C \quad (3.2)$$

Dengan N(x, y) menyatakan jendela ketetanggaan berukuran tertentu di sekitar piksel yang sedang diproses, G menyatakan bobot *Gaussian*, dan C menyatakan konstanta penyesuai. Sebuah piksel ditetapkan sebagai bagian objek apabila intensitasnya berada di bawah T(x, y), mengingat *spermatozoa* tampak lebih gelap daripada latar belakangnya.

Operasi Morfologi Matematis

Operasi morfologi merupakan teknik pengolahan citra yang memanipulasi struktur geometris objek melalui pendekatan himpunan dan sebuah matriks kecil yang disebut *structuring element*. Operasi yang paling mendasar adalah erosi, yang mengikis batas objek latar depan.

$$A \ominus B = \{ z \mid B_z \subseteq A \} \quad (3.3)$$

Dengan A menyatakan citra biner masukan, B menyatakan *structuring element*, dan B beraksen bawah z menyatakan hasil pergeseran B sehingga titik pusatnya berada pada posisi z. Persamaan tersebut menyatakan bahwa hasil erosi adalah himpunan semua titik z yang membuat *structuring element* sepenuhnya termuat di dalam objek. Secara praktis, sebuah piksel dipertahankan hanya apabila seluruh lingkungan yang tertutup *structuring element* juga merupakan bagian objek; apabila salah satu bagiannya menyentuh latar belakang, piksel tersebut dikikis. Akibatnya objek menyusut dan detail yang lebih kecil daripada *structuring element* menghilang sepenuhnya.

Operasi pasangannya adalah dilasi, yang memperluas objek dengan cara sebaliknya. Pada persamaan berikut, B beraksen prima menyatakan pencerminan *structuring element* terhadap titik pusatnya.

$$A \oplus B = \{ z \mid (B'_z \cap A) \neq \emptyset \} \quad (3.4)$$

Kombinasi keduanya menghasilkan dua operasi turunan yang justru paling banyak digunakan pada praktik. *Opening* adalah erosi yang diikuti dilasi, dan berfungsi menghapus derau berukuran kecil tanpa mengubah ukuran objek utama secara berarti.

$$A \circ B = (A \ominus B) \oplus B \quad (3.5)$$

Sebaliknya, *closing* adalah dilasi yang diikuti erosi, dan berfungsi menutup lubang maupun celah sempit di dalam objek.

$$A \bullet B = (A \oplus B) \ominus B \quad (3.6)$$

Catatan: Erosi Bersifat Merusak
Erosi digunakan pada bab ini untuk memutus sambungan tipis antara *spermatozoa* target dan sel lain yang bersinggungan dengannya. Namun perlu disadari bahwa operasi ini tidak dapat membedakan sambungan yang ingin diputus dari struktur asli yang ingin dipertahankan.
Ekor *spermatozoa* merupakan struktur yang sangat tipis, dan pada resolusi yang tersedia lebarnya hanya beberapa piksel. Erosi dengan *structuring element* yang terlalu besar akan memutus ekor tersebut sama seperti ia memutus sambungan antarsel. Padahal ekor yang bengkok atau ganda justru merupakan salah satu penanda utama abnormalitas.
Pada alur kerja bab ini, risiko tersebut ditekan melalui pengisian kontur dan operasi *closing* setelah objek target dipilih. Meskipun demikian, pemilihan ukuran *structuring element* tetap merupakan pertukaran antara kemampuan memisahkan sel yang berhimpit dan risiko menghapus informasi diagnostik.

Analisis Komponen Terhubung

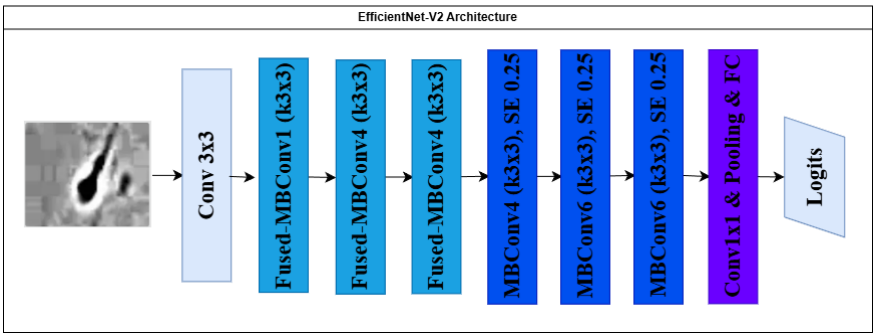
Setelah citra dibersihkan, umumnya masih terdapat lebih dari satu objek pada bidang potong. Analisis komponen terhubung (*connected component analysis*) memberi label berbeda pada setiap kelompok piksel yang saling bersambung, sehingga masing-masing objek dapat diperlakukan secara terpisah. Karena pemotongan dilakukan dengan menempatkan *spermatozoa* target tepat di titik pusat, objek yang dicari dapat ditentukan sebagai komponen dengan pusat massa terdekat terhadap pusat citra.

$$k^* = arg\,min\sqrt{(c_{x,k} - c_{x,0})^2 + (c_{y,k} - c_{y,0})^2} \quad (3.7)$$

Dengan c indeks k menyatakan pusat massa komponen ke-k, c indeks 0 menyatakan titik pusat citra, dan k bintang menyatakan indeks komponen yang terpilih sebagai objek target. Pendekatan ini sederhana dan efektif, tetapi mengandung satu asumsi yang perlu diwaspadai: apabila deteksi partikel pada tahap sebelumnya meleset sehingga sel target tidak benar-benar berada di tengah, aturan ini akan memilih sel yang keliru tanpa memberikan tanda kegagalan apa pun.

EfficientNet-V2

EfficientNet merupakan keluarga arsitektur CNN yang memperkenalkan *compound scaling*, yaitu penskalaan serentak terhadap kedalaman, lebar, dan resolusi sebagaimana telah dibahas pada Bab 1 melalui Persamaan 1.4. *EfficientNet-V2* merupakan pengembangannya yang versi terdahulu memang hemat parameter, namun lambat dilatih. Penyebab utamanya adalah penggunaan *depthwise convolution* pada lapisan-lapisan awal yang ternyata kurang optimal bagi akselerator perangkat keras modern, serta ukuran citra masukan yang besar sehingga memicu konsumsi memori tinggi.



Gambar 3.2 Arsitektur *EfficientNet-V2*

Sebagaimana ditunjukkan pada Gambar 3.2, arsitektur ini tersusun dari dua jenis blok utama. Blok MBConv (*Mobile Inverted Bottleneck Convolution*) memuat tahap ekspansi kanal melalui konvolusi 1×1, *depthwise convolution* 3×3, modul *squeeze-and-excitation* yang memberi bobot adaptif pada setiap kanal, dan tahap

reduksi kembali melalui konvolusi 1×1. Blok Fused-MBConv menggantikan pemisahan antara konvolusi 1×1 dan *depthwise* 3×3 dengan satu konvolusi reguler 3×3. Meskipun jumlah operasi aritmetikanya sedikit lebih besar, blok ini jauh lebih cepat dieksekusi karena mengurangi *overhead* akses memori. Penempatannya pada tahap-tahap awal jaringan merupakan inovasi utama *EfficientNet-V2*.

Fungsi aktivasi yang digunakan secara konsisten adalah SiLU (*Sigmoid Linear Unit*), yang juga dikenal sebagai Swish.

$$f(x) = x \cdot \sigma(x) = \frac{x}{1 + e^{-x}} \quad (3.8)$$

Berbeda dengan ReLU yang memiliki titik patah tajam pada nol, SiLU menghasilkan transisi gradien yang lebih halus dan memungkinkan sebagian kecil nilai negatif tetap diteruskan. Sifat ini membantu kestabilan konvergensi pada jaringan yang dalam. Varian yang digunakan pada bab ini adalah *EfficientNet-V2S*, yaitu varian berukuran kecil pada keluarga tersebut.

Binary Focal Cross-Entropy

Pada data yang sangat tidak seimbang, *binary cross-entropy* biasa memiliki kelemahan yang sudah dibahas pada Bab 2: kontribusi sampel kelas mayoritas menenggelamkan kontribusi kelas minoritas. Persoalan tambahan pada kasus ini adalah bahwa sebagian besar sampel mayoritas sesungguhnya mudah diklasifikasikan, sehingga model menghabiskan sebagian besar kapasitas belajarnya untuk mempertegas keputusan yang sudah benar. *Focal loss* mengatasi hal ini dengan menambahkan faktor modulasi yang menekan kontribusi sampel yang sudah dikenali dengan baik.

$$L_{FL} = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \quad (3.9)$$

Dengan p indeks t menyatakan probabilitas yang diberikan model bagi kelas yang benar, γ menyatakan parameter pemfokusan, dan α indeks t menyatakan bobot penyeimbang kelas. Cara kerjanya dapat dipahami melalui contoh sederhana. Apabila model sudah sangat yakin dan benar dengan p indeks t sebesar 0,95 dan digunakan γ sebesar 2, maka faktor modulasi bernilai $(1 - 0{,}95)^2 = 0{,}0025$, sehingga kontribusi sampel tersebut ditekan hingga tinggal seperempat persen. Sebaliknya, sampel sulit dengan p indeks t

sebesar 0,3 memperoleh faktor sebesar 0,49 sehingga kontribusinya nyaris utuh. Dengan demikian proses pelatihan secara otomatis memusatkan perhatian pada kasus batas yang justru paling menentukan.

Pada bab ini digunakan γ sebesar 2,0 dan α sebesar 0,75. Nilai α tersebut memberikan bobot lebih besar kepada kelas positif, yang perlu diperhatikan penetapannya secara cermat agar sejalan dengan kelas mana yang sesungguhnya merupakan minoritas pada dataset.

Metrik Evaluasi pada Data Sangat Tidak Seimbang

Definisi *accuracy*, *precision*, *recall*, dan *F1-score* telah diuraikan pada Bab 1 dan tidak diulang di sini. Yang perlu ditambahkan adalah satu pembanding sederhana yang wajib dilaporkan setiap kali data uji sangat tidak seimbang, yaitu akurasi yang dicapai oleh sebuah model tanpa kecerdasan sama sekali yang selalu memprediksi kelas mayoritas.

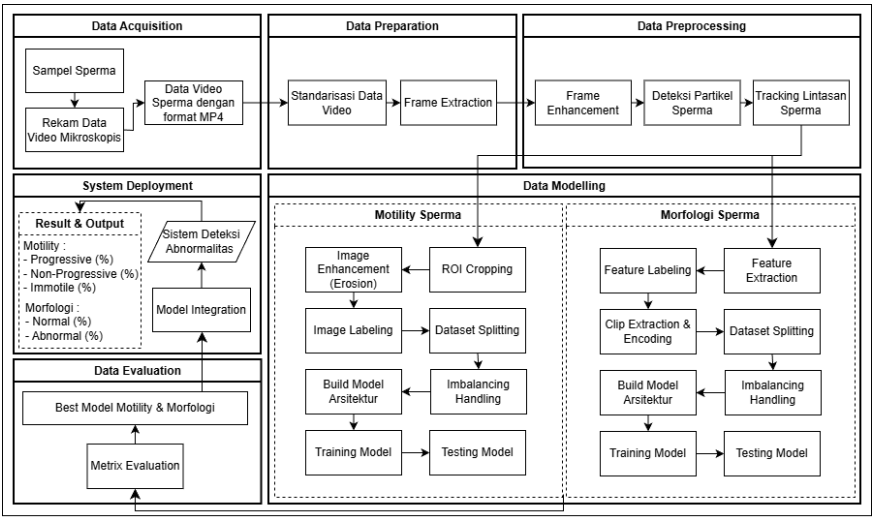
$$Acc_{baseline} = \frac{n}{N} \qquad (3.10)$$

Dengan n menyatakan jumlah sampel kelas mayoritas dan N menyatakan jumlah seluruh sampel uji. Nilai inilah yang menjadi titik nol sesungguhnya bagi penilaian sebuah model. Sebuah akurasi sebesar 97% terdengar mengesankan, tetapi apabila *baseline* kelas mayoritasnya sudah mencapai 94%, maka sumbangan model yang sebenarnya hanya sebesar tiga poin persentase. Perbandingan ini akan digunakan secara langsung pada Subbab Membaca Angka 97% Secara Kritis.

➢ Metodologi

Alur Metodologi

Sistem yang dibahas pada bab ini merupakan bagian dari kerangka kerja yang lebih besar dan menganalisis dua aspek sekaligus, yaitu motilitas melalui 3D-CNN dan morfologi melalui *EfficientNet-V2*. Keduanya berbagi satu jalur praproses yang sama sebelum bercabang menjadi dua jalur pemodelan paralel sebagaimana ditunjukkan pada Gambar 3.3. Bab ini menelusuri cabang morfologi secara mendalam, sedangkan cabang motilitas hanya dibahas seperlunya sebagai konteks.



Gambar 3.3 Diagram Alir Metodologi Deteksi Abnormalitas *Spermatozoa*

Proses diawali dengan akuisisi video mikroskopis, dilanjutkan dengan standardisasi format dan ekstraksi *frame*. Tahap praproses meliputi peningkatan kualitas citra, deteksi partikel, dan pelacakan lintasan. Keluaran tahap ini bercabang: koordinat lintasan digunakan pada jalur motilitas, sedangkan posisi partikel digunakan pada jalur morfologi sebagai titik acuan pemotongan *Region of Interest*. Jalur morfologi kemudian melanjutkan dengan peningkatan citra berbasis erosi, pelabelan, pembagian data, penanganan ketidakseimbangan kelas, pembangunan model, pelatihan, dan pengujian. Kedua jalur akhirnya bertemu kembali pada tahap integrasi sistem.

Akuisisi Data

Data berupa video mikroskopis *spermatozoa* yang diperoleh melalui kerja sama dengan Fakultas Kedokteran UPN "Veteran" Jawa Timur. Perekaman dilakukan menggunakan mikroskop binokuler OptiLab tipe IRIS-5 pada perbesaran 40× yang terhubung dengan kamera digital. Video hasil perekaman merekam pergerakan *spermatozoa* secara langsung dalam medium pengamatan. Karakteristik data mentah yang diperoleh sangat beragam sebagaimana dirangkum pada Tabel 3.1.

Tabel 3.1 Karakteristik Data Video Mikroskopis Hasil Akuisisi

Aspek	**Rentang Nilai pada Data Mentah**
Format berkas	MP4 dan AVI
Durasi	0,60 – 9,72 detik
Jumlah frame	19 – 583 frame
Frame rate	30 FPS dan 60 FPS
Resolusi	698 × 528 dan 1280 × 960 piksel

Keberagaman tersebut merupakan gambaran yang jujur mengenai kondisi data yang diperoleh dari laboratorium nyata, dan berbeda jauh dari dataset publik yang sudah terstandar seperti pada dua bab sebelumnya. Rentang durasi yang membentang dari kurang dari satu detik hingga hampir sepuluh detik berimplikasi langsung pada jumlah sampel yang dapat diperoleh dari setiap video, sedangkan perbedaan resolusi berarti satu mikrometer pada dunia nyata tidak terwakili oleh jumlah piksel yang sama antarvideo. Standardisasi karena itu bukan sekadar langkah kerapian, melainkan syarat agar sampel dari berbagai video dapat diperlakukan setara.

Standardisasi dan Peningkatan Kualitas Video

Standardisasi dilakukan menggunakan FFmpeg dengan menyeragamkan seluruh video ke *frame rate* 60 FPS dan resolusi 512 × 512 piksel. Penyeragaman *frame rate* menjaga konsistensi informasi temporal, sedangkan penyesuaian resolusi memastikan keseragaman ukuran spasial sekaligus menekan beban komputasi.

```
TARGET_FPS = 60
TARGET_SIZE = "512:512"

command = [
    "ffmpeg", "-y", "-i", input_path,
    "-r", str(TARGET_FPS),
    "-vf", f"scale={TARGET_SIZE}",
    "-c:v", "libx264", "-preset", "slow", "-crf", "18",
    output_path
]
```

Kode 3.1 Standardisasi Video Menggunakan FFmpeg

Video yang telah diseragamkan kemudian dikonversi menjadi citra keabuan untuk menyederhanakan representasi, lalu ditingkatkan kontrasnya menggunakan Persamaan 3.1. Implementasi peningkatan kontras ditunjukkan pada Kode 3.2.

```
def contrast_stretch(img):
    p2, p98 = np.percentile(img, (2, 98))
    return np.clip((img - p2) * 255.0 / (p98 - p2),
                   0, 255).astype(np.uint8)
```

Kode 3.2 Peningkatan Kontras dengan Pemetaan Persentil

Perlu dicatat bahwa penyeragaman resolusi menjadi 512 × 512 piksel dilakukan tanpa mempertahankan rasio aspek asli. Video beresolusi 698 × 528 dan 1280 × 960 keduanya memiliki rasio aspek sekitar 4:3, sehingga pemaksaan ke bentuk bujur sangkar menimbulkan peregangan pada sumbu vertikal. Pada tugas klasifikasi bentuk, distorsi semacam ini secara prinsip dapat memengaruhi penilaian proporsi kepala terhadap ekor. Karena seluruh data mengalami distorsi yang sama, dampaknya terhadap konsistensi internal dapat diabaikan; namun hal ini menjadi persoalan apabila model kelak diterapkan pada video dengan rasio aspek yang berbeda.

Pemotongan *Region of Interest*

Posisi setiap *spermatozoa* diperoleh dari tahap deteksi partikel menggunakan pustaka *TrackPy*, yang mengidentifikasi objek berdasarkan ukuran dan intensitas piksel serta menghitung pusat massanya dengan ketelitian melampaui satu piksel. Untuk keperluan analisis morfologi, setiap partikel diwakili oleh satu *frame* saja, yaitu *frame* dengan nilai *signal* tertinggi yang merepresentasikan kualitas visual terbaik. Pemotongan dilakukan secara simetris di sekitar koordinat pusat partikel, kemudian hasilnya diubah ukurannya menjadi seragam.

```
best_frames = (df.sort_values("signal", ascending=False)
               .groupby("particle").first().reset_index())

half = CROP_SIZE // 2
x1, y1 = max(0, x - half), max(0, y - half)
```

```
x2, y2 = min(w, x + half), min(h, y + half)
crop = img[y1:y2, x1:x2]
crop_resized = cv2.resize(crop, (RESIZE_TO, RESIZE_TO))
```

Kode 3.3 Pemilihan *Frame* Terbaik dan Pemotongan ROI

Pemilihan *frame* berdasarkan nilai *signal* tertinggi merupakan keputusan yang masuk akal karena menghindari citra yang buram atau tidak fokus. Namun keputusan ini sekaligus memperkenalkan bias seleksi yang perlu disadari: sel yang sepanjang perekaman selalu tampil samar, misalnya karena berada di luar bidang fokus atau tertutup sel lain, akan diwakili oleh citra terbaiknya yang tetap berkualitas rendah. Sebaliknya, tidak ada mekanisme yang menyingkirkan sel semacam itu dari dataset. Akibatnya sebagian sampel yang masuk ke tahap pelabelan sesungguhnya tidak memuat informasi yang cukup untuk dinilai bentuknya.

Isolasi Objek Menggunakan Operasi Morfologi

Citra hasil pemotongan belum siap digunakan karena masih memuat derau latar belakang serta sebagian tubuh sel lain yang berada di sekitar sel target. Rangkaian operasi morfologi diterapkan untuk memperoleh siluet sel target yang bersih dan utuh, melalui tujuh langkah berurutan sebagai berikut.

a. Binerisasi dengan ambang adaptif. Citra keabuan diubah menjadi biner menggunakan Persamaan 3.2 sehingga tahan terhadap ketidakrataan pencahayaan mikroskop.
b. Erosi. Diterapkan satu iterasi erosi sesuai Persamaan 3.3 untuk memutus sambungan tipis antara sel target dan sel lain yang bersinggungan.
c. *Opening*. Operasi ringan sesuai Persamaan 3.5 diterapkan untuk menghapus bercak derau berukuran kecil.
d. Analisis komponen terhubung. Seluruh objek yang tersisa diberi label terpisah.
e. Pemilihan objek target. Komponen dengan pusat massa terdekat terhadap pusat citra dipilih menggunakan Persamaan 3.7, dan komponen lainnya dibuang.
f. Pengisian kontur. Bagian dalam objek terpilih diisi penuh melalui ekstraksi kontur sehingga menghasilkan siluet yang solid.

g. *Closing* dan *inversi.* Operasi sesuai Persamaan 3.6 merapikan tepi dan menutup celah kecil, kemudian warna dibalik sehingga sel tampil gelap dengan latar belakang terang.

```
binary = cv2.adaptiveThreshold(img, 255,
   cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
   cv2.THRESH_BINARY_INV, 11, 2)

binary = cv2.erode(binary, KERNEL_ERODE, iterations=1)
binary = cv2.morphologyEx(binary, cv2.MORPH_OPEN,
KERNEL_OPEN)

num_labels, labels, stats, centroids = \
   cv2.connectedComponentsWithStats(binary)

# pilih komponen dengan centroid terdekat ke pusat citra
target_label = min(range(1, num_labels),
   key=lambda i: np.hypot(centroids[i][0] - cx_img,
                          centroids[i][1] - cy_img))

mask = np.zeros_like(binary)
mask[labels == target_label] = 255
contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL,
                               cv2.CHAIN_APPROX_SIMPLE)
filled = np.zeros_like(mask)
cv2.drawContours(filled, contours, -1, 255, thickness=-1)
filled = cv2.morphologyEx(filled, cv2.MORPH_CLOSE,
KERNEL_CLOSE)
final = 255 - filled
```

Kode 3.4 Rangkaian Operasi Morfologi untuk Isolasi Objek

Pelabelan Morfologi

Pelabelan dilakukan melalui pengamatan visual terhadap setiap citra hasil isolasi, dengan mengelompokkan *spermatozoa* ke dalam kategori normal dan abnormal. Kriteria penilaian mencakup keutuhan bentuk kepala, proporsi ukuran kepala terhadap ekor, serta keberadaan kelainan seperti kepala yang membulat tidak sempurna, ekor bengkok, atau struktur yang terputus.

Perhatian: Batas Atas Kualitas yang Ditentukan oleh Label
Terdapat satu ketegangan mendasar pada pendekatan ini yang perlu dinyatakan secara terbuka. Sistem ini dibangun untuk mengatasi subjektivitas penilaian manual, namun *ground truth* yang digunakan untuk melatihnya berasal dari penilaian visual manual itu sendiri.
Konsekuensinya, model tidak dapat menjadi lebih objektif daripada label yang mengajarinya; yang dicapainya adalah konsistensi, yaitu kemampuan menerapkan penilaian satu orang pengamat secara seragam pada data baru. Ini tetap merupakan manfaat nyata, sebab pengamat manusia dapat memberi penilaian berbeda pada citra yang sama di waktu yang berbeda, sedangkan model tidak.
Praktik yang lebih kuat menuntut pelabelan oleh sekurang-kurangnya dua analis secara independen, disertai pelaporan tingkat kesepakatan antarpengamat, misalnya melalui koefisien Cohen kappa. Angka tersebut menetapkan batas atas yang wajar bagi performa model: apabila dua ahli hanya sepakat pada 80% kasus, maka akurasi model di atas 80% terhadap label salah satu di antaranya patut dicurigai sebagai gejala *overfitting* terhadap kebiasaan menilai orang tersebut.

Pembagian Data dan Penanganan Ketidakseimbangan

Dataset dibagi menjadi data pelatihan, validasi, dan pengujian dengan skema yang sama seperti pada jalur motilitas. Karena jumlah citra kelas normal jauh lebih sedikit daripada kelas abnormal, diterapkan augmentasi pada kelas minoritas hingga jumlahnya setara dengan kelas mayoritas.

```
augmentor = ImageDataGenerator(
    rotation_range=20,
    zoom_range=0.2,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True)

while len(os.listdir(minority_dir)) < len(majority_files):
    img = cv2.imread(os.path.join(minority_dir,
```

```
            random.choice(minority_files)))
    img = cv2.resize(img, (224, 224))
    aug_img = next(augmentor.flow(np.expand_dims(img, 0),
                    batch_size=1))[0]
    cv2.imwrite(save_path, aug_img)
```

Kode 3.5 Penyeimbangan Kelas Melalui Augmentasi

Transformasi yang dipilih meliputi rotasi hingga 20 derajat, perbesaran, pergeseran, dan pembalikan horizontal. Seluruhnya merupakan transformasi yang sah pada konteks ini, sebab *spermatozoa* pada preparat dapat berada pada orientasi mana pun sehingga rotasi maupun pembalikan tetap menghasilkan citra yang mungkin ditemui di laboratorium. Perlu diperhatikan bahwa perbesaran sebesar 20% mengubah ukuran objek, padahal ukuran absolut kepala merupakan salah satu kriteria penilaian. Selama perbandingan proporsi antara kepala dan ekor tetap terjaga, dampaknya terbatas; namun hal ini menjadi alasan agar rentang perbesaran tidak diperbesar lebih jauh.

Satu hal yang lebih mendasar perlu ditambahkan. Menyeimbangkan kelas hingga sama banyak berarti menyampaikan kepada model bahwa peluang awal sebuah sel berbentuk normal adalah 50%, padahal secara biologis peluang itu berada di bawah 10%. Bagi tugas klasifikasi per sel, distorsi prior semacam ini merupakan pertukaran yang lazim diterima. Namun apabila keluaran model kelak digunakan untuk menghitung persentase sel normal pada suatu sampel, angka yang dihasilkan berpotensi bias ke atas dan perlu dikalibrasi ulang terhadap distribusi sebenarnya.

Arsitektur Model dan Konfigurasi Pelatihan

Model dibangun menggunakan pendekatan *transfer learning* dengan *EfficientNet-V2S* sebagai *base model*. Lapisan klasifikasi bawaannya dilepas, seluruh bobot *base model* dibekukan, dan di atasnya ditambahkan kepala klasifikasi baru berupa *global average pooling*, lapisan terhubung penuh dengan 128 neuron beraktivasi ReLU, *dropout* sebesar 0,4, serta satu neuron keluaran beraktivasi sigmoid.

```
base_model = EfficientNetV2S(include_top=False,
                    weights="imagenet",
                    input_shape=(224, 224, 3))
```

```
base_model.trainable = False

x = GlobalAveragePooling2D()(base_model.output)
x = Dense(128, activation="relu")(x)
x = Dropout(0.4)(x)
output = Dense(1, activation="sigmoid")(x)
model = Model(inputs=base_model.input, outputs=output)

loss_fn = tf.keras.losses.BinaryFocalCrossentropy(
   gamma=2.0, alpha=0.75)
model.compile(optimizer=tf.keras.optimizers.Adam(1e-4),
          loss=loss_fn,
          metrics=["accuracy",
                tf.keras.metrics.Recall(name="recall")])
```

Kode 3.6 Pembangunan Model *EfficientNet-V2S* dengan Focal *Loss*

Konfigurasi pelatihan selengkapnya dirangkum pada Tabel 3.2.

Tabel 3.2 Konfigurasi Pelatihan Model Morfologi

Komponen	**Konfigurasi**	**Alasan Pemilihan**
Arsitektur dasar	*EfficientNet-V2S*	Efisien secara parameter dan cepat dilatih
Bobot awal	*ImageNet*	Memanfaatkan representasi visual umum
Status base model	Dibekukan	Mencegah *overfitting* pada data terbatas
Ukuran citra	224 × 224 piksel	Sesuai masukan bawaan arsitektur
Kepala klasifikasi	GAP → Dense 128 → *Dropout* 0,4	Ringan dan disertai regularisasi
Aktivasi keluaran	Sigmoid, satu neuron	Klasifikasi biner normal dan abnormal
Fungsi kerugian	*Binary Focal Cross-Entropy*	Memfokuskan pelatihan pada kasus sulit
Parameter focal *loss*	γ = 2,0 dan α = 0,75	Menekan kontribusi sampel mudah

Optimizer	Adam, *learning rate* 1e-4	Konvergensi stabil pada kepala klasifikasi
Jumlah *epoch*	200	Ditetapkan tetap tanpa *early stopping*
Augmentasi	Rotasi, zoom, shift, horizontal flip	Menyeimbangkan kelas minoritas

Terdapat satu perbedaan penting antara konfigurasi ini dan yang digunakan pada dua bab sebelumnya, yaitu tidak diterapkannya *early stopping*. Pelatihan dijalankan hingga *epoch* ke-200 secara penuh. Konsekuensinya akan terlihat langsung pada kurva pelatihan yang dibahas pada subbab berikutnya.

➢ Hasil dan Ujicoba

Hasil Isolasi Objek

Keberhasilan keseluruhan sistem bertumpu pada kualitas citra yang dihasilkan tahap isolasi objek, sehingga bagian ini dibahas terlebih dahulu. Gambar 3.4 menampilkan contoh citra hasil pemotongan *Region of Interest* sebelum operasi morfologi diterapkan.



Gambar 3.4 Citra Morfologi *Spermatozoa* Hasil Pemotongan ROI

Ketiga contoh tersebut memperlihatkan persoalan yang nyata. Sel target memang berada di tengah bidang, tetapi latar belakangnya penuh dengan variasi intensitas, dan pada contoh kedua maupun ketiga terdapat sel lain yang masuk ke dalam bidang potong. Halo terang yang mengelilingi setiap sel merupakan artefak khas mikroskopi lapangan terang yang diperkuat oleh tahap peningkatan kontras sebelumnya. Sebuah model yang dilatih langsung pada citra semacam ini berisiko mempelajari pola latar belakang, bukan bentuk

sel. Gambar 3.5 menampilkan hasil setelah rangkaian operasi morfologi diterapkan.



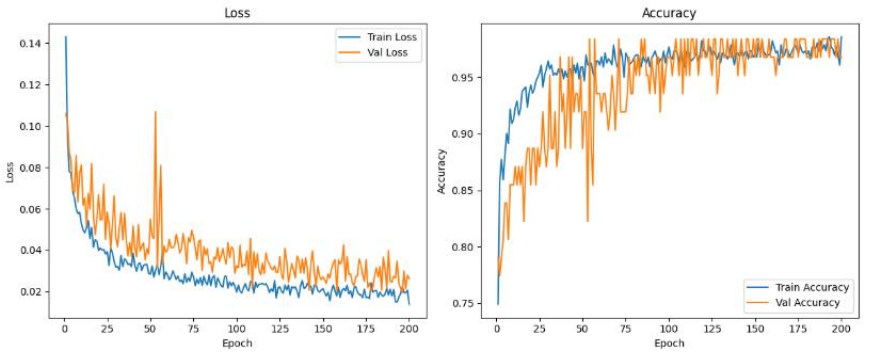
Gambar 3.5 Citra Morfologi *Spermatozoa* Setelah Operasi Morfologi

Perubahannya menyeluruh. Latar belakang menjadi putih bersih, sel lain telah tersingkir, dan yang tersisa hanyalah siluet sel target berwarna hitam pekat. Pada contoh kedua, kepala yang membesar disertai ekor yang melengkung tampak sangat jelas dan mudah dinilai. Perlu diperhatikan bahwa keluaran tahap ini merupakan siluet biner, bukan citra keabuan. Seluruh informasi tekstur internal, termasuk struktur akrosom yang turut menjadi bagian penilaian pada kriteria ketat WHO, telah hilang sepenuhnya. Yang tersisa bagi model hanyalah informasi bentuk luar, yaitu kontur, proporsi, dan kelengkungan.

Keputusan ini merupakan pertukaran yang disengaja dan konsekuensinya berlangsung dua arah. Di satu sisi, penyederhanaan menjadi siluet menghilangkan variasi pencahayaan antarvideo sehingga model tidak dapat memanfaatkan jalan pintas berupa perbedaan latar belakang. Di sisi lain, model kehilangan akses pada sebagian kriteria diagnostik dan menjadi sangat bergantung pada ketepatan tahap segmentasi. Apabila erosi memutus ekor sebuah sel normal, model akan menerima siluet sel berekor terputus sebagai citra dengan label normal, dan kekeliruan semacam ini sama sekali tidak terlihat pada metrik evaluasi mana pun.

Hasil Pelatihan

Perkembangan nilai *loss* dan *accuracy* selama 200 *epoch* pelatihan ditunjukkan pada Gambar 3.6.



Gambar 3.6 Kurva *Loss* dan *Accuracy* Pelatihan Model Morfologi

Pada grafik *loss*, nilai *training loss* menurun tajam pada *epoch* awal lalu berlanjut secara gradual hingga mencapai sekitar 0,02 pada akhir pelatihan. Nilai *validation loss* mengikuti tren yang sama meskipun disertai fluktuasi, dan tidak menunjukkan kecenderungan meningkat secara berkelanjutan. Pada grafik *accuracy*, akurasi pelatihan meningkat cepat menuju kisaran 0,97 sedangkan akurasi validasi menyusul dengan fluktuasi yang jauh lebih besar sebelum akhirnya menyatu pada tingkat yang sebanding.

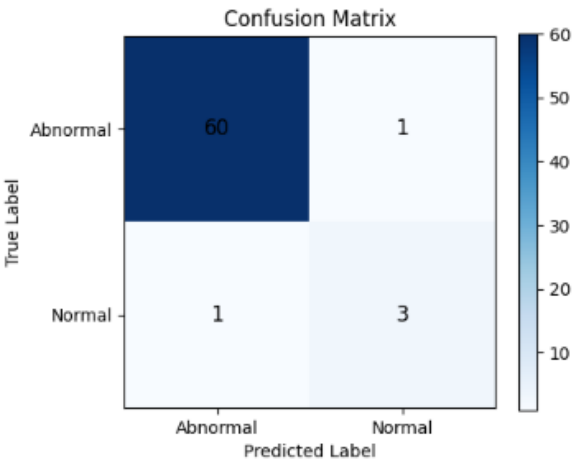
Tiga hal pada kurva tersebut layak diperhatikan lebih dekat. Yang pertama adalah amplitudo fluktuasi kurva validasi yang sangat besar, dengan akurasi yang beberapa kali terjun dari 0,97 ke sekitar 0,82 dalam satu *epoch*. Perilaku semacam ini merupakan ciri khas himpunan validasi berukuran sangat kecil; pada himpunan yang hanya memuat beberapa puluh sampel, perubahan hasil pada satu atau dua citra saja sudah cukup menggeser akurasi beberapa poin persentase sekaligus.

Yang kedua adalah lonjakan tajam *validation loss* di sekitar *epoch* ke-52 yang bersamaan dengan penurunan akurasi validasi. Peristiwa semacam ini biasanya menandakan model sempat melewati wilayah tidak stabil pada permukaan optimasi sebelum kembali pulih. Yang ketiga, dan paling berkonsekuensi, adalah bahwa pelatihan berjalan hingga *epoch* ke-200 tanpa *early stopping* maupun *ModelCheckpoint*. Ini berarti model yang digunakan pada

tahap evaluasi adalah model pada *epoch* terakhir, bukan model dengan performa validasi terbaik. Mengingat kurva validasi berfluktuasi begitu besar, *epoch* ke-200 tidak memiliki keistimewaan apa pun dibandingkan puluhan *epoch* lain di sekitarnya, dan pemilihannya sepenuhnya merupakan kebetulan. Penambahan *ModelCheckpoint* yang memantau *validation loss* merupakan perbaikan yang sangat murah dan seharusnya diterapkan.

Evaluasi Model

Evaluasi dilakukan menggunakan data uji yang tidak terlibat pada tahap pelatihan maupun validasi. Hasilnya disajikan dalam bentuk *confusion matrix* pada Gambar 3.7.



Gambar 3.7 *Confusion Matrix* Model *EfficientNet-V2S*

Dari 61 citra berlabel abnormal, sebanyak 60 citra berhasil dikenali dengan benar dan hanya satu yang keliru diprediksi sebagai normal. Dari 4 citra berlabel normal, sebanyak 3 citra dikenali dengan benar dan satu keliru diprediksi sebagai abnormal. Rangkuman metrik selengkapnya disajikan pada Tabel 3.3.

Tabel 3.3 *Classification Report* Model *EfficientNet-V2S*

Kelas	***Precision***	***Recall***	**F1-Score**	**Jumlah Sampel**
Abnormal	0,98	0,98	0,98	61
Normal	0,75	0,75	0,75	4
Accuracy	—	—	**0,97**	**65**

Pada kelas abnormal, nilai *precision*, *recall*, dan *F1-score* yang seluruhnya mencapai 0,98 menunjukkan konsistensi yang sangat baik. Pada kelas normal, ketiga nilai tersebut sebesar 0,75. Akurasi keseluruhan mencapai 0,97. Angka-angka tersebut tampak menjanjikan, tetapi pembacaannya menuntut kehati-hatian yang akan diuraikan pada subbab berikut.

Membaca Angka 97% Secara Kritis

Kolom jumlah sampel pada Tabel 3.3 merupakan bagian terpenting dari keseluruhan tabel tersebut, dan justru bagian yang paling mudah terlewat. Data uji hanya terdiri atas 65 citra, dengan komposisi 61 abnormal dan 4 normal. Tiga konsekuensi mengalir langsung dari dua angka itu.

Konsekuensi pertama berkaitan dengan *baseline*. Menggunakan Persamaan 3.10, sebuah model tanpa kecerdasan apa pun yang selalu memprediksi abnormal akan memperoleh akurasi sebesar 61 dibagi 65, yaitu 93,8%. Akurasi model sebesar 96,9% karena itu hanya unggul sekitar tiga poin persentase dari *baseline* tersebut. Dalam hitungan absolut, keunggulan itu setara dengan dua citra: model berhasil mengenali dengan benar 63 citra sementara *baseline* mengenali 61. Ini tidak berarti model tersebut tidak berguna, melainkan bahwa angka 97% jauh lebih sedikit bermakna daripada kesan yang ditimbulkannya. Setiap laporan hasil pada data yang sangat tidak seimbang seharusnya mencantumkan *baseline* kelas mayoritas berdampingan dengan akurasi model.

Konsekuensi kedua berkaitan dengan ketelitian pengukuran pada kelas normal. Seluruh metrik kelas normal sebesar 0,75 berasal dari tiga citra yang benar di antara empat citra yang tersedia. Andaikan satu citra tambahan salah diklasifikasikan, nilainya akan turun menjadi 0,50; andaikan satu citra yang keliru berhasil dikenali, nilainya naik menjadi 1,00. Dengan kata lain, ketelitian pengukuran pada kelas ini hanya sebesar 25 poin persentase. Angka 0,75 tersebut secara statistik tidak dapat dibedakan dari 0,50 maupun 1,00, dan sesungguhnya lebih tepat dilaporkan sebagai "3 dari 4" daripada sebagai sebuah rasio yang memberi kesan ketelitian yang tidak dimilikinya.

Konsekuensi ketiga bersifat lebih halus namun paling penting bagi penerapan klinis. Kelas normal justru merupakan kelas yang paling menentukan secara diagnostik, sebab keputusan klinis bertumpu pada persentase sel berbentuk normal dengan ambang WHO sebesar 4%. Sistem ini memiliki data uji paling sedikit tepat pada kelas yang paling menentukan hasilnya. Perlu ditambahkan bahwa untuk membedakan sampel dengan 3% sel normal dari sampel dengan 5% sel normal, sistem harus mampu menghitung sel normal dengan ketelitian jauh lebih tinggi daripada yang dapat dijamin oleh empat sampel uji.

Ketidakseimbangan yang Bukan Cacat Data

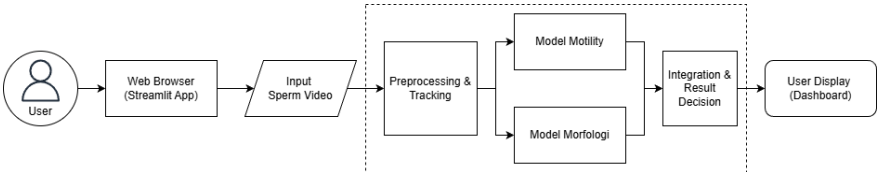
Perbandingan 61 berbanding 4 pada data uji setara dengan 6,2% sel normal. Angka tersebut sekilas tampak sebagai persoalan pengumpulan data yang parah, tetapi kriteria ketat WHO menetapkan batas bawah rujukan untuk sel berbentuk normal sebesar 4%. Dengan demikian, proporsi tersebut justru merupakan gambaran yang wajar bagi sampel semen manusia; ketidakseimbangan ini merupakan sifat domain, bukan kelalaian pengambilan data.

Pemahaman ini mengubah cara persoalannya harus disikapi. Menyeimbangkan kelas melalui augmentasi tetap dapat dibenarkan untuk keperluan pelatihan, tetapi data uji harus mempertahankan distribusi alaminya agar hasil evaluasi tetap merepresentasikan kondisi nyata dan pada bab ini hal itu sudah dilakukan dengan benar.

Yang belum memadai adalah ukurannya. Karena kelas normal secara alamiah langka, memperoleh jumlah sampel uji yang cukup pada kelas tersebut menuntut himpunan uji yang jauh lebih besar secara keseluruhan. Untuk memperoleh sekitar 100 sel normal pada data uji, diperlukan sekitar 1.600 sel yang dinilai. Inilah biaya sebenarnya dari evaluasi yang layak pada domain semacam ini, dan biaya tersebut tidak dapat dihindari melalui teknik statistik apa pun.

Integrasi Sistem

Model morfologi diintegrasikan bersama model motilitas ke dalam satu sistem berbasis web yang dibangun menggunakan kerangka kerja *Streamlit*. Alur kerja sistem ditunjukkan pada Gambar 3.8.



Gambar 3.8 Alur Kerja Sistem Terintegrasi

Pengguna mengunggah video melalui peramban, sistem menjalankan praproses dan pelacakan partikel, kemudian data bercabang menuju dua jalur inferensi paralel. Model motilitas menganalisis pola pergerakan menjadi kategori progresif, non-progresif, dan immotil, sedangkan model morfologi mengklasifikasikan bentuk setiap sel yang berhasil diisolasi. Kedua hasil kemudian digabungkan kembali menjadi satu basis data terpadu sebelum ditampilkan pada dasbor sebagaimana ditunjukkan pada Gambar 3.9.



Gambar 3.9 Tampilan Halaman *Summary Dashboard*

Dasbor menampilkan status fertilitas secara mencolok pada bagian atas, diikuti perincian jumlah sel pada setiap kategori motilitas dan morfologi. Pada contoh yang ditampilkan, sistem menyimpulkan status FERTIL berdasarkan persentase motilitas

progresif sebesar 61,25% yang memang jauh melampaui ambang WHO sebesar 32%.

Perhatian: Keputusan yang Mengabaikan Separuh Datanya Sendiri

Perhatikan panel morfologi pada Gambar 3.9: tercatat 6 sel normal dan 74 sel abnormal, yang berarti hanya 7,5% sel berbentuk normal. Meskipun angka ini masih berada di atas ambang WHO sebesar 4%, ia berada di bawah 9% yang oleh sebagian literatur dijadikan penanda risiko subfertil.

Persoalannya, keputusan FERTIL yang ditampilkan sistem sepenuhnya didasarkan pada ambang motilitas sebesar 32% dan sama sekali tidak mempertimbangkan angka morfologi tersebut. Padahal seluruh jalur pemodelan morfologi yang dibahas sepanjang bab ini dibangun justru untuk menghasilkan angka itu.

Sebuah sistem yang menampilkan dua parameter berdampingan namun hanya menggunakan salah satunya untuk mengambil keputusan berisiko menimbulkan salah tafsir, sebab pengguna wajar menduga bahwa kesimpulan yang ditampilkan telah memperhitungkan seluruh informasi yang tersaji di layar. Perbaikannya sederhana: aturan keputusan perlu menyertakan ambang morfologi sebesar 4%, dan status yang ditampilkan sebaiknya memuat kategori antara untuk kasus yang berada di zona ambang, bukan sekadar dikotomi fertil dan infertil.

Pembahasan dan Keterbatasan

Hasil pada bab ini perlu dibaca bersama sejumlah keterbatasan berikut.

a. Ukuran data uji. Himpunan uji hanya memuat 65 citra dengan 4 citra pada kelas normal, sehingga metrik kelas normal memiliki ketelitian yang sangat rendah dan akurasi keseluruhan hanya unggul tiga poin dari baseline kelas mayoritas.
b. Ground truth pengamat tunggal. Pelabelan dilakukan melalui pengamatan visual satu orang tanpa pengukuran kesepakatan antarpengamat, sehingga batas atas performa model tidak diketahui.

c. Pemilihan model tanpa *checkpoint*. Ketiadaan early stopping maupun *ModelCheckpoint* membuat model yang dievaluasi merupakan model pada *epoch* terakhir, bukan model terbaik.
d. Kehilangan informasi tekstur. Isolasi objek menghasilkan siluet biner sehingga struktur internal seperti akrosom, yang termasuk kriteria penilaian WHO, tidak lagi tersedia bagi model.
e. Base model dibekukan sepenuhnya. Tidak dilakukan *fine-tuning* pada lapisan akhir encoder, padahal siluet biner sangat berbeda dari citra alami *ImageNet* sehingga penyesuaian berpotensi memberikan peningkatan berarti.
f. Bias seleksi frame. Pemilihan *frame* berdasarkan nilai signal tertinggi tidak menyingkirkan sel yang sepanjang perekaman selalu tampil samar.
g. Aturan keputusan yang belum lengkap. Status fertilitas pada dasbor hanya mempertimbangkan motilitas dan mengabaikan hasil analisis morfologi.

Sejumlah arah pengembangan dapat ditempuh untuk mengatasi keterbatasan tersebut. Prioritas tertinggi jelas berada pada perluasan data uji, khususnya pada kelas normal, sebab tanpa itu setiap perbaikan lain tidak dapat diukur secara meyakinkan. Berikutnya adalah pelabelan oleh sekurang-kurangnya dua analis disertai pelaporan koefisien kesepakatan, yang akan menetapkan tolok ukur yang wajar bagi performa model. Perbaikan teknis yang murah namun berdampak mencakup penambahan *Model Checkpoint*, penerapan *fine-tuning* bertahap pada lapisan akhir *encoder*, serta pelaporan *baseline* kelas mayoritas berdampingan dengan akurasi. Pengembangan yang lebih ambisius dapat mencakup penyediaan citra keabuan berdampingan dengan siluet biner sebagai masukan dua kanal, sehingga model memperoleh kembali akses pada informasi tekstur tanpa kehilangan manfaat siluet yang bersih.

➢ Ringkasan

Bab ini telah menguraikan pembangunan jalur klasifikasi morfologi *spermatozoa* secara menyeluruh, mulai dari video mikroskopis mentah hingga dasbor terintegrasi. Beberapa hal penting dapat dirangkum sebagai berikut.

a. Rangkaian operasi morfologi berupa ambang adaptif, erosi, *opening*, analisis komponen terhubung, pengisian kontur, dan *closing* berhasil mengubah citra mikroskopis yang padat menjadi siluet sel tunggal yang bersih dan siap diklasifikasikan.
b. Model *EfficientNet-V2S* dengan *transfer learning* dan *Binary Focal Cross-Entropy* mencapai akurasi sebesar 96,9% dengan *F1-score* kelas abnormal sebesar 0,98.
c. Akurasi tersebut hanya unggul tiga poin persentase dari *baseline* kelas mayoritas sebesar 93,8%, sehingga sumbangan sesungguhnya dari model jauh lebih sederhana daripada kesan yang ditimbulkan angka 97%.
d. Metrik kelas normal sebesar 0,75 berasal dari tiga citra benar di antara empat citra, dan karena itu memiliki ketelitian pengukuran sebesar 25 poin persentase serta tidak dapat dibedakan secara statistik dari 0,50 maupun 1,00.
e. Ketidakseimbangan kelas sebesar 6,2% sel normal merupakan sifat alamiah domain yang sejalan dengan ambang WHO sebesar 4%, bukan cacat pengumpulan data; konsekuensinya, evaluasi yang layak menuntut himpunan uji yang jauh lebih besar.
f. Isolasi menjadi siluet biner menghilangkan seluruh informasi tekstur internal, sehingga model hanya menilai bentuk luar dan menjadi sangat bergantung pada ketepatan tahap segmentasi.

Gagasan yang paling penting untuk dibawa dari bab ini adalah bahwa membaca hasil dengan benar merupakan keterampilan yang setara pentingnya dengan membangun model. Bab 1 memperlihatkan bahwa akurasi tinggi dapat diperoleh melalui alasan yang keliru, Bab 2 memperlihatkan bahwa satu model dapat tampak baik atau buruk bergantung pada cara pembacaan metriknya, dan bab ini memperlihatkan bahwa sebuah angka dapat sepenuhnya benar secara perhitungan namun tetap menyesatkan apabila dilaporkan tanpa pembanding dan tanpa jumlah sampel yang mendasarinya.

➢ Daftar Pustaka

Aulia, S. N., dkk. (2017). The Pattern of Abnormalities on Sperm Analysis: A Study of 1186 Infertile Male in Yasmin IVF Clinic Jakarta. Dalam The 1st Physics and Technologies in Medicine and Dentistry. IOP Publishing.

Crocker, J. C., & Grier, D. G. (1996). Methods of Digital Video Microscopy for Colloidal Studies. Journal of Colloid and Interface Science, 179(1), 298–310.

Ghosh, S., & Das, S. (2024). Multi-scale Morphology-aided Deep Medical Image Segmentation. Engineering Applications of *Artificial Intelligence*, 137.

Gonzalez, R. C., & Woods, R. E. (2009). Digital Image Processing (3rd ed.). Pearson Prentice Hall.

Haralick, R. M., Sternberg, S. R., & Zhuang, X. (1987). Image Analysis Using Mathematical Morphology. IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(4), 532–550.

Haugen, T. B., Witczak, O., Hicks, S. A., Björndahl, L., Andersen, J. M., & Riegler, M. A. (2023). Sperm Motility Assessed by Deep *Convolutional Neural Network*s into WHO Categories. Scientific Reports, 13.

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal *Loss* for Dense Object Detection. Dalam Proceedings of the IEEE International Conference on Computer Vision (ICCV), hlm. 2980–2988.

Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for Activation Functions. arXiv preprint arXiv:1710.05941.

Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller Models and Faster *Training*. Dalam Proceedings of the 38th International Conference on Machine Learning (ICML), hlm. 10096–10106.

Tempola, F., Munazat, A., & Refialy, S. L. P. (2026). A *Convolutional Neural Network* Model with *Feature* Fusion for Sperm Morphology Classification. Engineering, Technology and Applied Science Research, 16.

World Health Organization. (2021). WHO Laboratory Manual for the Examination and Processing of Human Semen (6th ed.). Jenewa: WHO.

World Health Organization. (2023). 1 in 6 People Globally Affected by Infertility. Jenewa: WHO.

BAB 4
KLASIFIKASI DELAPAN KELAS PENYAKIT RETINA PADA CITRA OCT: PERBANDINGAN *RESNET-50* DAN *EFFICIENTNET-B0* DENGAN *GRAD-CAM++*

Setelah mempelajari bab ini, pembaca diharapkan mampu:

a. Menjelaskan karakteristik citra *Optical Coherence Tomography* dan delapan kelas penyakit retina yang dianalisis pada bab ini.
b. Merancang eksperimen perbandingan dua arsitektur secara adil, serta mengenali faktor-faktor yang diam-diam merusak keadilan perbandingan tersebut.
c. Menerapkan strategi pelatihan dua tahap berupa *feature extraction* yang dilanjutkan *fine-tuning.*
d. Membaca *confusion matrix* multikelas untuk menemukan pola kekeliruan yang bermakna secara klinis, bukan sekadar menghitung akurasi.
e. Membedakan rerata makro dan rerata mikro pada metrik multikelas serta memahami kapan masing-masing digunakan.
f. Memeriksa keluaran *Grad-CAM++* secara kritis, termasuk memverifikasi arah pembacaan peta warnanya sebelum menarik kesimpulan.

➢ Pendahuluan

Gangguan retina merupakan salah satu penyebab utama gangguan penglihatan dan kebutaan di dunia. Berbagai penyakit retina seperti *Choroidal Neovascularization* (CNV), *Diabetic Macular Edema* (DME), dan drusen dapat menimbulkan kerusakan permanen pada fungsi penglihatan apabila tidak dideteksi dan ditangani sejak dini. Sifat kerusakan yang tidak dapat dipulihkan inilah yang menjadikan deteksi dini bukan sekadar keunggulan, melainkan penentu apakah penglihatan seorang pasien masih dapat diselamatkan atau tidak.

Optical Coherence Tomography (OCT) telah menjadi modalitas utama untuk keperluan tersebut. Teknik ini menghasilkan citra penampang melintang retina dengan resolusi mikrometer secara non-invasif, sehingga lapisan-lapisan retina yang pada pemeriksaan funduskopi hanya tampak sebagai permukaan dapat diamati satu per satu. Perubahan halus seperti penebalan lapisan, penumpukan cairan di bawah retina, maupun deposit kecil pada membran Bruch dapat teridentifikasi jauh sebelum pasien menyadari adanya penurunan penglihatan.

Interpretasi citra OCT tetap bergantung pada tenaga medis dengan kompetensi khusus di bidang oftalmologi. Proses analisis manual memerlukan waktu, pengalaman klinis, dan konsistensi dalam pengambilan keputusan. Peningkatan jumlah pasien serta kompleksitas karakteristik penyakit retina meningkatkan beban kerja tenaga medis dan berpotensi menimbulkan variasi hasil diagnosis. Kondisi tersebut menunjukkan adanya kebutuhan akan sistem otomatis yang mampu membantu proses klasifikasi secara cepat dan konsisten.

Bab ini berbeda dari tiga bab sebelumnya dalam satu hal yang berdampak menyeluruh: permasalahan yang dihadapi bersifat multikelas dengan delapan kategori sekaligus, bukan biner. Pergeseran ini mengubah banyak hal. Kekeliruan tidak lagi hanya memiliki dua bentuk, melainkan dapat terjadi antara pasangan kelas mana pun, sehingga *confusion matrix* berubah dari tabel 2 × 2 menjadi tabel 8 × 8 yang justru menjadi sumber informasi terkaya pada keseluruhan evaluasi. Selain itu, metrik seperti *precision* dan *recall* tidak lagi menghasilkan satu angka, melainkan delapan angka yang perlu diringkas melalui suatu bentuk rerata, dan pemilihan cara meringkas tersebut sendiri merupakan keputusan yang perlu dipahami.

Dua arsitektur dibandingkan pada bab ini, yaitu *ResNet-50* dan *EfficientNet-B0*. Keduanya sudah dikenal pembaca sejak Bab 1, sehingga fokus pembahasan di sini bukan lagi pada cara kerjanya, melainkan pada rancangan perbandingannya. Membandingkan dua arsitektur secara adil jauh lebih sulit daripada yang tampak: menyamakan nilai *hyperparameter* pada tabel konfigurasi belum tentu berarti kedua model memperoleh perlakuan yang setara. Bab

ini akan menunjukkan beberapa titik di mana keadilan perbandingan dapat rusak tanpa disadari.

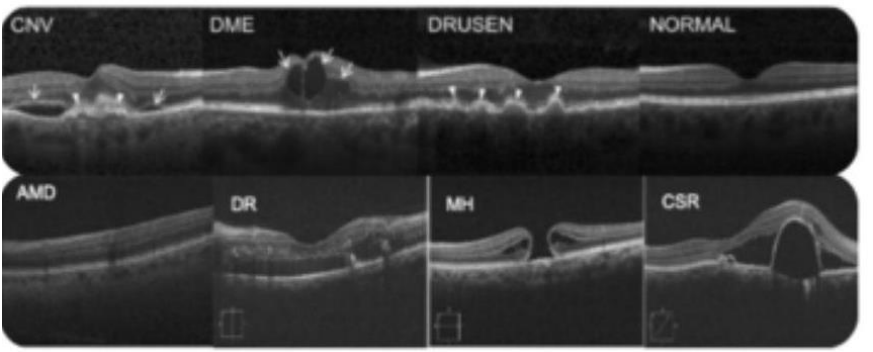
Interpretabilitas juga tetap menjadi perhatian. Model CNN bekerja sebagai sistem *black box* sehingga proses pengambilan keputusannya sulit dipahami tenaga medis, dan kurangnya transparansi tersebut menurunkan tingkat kepercayaan terhadap penerapan kecerdasan buatan dalam praktik klinis. Bab ini menerapkan *Grad-CAM++* untuk memvisualisasikan area citra yang paling berpengaruh terhadap keputusan model. Yang membedakan pembahasan di sini dari Bab 2 adalah penekanan pada satu langkah yang sering terlewat, yaitu memverifikasi arah pembacaan peta warna sebelum menarik kesimpulan apa pun dari sebuah *heatmap*.

➢ Pengantar Teori

Bagian ini menguraikan landasan teori yang khas bagi permasalahan pada bab ini. Konsep mengenai CNN, *residual connection*, *compound scaling*, dan *transfer learning* telah dibahas pada Bab 1, sedangkan *Grad-CAM* telah diuraikan pada Bab 1 dan *Grad-CAM++* pada Bab 3, sehingga tidak diulang kembali.

Citra OCT dan Delapan Kelas Penyakit Retina

OCT bekerja berdasarkan prinsip interferometri koherensi rendah. Berkas cahaya inframerah dipancarkan ke retina, lalu pantulannya dari berbagai kedalaman jaringan diukur dan disusun menjadi citra penampang melintang. Hasilnya menyerupai potongan histologis, tetapi diperoleh tanpa menyentuh pasien sama sekali. Pada citra tersebut, lapisan-lapisan retina tampak sebagai pita horizontal dengan tingkat keterangan berbeda, dan kelainan menampakkan diri sebagai penyimpangan dari susunan pita yang seharusnya. Contoh citra dari kedelapan kelas yang dianalisis ditunjukkan pada Gambar 4.1.



Gambar 4.1 Contoh Citra OCT pada Delapan Kelas Penyakit Retina

Perbedaan visual antarkelas dirangkum pada Tabel 4.1. Memahami perbedaan ini bukan sekadar pelengkap, sebab pola kekeliruan model yang dibahas pada Subbab 4.4.4 hanya dapat dinilai kewajarannya apabila pembaca mengetahui kelas mana yang memang mirip secara visual.

Tabel 4.1 Karakteristik Visual Kedelapan Kelas pada Citra OCT

Kelas	**Nama Lengkap**	**Ciri Visual pada Citra OCT**
AMD	*Age-related Macular Degeneration*	Perubahan lapisan luar retina akibat degenerasi terkait usia
CNV	*Choroidal Neovascularization*	Pembuluh darah baru di bawah retina disertai cairan subretina
CSR	*Central Serous Retinopathy*	Rongga cairan berbentuk kubah yang mengangkat retina neurosensorik
DME	*Diabetic Macular Edema*	Rongga cairan intraretina berbentuk kista pada area makula
DR	*Diabetic Retinopathy*	Penebalan dan gangguan susunan lapisan akibat komplikasi diabetes
DRUSEN	*Drusen*	Deposit kecil berbentuk gundukan pada membran *Bruch*
MH	*Macular Hole*	Terputusnya lapisan retina secara penuh pada area *fovea*
NORMAL	Retina Normal	Susunan lapisan retina utuh dengan lekukan *fovea* yang normal

Catatan Klinis: Mengapa DRUSEN Merupakan Kelas Tersulit

Drusen merupakan deposit berukuran kecil yang menumpuk di antara epitel pigmen retina dan membran Bruch. Pada citra OCT, kelainan ini tampak sebagai gundukan halus yang mengangkat sedikit lapisan di atasnya perubahan yang jauh lebih tidak mencolok dibandingkan rongga cairan pada DME atau kubah cairan pada CSR.

Kesulitannya berlipat karena drusen berada pada posisi tengah di sepanjang rangkaian perkembangan penyakit. Drusen berukuran kecil dapat ditemukan pada mata yang secara klinis dianggap normal, sedangkan drusen yang membesar dan menyatu merupakan penanda AMD tahap awal, dan AMD tahap lanjut dapat berkembang menjadi CNV. Dengan demikian, batas antara DRUSEN dengan NORMAL di satu sisi dan dengan CNV di sisi lain sesungguhnya merupakan garis yang ditarik pada suatu kesinambungan, bukan pemisah antara dua keadaan yang benar-benar terpisah.

Perlu pula dicatat bahwa dataset ini menempatkan AMD dan DRUSEN sebagai dua kelas yang berbeda, padahal drusen mendefinisikan AMD tahap awal. Pemisahan semacam ini bergantung sepenuhnya pada aturan pelabelan yang digunakan penyusun dataset. Ketika nantinya terlihat bahwa model mengenali AMD nyaris sempurna namun kesulitan pada DRUSEN, penjelasannya mungkin tidak sepenuhnya terletak pada kemampuan model.

Klasifikasi Multikelas dan Fungsi *Softmax*

Pada klasifikasi biner, satu neuron keluaran dengan aktivasi sigmoid sudah memadai. Pada klasifikasi multikelas, lapisan keluaran memuat sejumlah neuron sebanyak kelas yang ada, dan keluarannya diubah menjadi distribusi probabilitas melalui fungsi *softmax*.

$$softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad (4.1)$$

Dengan z menyatakan keluaran mentah atau *logit* pada setiap neuron dan K menyatakan jumlah kelas, yang pada bab ini bernilai delapan. Sifat penting fungsi ini adalah bahwa seluruh keluarannya

berjumlah satu, sehingga kelas-kelas bersaing satu sama lain: menaikkan probabilitas satu kelas dengan sendirinya menurunkan probabilitas kelas lainnya. Konsekuensi dari sifat inilah yang membuat kelas-kelas dengan penampakan mirip saling menekan dan menghasilkan pola kekeliruan yang khas.

Fungsi kerugian yang bersesuaian adalah *categorical cross-entropy*, yaitu perumuman dari *binary cross-entropy* yang telah dibahas pada Bab 1.

$$L = -\sum_{i=1}^{K} y_i \, log(\hat{y}_i) \qquad (4.2)$$

Karena label sebenarnya berbentuk vektor *one-hot* yang hanya bernilai satu pada kelas yang benar, penjumlahan tersebut sesungguhnya menyisakan satu suku saja, yaitu logaritma negatif dari probabilitas yang diberikan model kepada kelas yang benar. Dengan demikian, proses pelatihan pada dasarnya berupaya menaikkan probabilitas kelas yang benar setinggi mungkin, tanpa peduli bagaimana sisa probabilitasnya terbagi di antara ketujuh kelas lain.

Metrik Multikelas dan Pilihan Cara Meringkas

Pada permasalahan multikelas, *precision* dan *recall* dihitung untuk setiap kelas secara terpisah dengan memperlakukan kelas tersebut sebagai positif dan seluruh kelas lain sebagai negatif. Hasilnya berupa delapan pasang nilai yang perlu diringkas menjadi satu angka, dan terdapat dua cara yang lazim digunakan dengan makna yang berbeda. Rerata makro menghitung rata-rata sederhana atas seluruh kelas.

$$Precision_{makro} = \frac{1}{K}\sum_{i=1}^{K} Precision_i \qquad (4.3)$$

Rerata mikro sebaliknya menggabungkan seluruh perhitungan terlebih dahulu sebelum membentuk rasio.

$$Precision_{mikro} = \frac{\sum_{i=1}^{K} TP_i}{\sum_{i=1}^{K}(TP_i + FP_i)} \qquad (4.4)$$

Perbedaan keduanya menjadi penting ketika distribusi kelas tidak seimbang, sebab rerata makro memperlakukan setiap kelas sama pentingnya sedangkan rerata mikro memberi bobot menurut jumlah sampel sehingga didominasi kelas terbanyak. Pada bab ini dataset bersifat seimbang sempurna dengan 350 citra uji pada setiap

kelas, sehingga rerata mikro identik dengan akurasi keseluruhan dan rerata makro merupakan pilihan yang lebih informatif karena tetap memperlihatkan pengaruh kelas yang berperforma buruk.

Perlu pula ditegaskan bahwa *baseline* kelas mayoritas yang menjadi pembanding wajib pada Bab 3 hampir tidak berperan di sini. Pada dataset seimbang berisi delapan kelas, sebuah model yang selalu menebak satu kelas yang sama hanya memperoleh akurasi sebesar seperdelapan atau 12,5%. Setiap angka akurasi di atas 90% pada bab ini karena itu merupakan capaian yang bermakna, bukan sekadar cerminan distribusi data.

Strategi Pelatihan Dua Tahap

Pelatihan pada bab ini dilakukan melalui dua tahap yang berurutan. Tahap pertama berupa *feature extraction*, yaitu seluruh bobot model dasar dibekukan dan hanya kepala klasifikasi yang dilatih. Tahap kedua berupa *fine-tuning*, yaitu sebagian lapisan akhir model dasar dibuka kembali dan dilatih bersama kepala klasifikasi menggunakan *learning rate* yang jauh lebih kecil.

Urutan tersebut tidak dapat dipertukarkan. Kepala klasifikasi yang baru dibuat memiliki bobot acak, sehingga pada iterasi awal ia menghasilkan gradien yang besar dan tidak beraturan. Apabila lapisan model dasar sudah terbuka pada saat itu, gradien tersebut akan merambat masuk dan merusak representasi hasil pelatihan *ImageNet* yang justru menjadi alasan digunakannya *transfer learning*. Pembekuan pada tahap pertama memberi kesempatan kepada kepala klasifikasi untuk mencapai kondisi yang wajar terlebih dahulu. Penurunan *learning rate* sebesar sepuluh kali lipat pada tahap kedua melanjutkan logika perlindungan yang sama.

➢ Metodologi

Dataset

Metodologi ini menggunakan dataset sekunder berupa citra OCT retina dari *Retinal OCT Image Classification – C8* yang tersedia pada platform Kaggle. Dataset memuat 24.000 citra yang terbagi merata ke dalam delapan kelas, kemudian dibagi menjadi 18.400 citra pelatihan, 2.800 citra validasi, dan 2.800 citra pengujian sebagaimana dirangkum pada Tabel 4.2.

Tabel 4.2 Pembagian Dataset Retinal OCT C8

Subset	**Jumlah Citra**	**Proporsi**	**Citra per Kelas**
Training	18.400	76,7%	± 2.300
Validation	2.800	11,7%	350
Testing	2.800	11,7%	350
Total	**24.000**	**100%**	**3.000**

Keseimbangan sempurna dataset ini merupakan keunggulan sekaligus keterbatasan. Keunggulannya, perbandingan antar arsitektur menjadi bersih karena tidak terganggu pengaruh ketidakseimbangan kelas, dan seluruh metrik dapat dibaca secara langsung tanpa perlu penyesuaian. Keterbatasannya, keseimbangan semacam ini tidak pernah dijumpai pada praktik klinis. Pada populasi pasien yang datang ke klinik mata, kasus retina normal dan diabetik jauh lebih banyak daripada kasus *macular hole*, sehingga model yang dilatih pada distribusi seragam akan menghadapi distribusi yang sepenuhnya berbeda ketika diterapkan. Persoalan ini merupakan kebalikan dari yang dijumpai pada Bab 3, tempat ketidakseimbangan justru merupakan sifat alamiah domain yang harus dipertahankan.

***Preprocessing* dan Augmentasi**

Tahap *preprocessing* terdiri atas dua langkah. Seluruh citra diubah ukurannya menjadi 224 × 224 piksel agar sesuai dengan ukuran masukan bawaan kedua arsitektur, kemudian nilai pikselnya dinormalisasi menggunakan fungsi *preprocess_input* bawaan model *pretrained* agar distribusinya sesuai dengan kondisi saat model dilatih pada *ImageNet*.

Augmentasi diterapkan pada data pelatihan menggunakan *ImageDataGenerator* dengan parameter yang dirangkum pada Tabel 4.3. Data validasi dan pengujian menggunakan citra asli tanpa augmentasi.

Tabel 4.3 Parameter Augmentasi pada Data Pelatihan

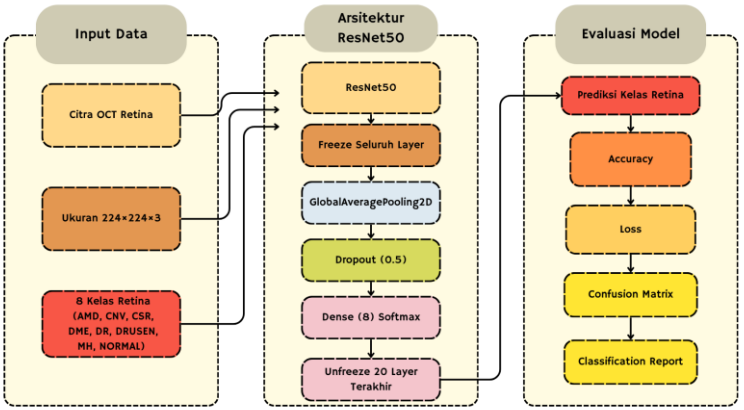
Transformasi	**Nilai**	**Catatan Kelayakan pada Citra OCT**
Rotasi	10°	Wajar; mengakomodasi kemiringan pemindaian
Zoom	10%	Wajar; variasi perbesaran antarperangkat
Horizontal flip	Aktif	Wajar; menyerupai citra mata kiri dan kanan
Width shift	5%	Wajar; variasi posisi lateral pemindaian
Height shift	5%	Perlu dibatasi; posisi vertikal retina bermakna

Pemilihan parameter tersebut secara umum konservatif dan sesuai dengan karakteristik citra OCT. *Vertical flip* tidak digunakan, dan hal ini sudah tepat: citra OCT memiliki orientasi atas-bawah yang bermakna secara anatomis karena lapisan retina tersusun dalam urutan tertentu dari vitreus di atas menuju koroid di bawah, sehingga pembalikan vertikal menghasilkan susunan yang mustahil. Sikap ini sejalan dengan pembahasan pada Bab 1 dan berbeda dari Bab 2, dan perbedaan tersebut kembali menegaskan bahwa kelayakan augmentasi selalu ditentukan oleh modalitas dan bidang irisannya.

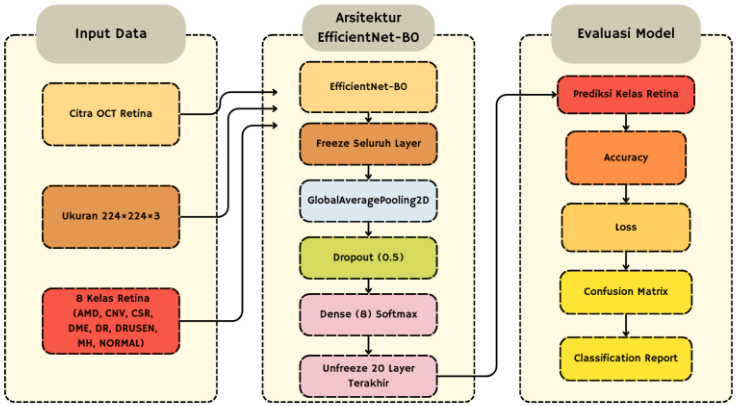
Nilai *height shift* sebesar 5% patut memperoleh perhatian tersendiri. Pada citra OCT, posisi vertikal lapisan retina di dalam bingkai citra tidak sepenuhnya arbitrer karena mencerminkan kedalaman jaringan. Pergeseran sebesar 5% dari 224 piksel setara dengan sekitar 11 piksel, yang masih tergolong kecil dan dapat diterima. Namun apabila nilai ini diperbesar, model berisiko kehilangan sinyal posisi yang justru berguna dalam membedakan kelainan pada lapisan dalam dan lapisan luar retina.

Arsitektur dan Alur Implementasi

Kedua arsitektur diterapkan menggunakan pendekatan *transfer learning* dengan struktur kepala klasifikasi yang identik. Lapisan klasifikasi bawaan dilepas, kemudian ditambahkan *Global Average Pooling2D*, *Dropout* sebesar 0,5, dan lapisan *Dense* dengan delapan neuron beraktivasi *softmax*. Alur implementasi *ResNet-50* ditunjukkan pada Gambar 4.2 dan *EfficientNet-B0* pada Gambar 4.3.



Gambar 4.2 Alur Implementasi *ResNet-50*



Gambar 4.3 Alur Implementasi *EfficientNet-B0*

Kedua alur tersebut sengaja dibuat sama persis agar perbedaan hasil dapat diatribusikan kepada karakteristik arsitektur, bukan kepada perbedaan perlakuan. Perhatikan bahwa nilai *dropout* sebesar 0,5 tergolong cukup besar untuk sebuah kepala klasifikasi yang hanya terdiri atas satu lapisan; nilai ini memang menekan risiko *overfitting*, namun sekaligus memperlambat pembelajaran dan menyebabkan gejala khas berupa akurasi validasi yang lebih tinggi daripada akurasi pelatihan, sebagaimana akan terlihat pada kurva di Subbab Hasil Pelatihan.

Konfigurasi Pelatihan

Untuk menjaga objektivitas perbandingan, kedua model dilatih menggunakan konfigurasi yang sama sebagaimana disajikan pada Tabel 4.4.

Tabel 4.4 Konfigurasi Pelatihan Kedua Model

Parameter	***ResNet-50***	***EfficientNet-B0***
Ukuran masukan	224 × 224 × 3	224 × 224 × 3
Bobot awal	*ImageNet*	*ImageNet*
Batch size	32	32
Optimizer	Adam	Adam
Learning rate tahap 1	0,0001	0,0001
Learning rate tahap 2	0,00001	0,00001
Epoch	5 + 5	5 + 5
Fine-tuning	20 lapisan terakhir	20 lapisan terakhir

Perhatian: Konfigurasi yang Sama Belum Tentu Perlakuan yang Setara

Menyamakan seluruh baris pada Tabel 4.4 merupakan langkah yang benar dan patut diapresiasi. Namun terdapat tiga hal yang tidak tampak pada tabel semacam ini dan berpotensi merusak keadilan perbandingan.

Pertama, arti "20 lapisan terakhir" berbeda pada kedua arsitektur. Implementasi *ResNet-50* pada Keras memuat sekitar 175 lapisan, sehingga 20 lapisan terakhir setara dengan sekitar 11% jaringan dan mencakup satu blok residual utuh. *EfficientNet-B0* memuat sekitar 240 lapisan karena setiap blok MBConv terurai menjadi banyak lapisan kecil, sehingga 20 lapisan terakhir hanya setara dengan sekitar 8% jaringan dan bahkan mungkin belum mencakup satu blok MBConv secara utuh. Dengan kata lain, *EfficientNet-B0* memperoleh kebebasan penyesuaian yang lebih sedikit meskipun angkanya sama. Cara yang lebih adil adalah membuka sejumlah blok, bukan sejumlah lapisan.

Kedua, lapisan Batch Normalization pada bagian yang dibuka. Ketika lapisan BN ikut dilatih pada *batch* berukuran 32, statistik

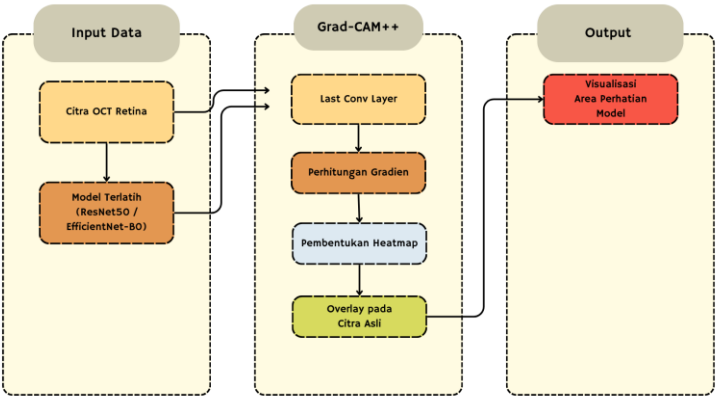
yang telah dipelajari dari *ImageNet* akan diperbarui menggunakan sampel yang sedikit. *EfficientNet* jauh lebih padat lapisan BN dibandingkan *ResNet*, sehingga dampaknya tidak setara. Praktik yang lazim adalah tetap membekukan seluruh lapisan BN selama *fine-tuning.*
Ketiga, fungsi normalisasi yang digunakan. Hal ini dibahas tersendiri pada kotak berikutnya karena merupakan persoalan yang paling berpotensi menjelaskan selisih hasil kedua model.

Jebakan Klasik: Normalisasi Masukan pada *EfficientNet* Keras

Implementasi *EfficientNet* pada Keras memuat lapisan penskalaan dan normalisasi di dalam arsitekturnya sendiri, sehingga model tersebut mengharapkan masukan berupa nilai piksel mentah pada rentang 0 sampai 255. Hal ini berbeda dari *ResNet-50*, yang mengharapkan masukan telah diproses melalui fungsi *preprocess_input* miliknya sendiri berupa pengurangan rerata *ImageNet* dan penukaran urutan kanal menjadi BGR.
Konsekuensinya, apabila fungsi *preprocess_input* milik *ResNet* atau pembagian sederhana dengan 255 diterapkan pada masukan EfficientNet, maka citra ternormalisasi dua kali. Nilai piksel yang seharusnya berada pada rentang 0–255 masuk ke jaringan pada rentang yang jauh lebih sempit, dan seluruh representasi *pretrained* bekerja di luar wilayah kerjanya. Gejalanya sangat khas: model tetap belajar dan tetap menghasilkan angka yang tampak wajar, namun secara konsisten tertinggal beberapa poin persentase dari yang seharusnya.
Laporan yang mendasari bab ini menyebutkan penggunaan *preprocess_input()* tanpa menegaskan fungsi milik arsitektur mana yang dipakai pada masing-masing jalur. Hal ini tidak berarti kekeliruan pasti terjadi, tetapi merupakan titik pertama yang harus diperiksa sebelum menyimpulkan bahwa *EfficientNet-B0* memang lebih lemah pada permasalahan ini. Pemeriksaannya sederhana: cukup pastikan bahwa setiap arsitektur menggunakan fungsi normalisasi bawaannya sendiri.

Penerapan *Grad-CAM++*

Grad-CAM++ diterapkan pada kedua model untuk memvisualisasikan area citra yang paling berkontribusi terhadap keputusan klasifikasi. Alur penerapannya ditunjukkan pada Gambar 4.4.



Gambar 4.4 Alur Penerapan *Grad-CAM++*

Prosesnya dimulai dengan mengambil keluaran lapisan konvolusi terakhir dari model terlatih, menghitung gradien skor kelas terhadap *feature map* tersebut, membentuk *heatmap* berbobot, lalu menumpangkannya pada citra asli. Rincian perhitungan *Grad-CAM* telah diuraikan pada Bab 1 melalui Persamaan 1.8 dan 1.9, sedangkan perbedaan *Grad-CAM++* telah dibahas pada Bab 3.

Satu langkah yang tidak tampak pada diagram mana pun namun sangat menentukan adalah pemetaan nilai *heatmap* menjadi warna. Sebuah peta panas hanyalah larik bilangan; warna yang muncul di layar sepenuhnya bergantung pada peta warna yang dipilih dan pada apakah nilai tersebut dinormalisasi atau dibalik terlebih dahulu. Pada peta warna *jet* yang paling lazim digunakan, merah menandakan nilai tertinggi dan biru menandakan nilai terendah. Apabila larik nilainya dibalik sebelum diwarnai, maka seluruh pembacaan ikut terbalik tanpa memberikan tanda apa pun. Kewajiban memeriksa hal ini akan menjadi pokok pembahasan pada Subbab Interpretasi dengan *Grad-CAM++*.

➢ Hasil dan Ujicoba

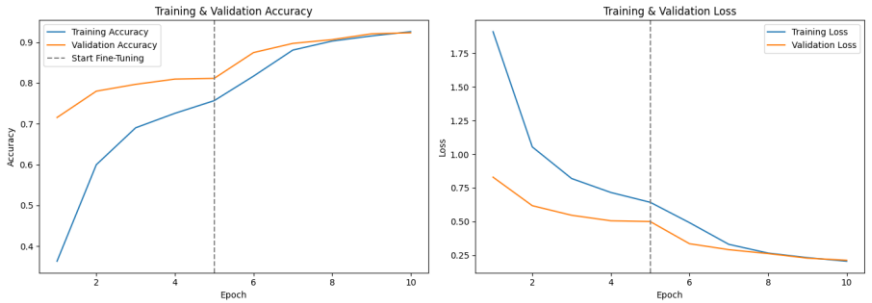
Hasil Pelatihan

Pelatihan dilakukan selama lima *epoch* pada tahap *feature extraction* dan lima *epoch* pada tahap *fine-tuning*, sehingga total sepuluh *epoch*. Ringkasan hasil pelatihan kedua model disajikan pada Tabel 4.5.

Tabel 4.5 Ringkasan Hasil Pelatihan Kedua Model

Model	***Epoch***	**Train Acc**	**Val Acc**	**Train *Loss***	**Val *Loss***
ResNet-50	10	92,59%	92,32%	0,2056	0,2131
EfficientNet-B0	10	86,37%	89,64%	0,4013	0,2916

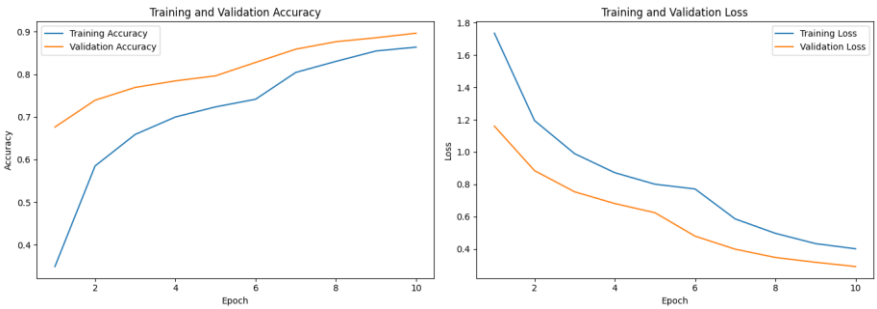
Perkembangan nilai akurasi dan *loss* pada *ResNet-50* ditunjukkan pada Gambar 4.5.



Gambar 4.5 Kurva *Accuracy* dan *Loss* Pelatihan *ResNet-50*

Garis putus-putus vertikal pada *epoch* kelima menandai peralihan dari tahap *feature extraction* ke tahap *fine-tuning*, dan pengaruhnya terlihat jelas. Sebelum garis tersebut, akurasi pelatihan tertahan di sekitar 0,75 sementara akurasi validasi bertahan di sekitar 0,81; keduanya menaik perlahan dan mulai mendatar. Segera setelah lapisan akhir dibuka, kedua kurva melonjak dan berlanjut naik hingga mencapai sekitar 0,92 pada *epoch* kesepuluh. Peningkatan sebesar tujuh belas poin persentase yang terjadi hanya dalam lima *epoch* ini merupakan bukti langsung bahwa *fine-tuning* memang diperlukan: representasi *ImageNet* sendiri tidak memadai untuk citra OCT, dan penyesuaian pada beberapa lapisan terakhir sudah cukup untuk menutup sebagian besar kesenjangannya.

Perkembangan pada *EfficientNet-B0* ditunjukkan pada Gambar 4.6.



Gambar 4.6 Kurva *Accuracy* dan *Loss* Pelatihan *EfficientNet-B0*

Polanya serupa namun lebih landai, dengan akurasi berakhir pada 0,86 untuk data pelatihan dan 0,90 untuk data validasi. Yang paling penting untuk diamati pada kedua gambar adalah bentuk kurva pada *epoch* terakhir: tidak satu pun di antaranya telah mendatar. Seluruh kurva masih menaik dan seluruh kurva *loss* masih menurun ketika pelatihan dihentikan. Ini berarti kedua model dihentikan sebelum mencapai konvergensi, dan implikasinya terhadap perbandingan bersifat mendasar sebagaimana dibahas pada kotak berikut.

Perhatian: Membandingkan Dua Model yang Belum Selesai Belajar

Sepuluh *epoch* merupakan anggaran pelatihan yang sangat kecil. Kurva pada Gambar 4.5 dan 4.6 memperlihatkan bahwa keduanya masih dalam fase peningkatan ketika pelatihan berakhir. Karena itu, angka yang dilaporkan bukanlah performa akhir masing-masing arsitektur, melainkan performa yang sempat dicapai dalam sepuluh *epoch*.

Perbedaannya penting. Kesimpulan yang sah dari eksperimen ini adalah bahwa *ResNet-50 mencapai performa lebih tinggi lebih cepat*, bukan bahwa *ResNet-50 lebih baik daripada EfficientNet-B0* pada permasalahan ini. Kedua pernyataan tersebut sering dianggap sama, padahal tidak. Sebuah arsitektur yang lebih lambat berkonvergensi dapat saja menyusul dan

bahkan melampaui pesaingnya apabila diberi anggaran pelatihan yang memadai.
Selisih laju konvergensi antara kedua arsitektur pun bukan hal yang mengejutkan. *EfficientNet-B0* hanya memiliki sekitar 5,3 juta parameter dibandingkan 25,6 juta pada *ResNet-50*, dan jaringan yang lebih kecil umumnya memerlukan lebih banyak *epoch* untuk mencapai kondisi terbaiknya.
Perbaikannya sederhana dan tidak menuntut perubahan rancangan apa pun: latih kedua model hingga *validation loss* benar-benar mendatar, gunakan *EarlyStopping* untuk menentukan titik berhentinya secara otomatis, lalu bandingkan hasil pada kondisi masing-masing model yang sudah selesai belajar.

Evaluasi pada Data Uji

Kedua model dievaluasi menggunakan 2.800 citra uji yang tidak terlibat pada tahap pelatihan maupun validasi. Hasilnya disajikan pada Tabel 4.6.

Tabel 4.6 Hasil Evaluasi Kedua Model pada Data Uji

Metrik	***ResNet-50***	***EfficientNet-B0***
Accuracy	92,86%	88,79%
Loss	0,1888	0,2959
Precision (rerata makro)	93%	89%
Recall (rerata makro)	93%	89%
F1-Score (rerata makro)	93%	89%

ResNet-50 unggul pada seluruh metrik dengan selisih akurasi sebesar 4,07 poin persentase. Dalam hitungan absolut, *ResNet-50* mengklasifikasikan dengan benar 2.600 dari 2.800 citra sedangkan *EfficientNet-B0* mencapai 2.486 citra, sehingga selisihnya setara 114 citra. Berbeda dengan situasi pada Bab 3, selisih sebesar ini pada himpunan uji berisi 2.800 sampel merupakan perbedaan yang bermakna dan tidak dapat dijelaskan sebagai keacakan semata.

Kesetaraan nilai *precision*, *recall*, dan *F1-score* pada masing-masing model merupakan konsekuensi langsung dari dataset yang seimbang sempurna. Ketika setiap kelas memiliki jumlah sampel

yang sama, jumlah seluruh *false positive* pasti sama dengan jumlah seluruh *false negative*, sebab setiap kekeliruan sekaligus merupakan keduanya bagi dua kelas yang berbeda. Kesamaan angka tersebut karena itu bukan kebetulan, melainkan sifat aritmetika yang memang harus muncul.

Performa per Kelas

Angka rerata menyembunyikan sebaran yang justru paling informatif. Nilai *F1-score* pada setiap kelas disajikan pada Tabel 4.7.

Tabel 4.7 *F1-Score* pada Setiap Kelas

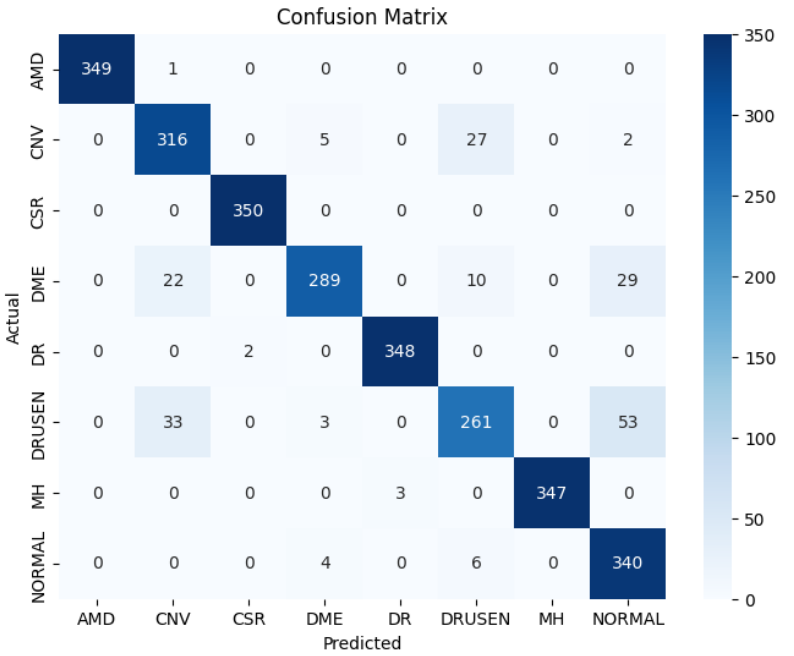
Kelas	***ResNet-50***	***EfficientNet-B0***	**Selisih**
AMD	1,00	0,99	0,01
CNV	0,88	0,82	0,06
CSR	1,00	0,96	0,04
DME	0,89	0,85	0,04
DR	0,99	0,93	0,06
DRUSEN	0,80	0,73	0,07
MH	1,00	0,93	0,07
NORMAL	0,88	0,87	0,01

ResNet-50 unggul pada seluruh kelas tanpa terkecuali, dan konsistensi ini memperkuat kesimpulan bahwa keunggulannya bersifat menyeluruh, bukan hasil dari satu atau dua kelas yang kebetulan menguntungkan. Sebaran nilainya sendiri sangat lebar: pada *ResNet-50*, rentangnya membentang dari 1,00 pada AMD, CSR, dan MH hingga 0,80 pada DRUSEN. Ini berarti pernyataan "model mencapai 93%" sesungguhnya merangkum dua kenyataan yang sangat berbeda, yaitu tiga kelas yang dikenali nyaris sempurna dan satu kelas yang seperlima kasusnya keliru.

Pola sebarannya pun masuk akal secara klinis. Kelas dengan nilai tertinggi adalah kelas dengan tanda visual yang mencolok: *macular hole* berupa terputusnya lapisan retina secara penuh, CSR berupa kubah cairan yang besar, dan AMD berupa perubahan lapisan luar yang khas. Sebaliknya, kelas dengan nilai terendah adalah kelas yang berada pada kesinambungan penyakit, yaitu DRUSEN, CNV, DME, dan NORMAL. Dengan kata lain, kesulitan model sejalan dengan kesulitan yang juga dihadapi manusia.

Analisis *Confusion Matrix*

Bagian paling informatif pada evaluasi multikelas bukanlah angka akurasi, melainkan pola kekeliruannya. *Confusion matrix ResNet-50* ditunjukkan pada Gambar 4.7.



Gambar 4.7 *Confusion Matrix ResNet-50*

Dominasi nilai pada diagonal utama memperlihatkan bahwa sebagian besar citra terklasifikasi dengan benar. Kekeliruan yang terjadi tidak tersebar merata, melainkan terpusat pada beberapa pasangan kelas tertentu sebagaimana dirangkum pada Tabel 4.8.

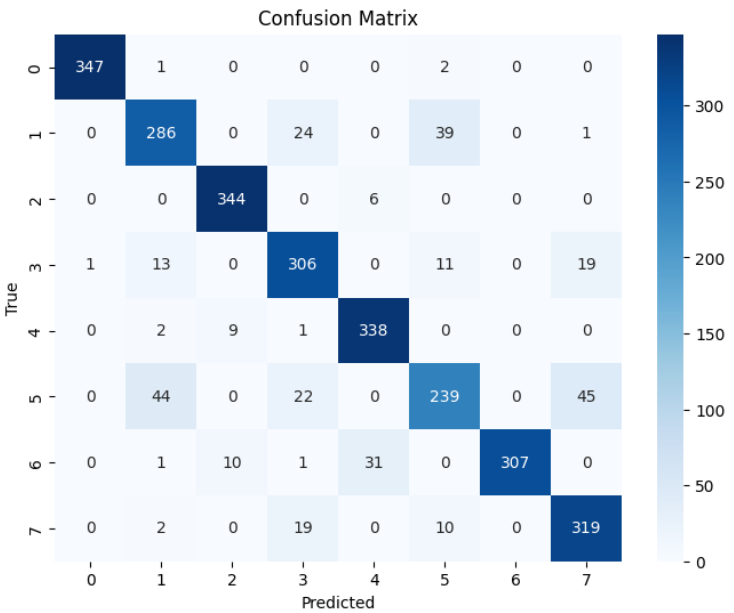
Tabel 4.8 Pola Kekeliruan Utama pada *ResNet-50*

Kelas Sebenarnya	**Diprediksi Sebagai**	**Jumlah**	***Recall* Kelas**
DRUSEN	NORMAL	53	0,746
DRUSEN	CNV	33	(261 dari 350)
DME	NORMAL	29	0,826
DME	CNV	22	(289 dari 350)
CNV	DRUSEN	27	0,903
NORMAL	DRUSEN	6	0,971

Satu angka pada tabel tersebut jauh lebih penting daripada yang lain. Sebanyak 53 citra DRUSEN diprediksi sebagai NORMAL, yang berarti 15% dari seluruh kasus drusen dinyatakan sebagai retina sehat. Secara statistik, angka ini hanyalah satu di antara 200 kekeliruan yang terjadi. Secara klinis, ini merupakan jenis kekeliruan yang paling merugikan pada keseluruhan sistem: drusen merupakan penanda AMD tahap awal, dan menyatakannya normal berarti pasien tidak memperoleh pemantauan pada tahap ketika penanganan justru paling berpeluang berhasil.

Perhatikan pula bahwa kekeliruan ini tidak simetris. Model mengirim 53 citra DRUSEN ke NORMAL, tetapi hanya 6 citra NORMAL yang dikirim ke DRUSEN. Ketidaksimetrisan sebesar hampir sembilan kali lipat menunjukkan bahwa model memiliki kecenderungan sistematis untuk meremehkan drusen, bukan sekadar kebingungan dua arah antara dua kelas yang mirip. Pada penerapan nyata, kecenderungan semacam ini dapat dikoreksi dengan memberi bobot lebih besar pada kelas DRUSEN saat pelatihan, atau dengan menurunkan ambang keputusan bagi kelas tersebut sehingga model menjadi lebih berhati-hati.

Confusion matrix EfficientNet-B0 ditunjukkan pada Gambar 4.8.



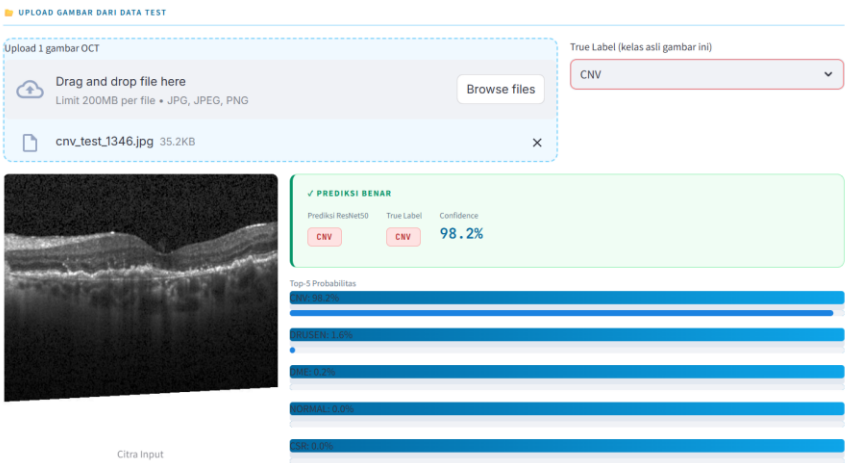
Gambar 4.8 *Confusion Matrix EfficientNet-B0*

Pola kekeliruannya serupa namun dengan besaran yang lebih tinggi. Kelas DRUSEN kembali menjadi yang paling bermasalah dengan hanya 239 dari 350 citra yang dikenali dengan benar, setara *recall* sebesar 0,683; sebanyak 45 citra dikirim ke NORMAL dan 44 citra ke CNV. Terdapat pula satu pola yang tidak muncul pada *ResNet-50*, yaitu 31 citra MH yang diprediksi sebagai DR. Kekeliruan ini agak mengherankan karena *macular hole* memiliki tanda visual yang sangat khas berupa terputusnya lapisan retina secara penuh. Kehadirannya menunjukkan bahwa *EfficientNet-B0* belum sepenuhnya mempelajari fitur pembeda kelas tersebut, yang sekali lagi konsisten dengan dugaan bahwa model ini dihentikan sebelum selesai belajar.

Kesamaan pola kekeliruan pada kedua model justru yang berharga. Ketika dua arsitektur yang sangat berbeda melakukan kesalahan pada pasangan kelas yang sama, penjelasan yang paling mungkin bukan terletak pada arsitekturnya, melainkan pada kemiripan visual yang memang ada di dalam data itu sendiri. Ini berarti mengganti arsitektur tidak akan menyelesaikan persoalan DRUSEN; yang diperlukan adalah data pelatihan yang lebih banyak pada kelas tersebut, definisi pelabelan yang lebih tegas antara DRUSEN, NORMAL, dan AMD, atau penambahan informasi di luar citra tunggal seperti data volumetrik.

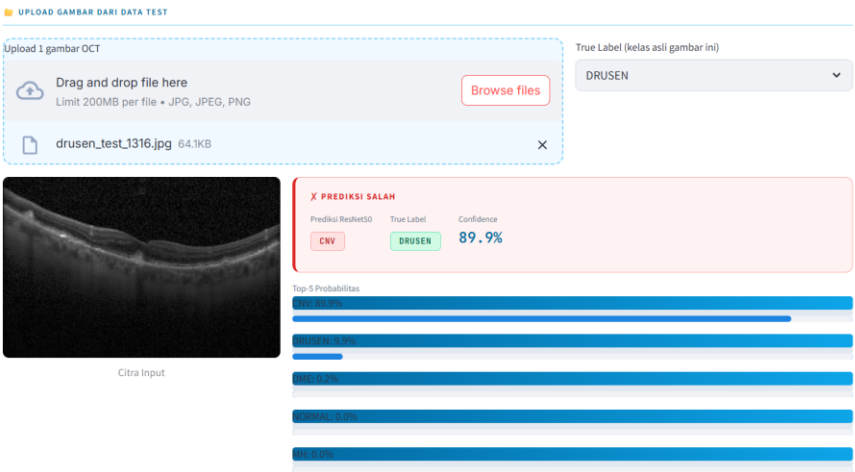
Antarmuka Sistem

Model terbaik diintegrasikan ke dalam antarmuka berbasis web bernama RetinaScan AI, yang memungkinkan pengguna mengunggah satu citra OCT dan memperoleh hasil klasifikasi beserta lima probabilitas tertinggi. Contoh keluaran pada kasus yang diprediksi dengan benar ditunjukkan pada Gambar 4.9.



Gambar 4.9 Antarmuka RetinaScan AI pada Kasus Prediksi Benar

Pada contoh tersebut, citra CNV diklasifikasikan dengan benar disertai tingkat keyakinan sebesar 98,2%, dengan DRUSEN sebagai kandidat kedua pada 1,6%. Penyajian lima probabilitas tertinggi merupakan keputusan rancangan yang tepat untuk konteks klinis, sebab pengguna dapat melihat apakah model benar-benar yakin atau sesungguhnya sedang ragu di antara dua kemungkinan. Contoh pada kasus yang keliru ditunjukkan pada Gambar 4.10.



Gambar 4.10 Antarmuka RetinaScan AI pada Kasus Prediksi Salah

Citra DRUSEN pada contoh tersebut diprediksi sebagai CNV dengan tingkat keyakinan sebesar 89,9%, sedangkan kelas yang benar hanya memperoleh 9,9%. Inilah yang disebut sebagai kekeliruan berkeyakinan tinggi, dan sifatnya jauh lebih berbahaya daripada kekeliruan yang disertai keraguan. Ketika model salah namun ragu, misalnya pada 53%, pengguna memperoleh isyarat untuk memeriksa ulang. Ketika model salah namun sangat yakin, isyarat itu tidak ada sama sekali.

Catatan: Keyakinan Bukan Probabilitas

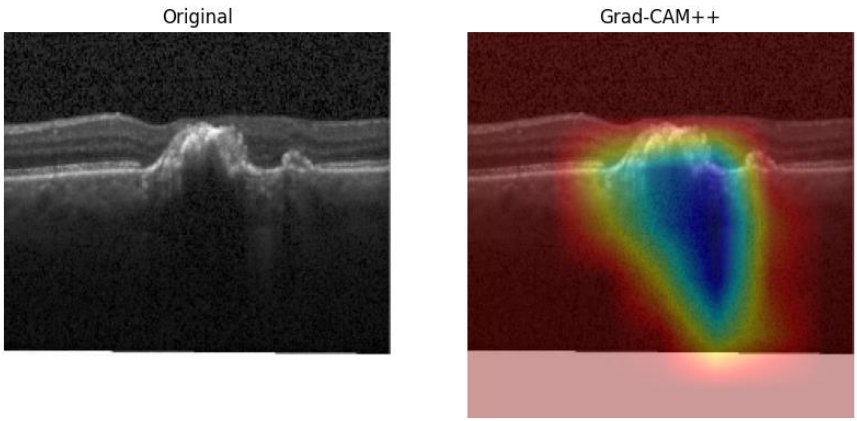
Kelima contoh kekeliruan pada lampiran laporan yang mendasari bab ini seluruhnya berasal dari kelas DRUSEN, dengan tingkat keyakinan berturut-turut sebesar 53,5%, 85,6%, 52,3%, 79,4%, dan 89,9%. Dua di antaranya berada di atas 79%, yang berarti model menyatakan keyakinan sangat tinggi pada jawaban yang salah.

Keluaran *softmax* kerap disalahartikan sebagai probabilitas dalam pengertian sesungguhnya, seolah nilai 90% berarti model akan benar pada sembilan dari sepuluh kasus serupa. Jaringan saraf modern justru dikenal berlebihan dalam menyatakan keyakinan; nilai *softmax* merupakan hasil normalisasi *logit* dan tidak dengan sendirinya terkalibrasi terhadap peluang kebenaran yang sebenarnya.

Untuk sistem yang akan digunakan di lingkungan klinis, kalibrasi merupakan kebutuhan dan bukan penyempurnaan. Teknik seperti *temperature scaling* dapat menyesuaikan keluaran menggunakan data validasi sehingga nilai keyakinan yang ditampilkan benar-benar mencerminkan peluang kebenaran. Selain itu, sistem sebaiknya menyediakan status ketiga di samping benar dan salah, yaitu penolakan untuk memutuskan ketika keyakinan tertinggi berada di bawah ambang tertentu, sehingga kasus meragukan diteruskan kepada dokter alih-alih dijawab dengan angka yang meyakinkan namun keliru.

Interpretasi dengan *Grad-CAM++*

Grad-CAM++ diterapkan untuk memeriksa apakah keputusan model bertumpu pada area citra yang relevan secara klinis. Hasil visualisasi pada sebuah citra kelas CNV ditunjukkan pada Gambar 4.11.



Gambar 4.11 Hasil Visualisasi *Grad-CAM++* pada Citra Kelas CNV

Panel kiri menampilkan citra OCT asli dengan penonjolan lapisan retina yang khas pada kasus CNV, sedangkan panel kanan menampilkan *heatmap* yang ditumpangkan pada citra tersebut. Sebelum menarik kesimpulan apa pun dari gambar ini, satu hal perlu diperiksa terlebih dahulu, dan pemeriksaan itu dibahas pada kotak berikut.

Perhatian: Periksa Arah Peta Warna Sebelum Menyimpulkan

Pada peta warna *jet* yang lazim digunakan untuk *Grad-CAM*, urutan warnanya membentang dari biru tua untuk nilai terendah, melalui hijau dan kuning, hingga merah tua untuk nilai tertinggi. Merah menandakan kontribusi terbesar dan biru menandakan kontribusi terkecil.

Amati Gambar 4.11 dengan urutan tersebut. Area lesi di tengah citra yaitu penonjolan lapisan retina yang menjadi ciri CNV berwarna *biru tua*, sedangkan seluruh area latar belakang gelap

di sekelilingnya berwarna *merah*. Dibaca menurut konvensi *jet*, gambar tersebut menyatakan bahwa kontribusi terbesar terhadap keputusan model berasal dari latar belakang, bukan dari lesi. Terdapat pula pita berwarna merah muda di bagian bawah yang berada di luar area citra retina sama sekali.

Terdapat dua kemungkinan penjelasan. *Kemungkinan pertama*, peta warna diterapkan pada nilai yang terbalik, misalnya karena *heatmap* dikurangkan dari nilai maksimumnya sebelum diwarnai. Ini merupakan kekeliruan implementasi yang lazim terjadi dan mudah diperbaiki, dan apabila benar demikian maka gambar tersebut sesungguhnya justru memperlihatkan perhatian yang tepat pada lesi. *Kemungkinan kedua*, tidak ada kekeliruan pewarnaan dan model memang bertumpu pada latar belakang yaitu gejala *shortcut learning* sebagaimana ditemukan pada Bab 1.

Yang tidak dapat dibenarkan adalah menyimpulkan tanpa memeriksa. Laporan yang mendasari bab ini menyatakan bahwa aktivasi model berfokus pada area retina yang mengalami perubahan morfologi, padahal pembacaan langsung menurut konvensi *jet* justru menyatakan sebaliknya. Pemeriksaannya membutuhkan waktu beberapa menit saja: tampilkan larik *heatmap* mentah tanpa pewarnaan, atau cetak koordinat nilai maksimumnya dan bandingkan dengan lokasi lesi. Keduanya akan langsung menjawab kemungkinan mana yang benar.

Kotak ini sengaja ditutup tanpa kesimpulan karena memang tidak ada kesimpulan yang dapat ditarik dari satu gambar saja. Inilah sikap yang benar terhadap bukti yang belum memadai, dan sikap itu sendiri merupakan bagian dari pelajaran pada bab ini.

Terlepas dari persoalan pewarnaan tersebut, pemeriksaan *Grad-CAM++* pada bab ini masih memiliki keterbatasan lain yang perlu dicatat. Visualisasi hanya ditampilkan pada satu citra dari satu kelas, yaitu CNV, dan hanya untuk kasus yang diprediksi dengan benar. Pemeriksaan yang memadai menuntut visualisasi pada seluruh delapan kelas, disertai perbandingan antara kasus benar dan kasus salah terutama pada kekeliruan DRUSEN menjadi NORMAL

yang merupakan persoalan terpenting sistem ini. Justru pada kasus yang keliru itulah *Grad-CAM++* paling berpeluang mengungkap penyebab kekeliruannya, apakah karena model memperhatikan area yang salah atau karena gundukan drusen memang terlalu halus untuk menghasilkan aktivasi yang berarti.

Pembahasan dan Keterbatasan

Hasil pada bab ini perlu dibaca bersama sejumlah keterbatasan berikut.

a. Anggaran pelatihan yang belum memadai. Sepuluh *epoch* menghentikan kedua model sebelum konvergensi, sehingga perbandingan yang dihasilkan mengukur laju konvergensi dan bukan performa akhir masing-masing arsitektur.
b. Kesetaraan *fine-tuning* yang semu. Membuka 20 lapisan terakhir berarti proporsi jaringan yang berbeda pada kedua arsitektur, dan lapisan *Batch Normalization* pada bagian yang dibuka tidak dibekukan.
c. Fungsi normalisasi yang belum dipastikan. Laporan tidak menegaskan bahwa setiap arsitektur menggunakan fungsi *preprocess_input* miliknya sendiri, padahal hal ini merupakan penjelasan yang paling mungkin bagi sebagian selisih hasil.
d. Distribusi kelas yang tidak realistis. Dataset seimbang sempurna memudahkan perbandingan, tetapi tidak mencerminkan sebaran kasus pada populasi klinis yang sesungguhnya.
e. Kekeliruan berkeyakinan tinggi. Sistem menampilkan tingkat keyakinan hingga 89,9% pada prediksi yang salah, tanpa kalibrasi maupun mekanisme penolakan untuk memutuskan.
f. Interpretasi yang belum diverifikasi. Arah pembacaan peta warna *Grad-CAM++* belum dipastikan, dan visualisasi hanya dilakukan pada satu citra dari satu kelas untuk kasus yang benar saja.
g. Sumber data tunggal. Seluruh citra berasal dari satu dataset publik tanpa validasi eksternal pada data dari perangkat OCT maupun institusi yang berbeda.

Beberapa arah pengembangan dapat ditempuh, dan yang paling berdampak justru merupakan yang paling murah. Memastikan fungsi normalisasi yang tepat bagi masing-masing arsitektur serta melatih kedua model hingga konvergensi dengan *EarlyStopping*

akan mengubah eksperimen ini menjadi perbandingan yang benar-benar adil tanpa mengubah rancangannya sama sekali. Berikutnya adalah memverifikasi arah peta warna *Grad-CAM++* dan memperluas visualisasinya ke seluruh kelas serta kasus yang keliru. Perbaikan yang menyasar persoalan klinis utama mencakup pemberian bobot lebih besar pada kelas DRUSEN saat pelatihan dan penerapan kalibrasi keyakinan disertai ambang penolakan. Pengembangan yang lebih ambisius dapat mencakup eksplorasi arsitektur yang lebih mutakhir seperti *EfficientNet*-V2, *ConvNeXt*, atau *Vision Transformer*, serta pemanfaatan data OCT volumetrik yang memuat seluruh irisan alih-alih citra tunggal.

➢ Ringkasan

Bab ini telah menguraikan perbandingan dua arsitektur CNN pada permasalahan klasifikasi delapan kelas penyakit retina berbasis citra OCT, disertai interpretasi menggunakan *Grad-CAM++* dan integrasi ke dalam antarmuka berbasis web. Beberapa hal penting dapat dirangkum sebagai berikut.

a. *ResNet-50* mencapai akurasi sebesar 92,86% dengan *loss* 0,1888, sedangkan *EfficientNet-B0* mencapai 88,79% dengan *loss* 0,2959; *ResNet-50* unggul pada seluruh delapan kelas tanpa terkecuali.
b. Karena kedua model dihentikan pada *epoch* kesepuluh saat kurvanya masih menaik, kesimpulan yang sah adalah bahwa *ResNet-50* berkonvergensi lebih cepat, bukan bahwa *ResNet-50* lebih unggul secara mutlak.
c. Menyamakan nilai *hyperparameter* pada tabel konfigurasi tidak menjamin perlakuan yang setara; proporsi lapisan yang dibuka saat *fine-tuning*, penanganan lapisan *Batch Normalization*, dan fungsi normalisasi masukan seluruhnya berbeda pengaruhnya pada kedua arsitektur.
d. Nilai *F1-score* per kelas membentang dari 1,00 hingga 0,80, sehingga angka rerata sebesar 93% merangkum dua kenyataan yang sangat berbeda dan tidak layak dilaporkan sendirian.
e. DRUSEN merupakan kelas tersulit pada kedua arsitektur, dan 53 citra drusen yang diprediksi sebagai NORMAL merupakan kekeliruan yang paling merugikan secara klinis karena berarti kasus AMD tahap awal luput dari pemantauan.

f. Sistem menampilkan keyakinan hingga 89,9% pada prediksi yang salah, sehingga kalibrasi keluaran dan mekanisme penolakan untuk memutuskan merupakan kebutuhan sebelum sistem dapat dipertimbangkan untuk penggunaan klinis.
g. Pembacaan langsung terhadap peta warna *Grad-CAM++* tidak sejalan dengan kesimpulan yang dinyatakan, sehingga arah pewarnaannya perlu diverifikasi sebelum interpretasi apa pun dapat diterima.

Gagasan yang paling penting untuk dibawa dari bab ini adalah bahwa merancang perbandingan yang adil menuntut kecermatan yang setara dengan merancang modelnya sendiri. Menyamakan angka pada tabel konfigurasi merupakan langkah pertama yang perlu, tetapi keadilan sesungguhnya ditentukan oleh hal-hal yang tidak muncul pada tabel: apakah kedua model memperoleh anggaran pelatihan yang cukup untuk selesai belajar, apakah masing-masing memperoleh masukan dalam bentuk yang diharapkannya, dan apakah proporsi jaringan yang disesuaikan benar-benar setara. Tanpa ketiganya, yang dibandingkan bukanlah dua arsitektur, melainkan dua rangkaian keadaan yang kebetulan berbeda.

➢ Daftar Pustaka

Apon, T. S., Hasan, M. M., Islam, A., & Alam, M. G. R. (2021). Demystifying *Deep Learning* Models for Retinal OCT Disease Classification Using *Explainable* AI. Dalam IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), hlm. 1–6.

Chattopadhay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018). *Grad-CAM++*: Improved Visual Explanations for Deep Convolutional Networks. Dalam IEEE Winter Conference on Applications of Computer Vision (WACV), hlm. 839–847.

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. Dalam Proceedings of the 34th International Conference on Machine Learning (ICML), hlm. 1321–1330.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Dalam Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), hlm. 770–778.

Ioffe, S., & Szegedy, C. (2015). *Batch Normalization*: Accelerating Deep Network *Training* by Reducing *Internal Covariate Shift*. Dalam Proceedings of the 32nd International Conference on Machine Learning (ICML), hlm. 448–456.

Kermany, D. S., dkk. (2018). Identifying Medical Diagnoses and Treatable Diseases by Image-Based *Deep Learning*. Cell, 172(5), 1122–1131.

Ratnasari, A. (2024). Analisis Retinal *Optical Coherence Tomography* (OCT) untuk Deteksi Kerusakan Retina Menggunakan Metode Machine Learning. JSAI: Journal of Science and Applied Informatics, 7(2), 247–252.

Saldianovitta, R. A., & Dewi, N. A. (2024). Diagnostic Approaches for Macular Dystrophy by Using OCT and VEP: A Case Series. International Journal of Retina, 7(2), 148.

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for *Convolutional Neural Network*s. Dalam Proceedings of the 36th International Conference on Machine Learning (ICML), hlm. 6105–6114.

Umair, M., dkk. (2025). OculusNet: Detection of Retinal Diseases Using a Tailored Web-Deployed Neural Network and Saliency Maps for *Explainable* AI. Frontiers in Medicine, 12.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How Transferable Are Features in Deep Neural Networks? Dalam Advances in Neural Information Processing Systems (NeurIPS), hlm. 3320–3328.

BAB 5
DETEKSI DAN ESTIMASI KEPARAHAN STROKE ISKEMIK PADA CITRA MRI OTAK: *PIPELINE* DUA TAHAP *DENSENET121-CBAM* DAN *NVAUTONET*

Setelah mempelajari bab ini, pembaca diharapkan mampu:

a. Menjelaskan karakteristik sekuens MRI DWI, ADC, dan FLAIR serta peranannya dalam mendeteksi stroke iskemik pada berbagai fase.
b. Merancang *pipeline* dua tahap yang menggabungkan klasifikasi sebagai penapisan awal dengan segmentasi sebagai analisis lanjutan, serta memahami cara galat merambat di antara keduanya.
c. Menguraikan prinsip *dense connectivity* pada *DenseNet121* dan mekanisme perhatian ganda pada *Convolutional Block Attention Module*.
d. Menerapkan rangkaian prapemrosesan citra MRI meliputi penyelarasan orientasi kanonik, pemotongan persentil, normalisasi, CLAHE, dan penapisan *Gaussian*.
e. Menilai secara kritis perbandingan antarkonfigurasi dataset, termasuk mengenali kapan dua model sesungguhnya tidak diuji pada himpunan uji yang setara.
f. Menjelaskan konsep ruang fisik pada berkas *NIfTI* serta mendiagnosis kegagalan metrik yang disebabkan ketidakselarasan ruang, bukan oleh model.
g. Menghitung volume lesi dari masker segmentasi dan memahami syarat yang harus dipenuhi sebelum angka volume dapat dijadikan dasar penentuan tingkat keparahan.

➢ Pendahuluan

Stroke merupakan gangguan fungsi otak yang terjadi secara mendadak akibat terganggunya suplai darah ke otak, baik karena penyumbatan pembuluh darah yang dikenal sebagai stroke iskemik maupun karena pecahnya pembuluh darah yang dikenal sebagai stroke hemoragik. Kedua kondisi tersebut menyebabkan jaringan otak kehilangan pasokan oksigen dan nutrisi, yang berujung pada

kematian sel dalam waktu yang sangat singkat. World Stroke Organization mencatat lebih dari 12 juta kasus stroke baru setiap tahun di seluruh dunia, dengan 6,5 juta di antaranya berakhir dengan kematian. Di Indonesia, Survei Kesehatan Indonesia tahun 2023 mencatat prevalensi stroke sebesar 8,3 per 1.000 penduduk dan menempatkannya sebagai penyebab kematian tertinggi di rumah sakit.

Beban yang ditimbulkan stroke tidak berhenti pada angka kematian. Laporan World Stroke Organization menyebutkan bahwa beban ekonomi global akibat stroke mencapai lebih dari 890 miliar dolar Amerika setiap tahun, atau sekitar 0,66% dari produk domestik bruto dunia, yang mencakup biaya perawatan medis, rehabilitasi jangka panjang, serta hilangnya produktivitas pasien. Analisis *Global Burden of Disease* tahun 2021 menunjukkan bahwa stroke menyumbang sekitar 160,5 juta *Disability-Adjusted Life Years* setiap tahun. Lebih dari separuh penyintas stroke mengalami kecacatan permanen yang memengaruhi kemampuan berbicara, bergerak, maupun menjalankan aktivitas sehari-hari.

Terdapat satu kata yang mendasari seluruh urgensi penanganan stroke, yaitu waktu. Pada stroke iskemik, jaringan otak yang kehilangan aliran darah mengalami kematian secara bertahap; inti infark terbentuk dengan cepat, sedangkan jaringan di sekitarnya yang masih dapat diselamatkan bertahan untuk sementara waktu sebelum turut mati. Terapi reperfusi memiliki jendela waktu yang sempit dan terukur dalam hitungan jam. Konsekuensinya, setiap menit yang terpakai untuk menunggu pembacaan citra merupakan menit yang hilang bagi jaringan yang sebenarnya masih dapat diselamatkan. Inilah yang membedakan bab ini dari empat bab sebelumnya: pada kasus *pneumonia*, tumor otak, morfologi spermatozoa, maupun penyakit retina, keterlambatan beberapa jam umumnya tidak mengubah luaran secara dramatis, sedangkan pada stroke keterlambatan tersebut menentukan segalanya.

Magnetic Resonance Imaging (MRI) merupakan modalitas yang paling diandalkan untuk keperluan ini karena mampu memberikan kontras jaringan lunak yang tinggi serta mendeteksi perubahan jaringan otak jauh lebih sensitif dibandingkan *Computed Tomography*, terutama pada fase hiperakut dan akut. Berbagai sekuens MRI menyediakan informasi yang saling melengkapi:

Diffusion Weighted Imaging sangat sensitif terhadap perubahan difusi air pada fase awal, *Apparent Diffusion Coefficient* membantu membedakan infark akut dari perubahan jaringan lain, sedangkan *Fluid Attenuated Inversion Recovery* memperjelas batas lesi sekaligus memberi petunjuk mengenai perkembangan infark. Meskipun demikian, interpretasi citra MRI masih dilakukan secara manual oleh radiolog, dan proses ini memerlukan waktu, bergantung pada pengalaman pemeriksa, serta berpotensi menimbulkan variasi antarpengamat terutama ketika lesi berukuran kecil atau berbatas tidak tegas.

Persoalan analisis pada bab ini juga lebih kompleks daripada bab-bab sebelumnya dalam hal jumlah keluaran yang dibutuhkan. Sistem tidak cukup hanya menyatakan apakah pasien mengalami stroke, sebab jawaban tersebut sudah diketahui ketika pasien tiba di unit gawat darurat. Yang dibutuhkan klinisi adalah lokasi lesi, luasnya, dan volumenya, karena ketiga hal itulah yang menentukan keputusan terapi dan prognosis. Dengan demikian, sistem harus menghasilkan keluaran pada tiga tingkat sekaligus: keputusan per irisan mengenai keberadaan lesi, pemetaan spasial batas lesi, dan angka kuantitatif berupa volume infark.

Menjalankan segmentasi pada seluruh volume MRI merupakan pekerjaan yang mahal secara komputasi karena prediksi harus dilakukan pada setiap *voxel*. Bab ini karena itu menerapkan pendekatan dua tahap. Tahap pertama menggunakan model klasifikasi *DenseNet121* yang dipadukan dengan *Convolutional Block Attention Module* untuk memilah setiap irisan MRI menjadi lesi dan non-lesi sebagai proses penapisan awal. Hanya irisan yang diprediksi mengandung lesi yang diteruskan ke tahap kedua, yaitu segmentasi menggunakan *NVAutoNet* dari kerangka kerja *DeepISLES*, yang menghasilkan masker lesi sekaligus menjadi dasar penghitungan volume infark. Rancangan semacam ini menghemat komputasi, tetapi memperkenalkan satu konsekuensi yang akan dibahas secara mendalam pada bab ini: galat pada tahap pertama tidak dapat diperbaiki oleh tahap kedua.

Bab ini juga memuat satu hal yang barangkali paling berharga di antara seluruh bab pada buku ini, dan temuan itu bukan mengenai arsitektur maupun akurasi. Pada percobaan awal, evaluasi segmentasi menghasilkan nilai *Dice Similarity Coefficient* sebesar 0,0866 angka yang praktis menyatakan bahwa model gagal total.

Penyebabnya ternyata sama sekali bukan model, melainkan ketidakselarasan sistem koordinat ruang fisik antara berkas prediksi dan berkas *ground truth*. Setelah penyelarasan dilakukan, nilai yang sama melonjak menjadi 0,8347. Kasus ini akan diuraikan secara rinci karena mewakili kelas kesalahan yang paling sulit dideteksi dalam pekerjaan citra medis, yaitu kesalahan yang tidak menimbulkan pesan galat apa pun dan hanya menampakkan diri sebagai angka yang buruk.

➢ Pengantar Teori

Bagian ini menguraikan landasan teori yang diperlukan pada bab ini. Konsep dasar mengenai konvolusi, *pooling*, fungsi aktivasi, *transfer learning*, dan metrik evaluasi dasar telah dibahas pada bab-bab sebelumnya dan hanya dirujuk seperlunya. Pembahasan difokuskan pada hal-hal yang khas bagi permasalahan stroke, yaitu karakteristik sekuens MRI, arsitektur *DenseNet* dan CBAM, teknik pra-pemrosesan dan regularisasi yang digunakan, serta konsep ruang fisik pada berkas medis volumetrik.

Stroke Iskemik dan Dimensi Waktu

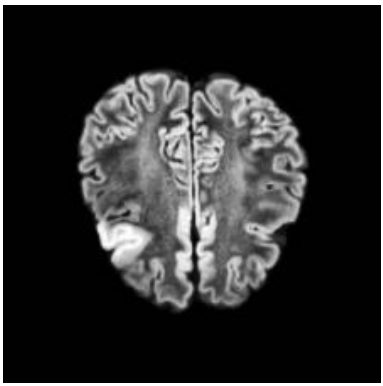
Stroke iskemik terjadi ketika sebuah pembuluh darah otak tersumbat sehingga aliran darah ke wilayah yang diperdarahinya terhenti. Wilayah yang terdampak terbagi menjadi dua bagian dengan nasib yang berbeda. Bagian inti atau *core* mengalami penurunan aliran darah yang paling parah dan selnya mati dalam hitungan menit; kerusakan pada bagian ini bersifat menetap. Di sekelilingnya terdapat *penumbra*, yaitu jaringan yang menerima aliran darah kolateral sehingga masih hidup namun tidak berfungsi, dan bagian inilah yang menjadi sasaran terapi. Apabila aliran darah dipulihkan tepat waktu, penumbra dapat diselamatkan; apabila tidak, penumbra akan bergabung menjadi bagian dari inti infark.

Kenyataan tersebut menjelaskan mengapa volume lesi menjadi ukuran yang begitu penting pada kasus stroke, dan mengapa ukuran itu berubah dari waktu ke waktu. Volume infark yang terukur pada citra bukanlah besaran tetap seperti ukuran sebuah tumor, melainkan potret sesaat dari proses yang masih berlangsung. Dua pasien dengan volume infark yang sama namun diperiksa pada jam yang berbeda sejak awitan gejala memiliki prognosis yang berbeda pula. Sebuah

sistem otomatis yang melaporkan volume tanpa menyertakan waktu pengambilan citra karena itu menyampaikan informasi yang tidak lengkap.

Sekuens MRI pada Stroke Iskemik

Diagnosis stroke iskemik pada MRI bertumpu pada tiga sekuens utama yang bekerja secara saling melengkapi. Contoh visualisasi citra MRI otak pada kasus yang mengandung lesi ditunjukkan pada Gambar 5.1.



Gambar 5.1 Citra MRI Otak pada Kasus dengan Lesi Stroke

Diffusion Weighted Imaging (DWI) bekerja dengan mendeteksi pergerakan molekul air di dalam jaringan otak. Ketika terjadi penyumbatan, pompa ion pada membran sel kehilangan pasokan energi sehingga air berpindah dari ruang antarsel ke dalam sel dan pergerakannya menjadi terbatas. Kondisi yang dikenal sebagai restriksi difusi ini membuat jaringan yang rusak tampak sebagai bercak putih terang atau hiperintens pada DWI hanya dalam hitungan menit setelah stroke terjadi. Metodologi Lansberg dan rekan membuktikan bahwa DWI mampu mendeteksi kasus stroke akut hingga 100%, sedangkan CT konvensional hanya mendeteksi sekitar 42% sampai 63% kasus.

Apparent Diffusion Coefficient (ADC) merupakan peta kuantitatif yang dihitung dari beberapa citra DWI dengan pembobotan difusi berbeda. Peran utamanya adalah memastikan bahwa bercak terang pada DWI benar-benar mencerminkan restriksi difusi dan bukan sekadar efek pembobotan T2 yang dikenal sebagai *T2 shine-through*. Pada ADC, area infark akut justru tampak gelap

atau hipointens. Kombinasi terang pada DWI dan gelap pada ADC merupakan penanda pasti stroke akut.

Fluid Attenuated Inversion Recovery (FLAIR) meredam sinyal cairan serebrospinal menjadi gelap sehingga edema dan perubahan jaringan di sekitar area yang rusak tampak terang secara kontras. FLAIR umumnya baru menunjukkan perubahan setelah empat hingga enam jam sejak awitan. Justru keterlambatan inilah yang membuatnya berguna secara klinis: lesi yang tampak terang pada DWI namun belum tampak pada FLAIR menandakan stroke berusia kurang dari sekitar enam jam, sehingga pasien masih berada dalam jendela terapi. Fenomena ini dikenal sebagai *DWI–FLAIR mismatch* dan digunakan untuk memperkirakan waktu awitan pada pasien yang ditemukan dalam keadaan sudah bergejala tanpa saksi.

Catatan: Ketiga Sekuens Bukan Tiga Sudut Pandang yang Setara

Sekilas, ketiga sekuens tersebut tampak seperti tiga cara berbeda memandang objek yang sama, sehingga menggabungkannya terkesan pasti menguntungkan. Kenyataannya lebih rumit.

ADC secara matematis diturunkan dari DWI, sehingga keduanya tidak menyimpan informasi yang sepenuhnya bebas satu sama lain. Lebih dari itu, hubungan keduanya bersifat *berlawanan*: lesi yang terang pada DWI justru gelap pada ADC. Apabila keduanya ditempatkan pada kanal berbeda dari satu citra berwarna, maka pada lokasi lesi satu kanal bernilai tinggi sementara kanal lainnya bernilai rendah.

FLAIR memiliki persoalan berbeda, yaitu ketergantungan pada waktu. Pada pasien hiperakut, FLAIR sengaja tidak menunjukkan apa-apa di lokasi lesi dan ketiadaan sinyal itu justru merupakan informasi diagnostik. Namun bagi sebuah model yang hanya melihat piksel, kanal yang kosong tidak dapat dibedakan dari kanal yang tidak informatif.

Kedua hal ini menjadi latar yang perlu diingat ketika pada Subbab 5.4.3 nanti terlihat bahwa pendekatan penggabungan tiga sekuens justru berperforma lebih rendah daripada penggunaan DWI saja.

Representasi Masukan: *DWI-Only* dan *RGB Fusion*

Arsitektur CNN yang dilatih pada *ImageNet* mengharapkan masukan berupa citra tiga kanal. Citra MRI pada dasarnya bersifat satu kanal, sehingga diperlukan strategi untuk menyesuaikannya. Bab ini membandingkan dua strategi tersebut.

Pendekatan pertama disebut *DWI-Only*, yaitu mereplikasi citra DWI menjadi tiga kanal identik sehingga terbentuk citra *pseudo-RGB*. Seluruh kanal memuat informasi yang sama, dan model pada dasarnya bekerja pada citra keabuan yang disamarkan sebagai citra berwarna. Pendekatan ini memanfaatkan sekuens yang paling sensitif terhadap infark akut, tetapi mengabaikan informasi dari kedua sekuens lainnya.

Pendekatan kedua disebut *RGB Fusion*, yaitu menempatkan DWI pada kanal merah, ADC pada kanal hijau, dan FLAIR pada kanal biru. Citra yang dihasilkan memuat informasi ketiga sekuens sekaligus sehingga model berpeluang mempelajari hubungan antarsekuens tanpa harus memproses masing-masing secara terpisah. Secara konseptual pendekatan ini tampak lebih unggul, namun terdapat satu asumsi tersembunyi yang perlu disadari. Bobot *ImageNet* dipelajari dari citra alami, dan pada citra alami ketiga kanal warna berkorelasi kuat: sebuah piksel yang terang cenderung terang pada ketiga kanalnya. Filter-filter pada lapisan awal dibentuk berdasarkan keteraturan itu. Pada citra hasil *RGB Fusion*, keteraturan tersebut tidak berlaku sama sekali, sehingga representasi *pretrained* bekerja pada jenis masukan yang tidak pernah ditemuinya.

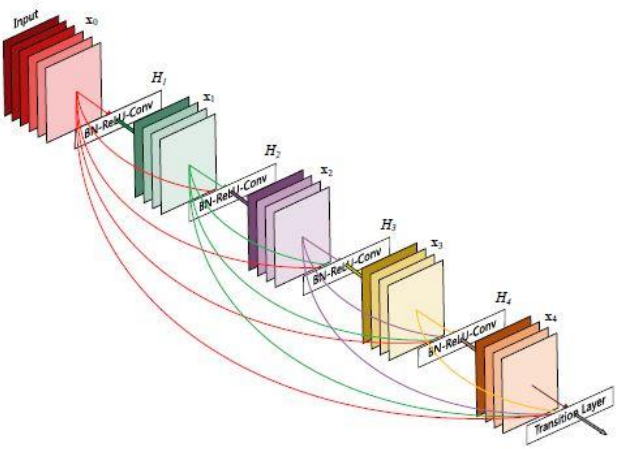
DenseNet121 dan Prinsip *Dense Connectivity*

DenseNet diperkenalkan oleh Huang dan rekan pada tahun 2017 sebagai pengembangan gagasan yang mendasari *ResNet*. Apabila *ResNet* menambahkan keluaran sebuah blok kepada masukannya melalui penjumlahan, *DenseNet* melangkah lebih jauh dengan menyambungkan setiap lapisan kepada seluruh lapisan sebelumnya melalui operasi penggabungan kanal. Hubungan antarlapisan pada *DenseNet* dinyatakan sebagai berikut.

$$x_l = H_l([\, x_0, x_1, \ldots, x_{l-1} \,]) \quad (5.1)$$

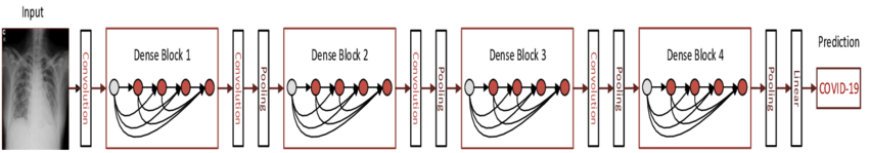
Dengan x indeks l menyatakan keluaran lapisan ke-l, H indeks l menyatakan fungsi komposit yang terdiri atas *Batch Normalization*, ReLU, dan operasi konvolusi, sedangkan tanda kurung siku menyatakan penggabungan seluruh keluaran lapisan sebelumnya sepanjang dimensi kanal. Perbedaan antara penjumlahan pada *ResNet* dan penggabungan pada *DenseNet* tampak kecil namun berdampak besar. Pada penjumlahan, informasi dari dua jalur bercampur menjadi satu dan tidak lagi dapat dipisahkan. Pada penggabungan, seluruh informasi tetap tersedia secara utuh bagi lapisan berikutnya, sehingga lapisan mana pun dapat memilih memanfaatkan fitur dari kedalaman mana pun.

Konsekuensi langsung dari rancangan tersebut adalah efisiensi parameter. Karena setiap lapisan memperoleh akses penuh terhadap seluruh fitur yang telah diekstraksi sebelumnya, lapisan tersebut tidak perlu mempelajari kembali representasi yang sudah ada dan cukup menambahkan sejumlah kecil peta fitur baru. Jumlah peta fitur yang ditambahkan setiap lapisan disebut *growth rate*, dan nilainya pada *DenseNet121* hanya 32. Akibatnya *DenseNet121* hanya memerlukan sekitar 8 juta parameter, jauh lebih sedikit daripada *ResNet*50 dengan sekitar 25 juta parameter maupun *VGG16* dengan sekitar 138 juta parameter, namun tetap mencapai performa yang setara atau lebih baik. Arsitektur dasarnya ditunjukkan pada Gambar 5.2.



Gambar 5.2 Arsitektur *DenseNet121*

DenseNet121 tersusun atas empat *dense block* yang dihubungkan oleh *transition layer*. Setiap unit di dalam *dense block* terdiri atas rangkaian *Batch Normalization*, ReLU, konvolusi 1×1 sebagai *bottleneck*, dan konvolusi 3×3. Konfigurasi jumlah lapisan pada keempat *dense block* adalah 6, 12, 24, dan 16, yang bersama lapisan transisi dan lapisan klasifikasi menghasilkan total 121 lapisan. *Transition layer* di antara blok berfungsi mengecilkan dimensi melalui konvolusi 1×1 dan *average pooling*, sehingga jumlah kanal tidak tumbuh tanpa kendali. Alur pemrosesan secara keseluruhan ditunjukkan pada Gambar 5.3.

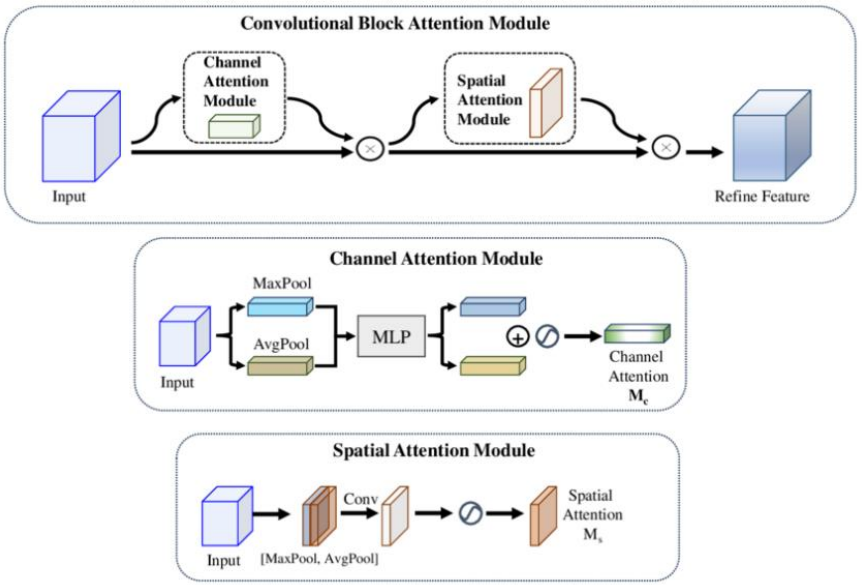


Gambar 5.3 Alur Pemrosesan *DenseNet121* pada Sistem Klasifikasi

Perlu dicatat bahwa efisiensi parameter tidak dengan sendirinya berarti efisiensi memori. Karena setiap lapisan memerlukan akses terhadap keluaran seluruh lapisan sebelumnya, seluruh peta fitur antara harus disimpan di memori selama proses pelatihan. *DenseNet* karena itu justru dikenal boros memori GPU meskipun jumlah parameternya kecil, dan hal ini merupakan pertukaran yang perlu diperhitungkan pada perangkat dengan sumber daya terbatas.

Convolutional Block Attention Module

Meskipun *DenseNet121* memiliki kemampuan ekstraksi fitur yang baik, model tetap berpotensi kehilangan informasi pada lesi berukuran kecil atau berkontras rendah. Persoalan ini bersifat mendasar pada CNN: operasi konvolusi memperlakukan seluruh posisi spasial dan seluruh kanal secara setara, padahal pada citra otak sebagian besar bidang citra berisi jaringan sehat yang tidak relevan bagi keputusan. *Convolutional Block Attention Module* (CBAM) yang diperkenalkan Woo dan rekan pada tahun 2018 mengatasi hal ini dengan menyisipkan dua tahap pembobotan secara berurutan. Arsitekturnya ditunjukkan pada Gambar 5.4.



Gambar 5.4 Arsitektur *Convolutional Block Attention Module*

Tahap pertama disebut *channel attention*, yang menjawab pertanyaan mengenai kanal fitur mana yang penting. Peta fitur masukan diringkas melalui *average pooling* dan *max pooling* pada seluruh dimensi spasial, sehingga setiap kanal terwakili oleh dua angka. Keduanya dilewatkan melalui satu *multilayer perceptron* yang sama, dijumlahkan, lalu diaktivasi dengan fungsi sigmoid.

$$M_c(F) = \sigma(\, MLP(AvgPool(F)) + MLP(MaxPool(F))\,) \quad (5.2)$$

Penggunaan dua jenis *pooling* sekaligus bukan tanpa alasan. *Average pooling* menangkap informasi mengenai seberapa aktif sebuah kanal secara keseluruhan, sedangkan *max pooling* menangkap informasi mengenai respons terkuat yang pernah muncul pada kanal tersebut. Pada kasus lesi berukuran kecil, sebuah kanal yang sangat responsif terhadap lesi akan memiliki nilai rata-rata yang rendah karena lesi hanya menempati sebagian kecil bidang, tetapi nilai maksimumnya tinggi. Tanpa *max pooling*, kanal semacam itu berisiko diabaikan.

Tahap kedua disebut *spatial attention*, yang menjawab pertanyaan mengenai lokasi mana yang penting. Peta fitur hasil tahap pertama diringkas melalui *average pooling* dan *max pooling* pada dimensi kanal, sehingga menghasilkan dua peta dua dimensi. Keduanya digabungkan, dilewatkan melalui konvolusi berkernel 7×7, dan diaktivasi dengan sigmoid.

$$M_s(F') = \sigma(f^{7\times7}([AvgPool(F') ; MaxPool(F')])) \quad (5.3)$$

Ukuran kernel 7×7 dipilih agar cakupan pandang cukup luas untuk menilai konteks di sekitar setiap posisi, sebab menentukan bahwa suatu lokasi penting menuntut perbandingan terhadap lingkungannya. Kedua peta perhatian tersebut kemudian diterapkan secara berurutan melalui perkalian elemen demi elemen.

$$F' = M_c(F) \otimes F, \quad F'' = M_s(F') \otimes F' \quad (5.4)$$

Urutan tersebut disengaja: penyaringan kanal dilakukan terlebih dahulu agar penentuan lokasi penting pada tahap berikutnya dihitung berdasarkan kanal-kanal yang sudah terbukti relevan. Karena nilai keluaran sigmoid berada pada rentang nol sampai satu, mekanisme ini hanya dapat menekan dan tidak dapat menguatkan; fitur yang dinilai tidak relevan didorong mendekati nol sedangkan fitur yang relevan dipertahankan mendekati nilai aslinya. Tambahan parameter yang dibutuhkan CBAM sangat kecil dibandingkan keseluruhan jaringan, sehingga peningkatan beban komputasinya dapat diabaikan.

Prapemrosesan Intensitas Citra MRI

Nilai piksel pada citra MRI, berbeda dari citra CT yang memiliki satuan Hounsfield yang terstandar, tidak memiliki satuan absolut. Nilai yang sama pada dua pemindai berbeda tidak merepresentasikan sifat jaringan yang sama. Karena itu standardisasi intensitas merupakan langkah yang wajib, bukan pilihan.

Langkah pertama adalah pemotongan nilai ekstrem menggunakan persentil. Nilai piksel yang berada di bawah persentil ke-1 dan di atas persentil ke-99 dipotong agar tidak memengaruhi tahap normalisasi berikutnya.

$$x' = clip(x, p_1, p_{99}) \quad (5.5)$$

Langkah ini penting karena satu piksel pencilan akibat artefak pemindai sudah cukup untuk merusak seluruh rentang normalisasi. Apabila nilai maksimum sebuah citra ditentukan oleh satu titik artefak yang nilainya sepuluh kali lipat jaringan normal, maka setelah normalisasi seluruh jaringan otak akan terkumpul pada rentang yang sangat sempit di dekat nol. Setelah pemotongan, dilakukan normalisasi min-maks sehingga seluruh nilai berada pada rentang nol sampai satu.

$$x'' = \frac{x\prime - min(x\prime)}{max(x\prime) - min(x\prime)} \quad (5.6)$$

Langkah berikutnya adalah peningkatan kontras menggunakan *Contrast Limited Adaptive Histogram Equalization* (CLAHE). Perataan histogram konvensional bekerja pada seluruh citra sekaligus dengan memetakan nilai intensitas melalui fungsi distribusi kumulatifnya.

$$T(k) = (L - 1) \sum_{i=0}^{k} p_i \quad (5.7)$$

Dengan L menyatakan jumlah aras keabuan dan p indeks i menyatakan proporsi piksel bernilai i. Pendekatan global semacam ini bermasalah pada citra otak karena sebagian besar bidang citra berisi latar belakang gelap, sehingga statistik histogram didominasi oleh area yang justru tidak menarik. CLAHE mengatasinya dengan membagi citra menjadi ubin-ubin kecil dan menghitung pemetaan tersendiri bagi setiap ubin. Selain itu, histogram setiap ubin dipotong pada suatu batas yang disebut *clip limit*, dan kelebihan nilainya didistribusikan kembali secara merata ke seluruh aras keabuan.

$$h_{clip}(i) = min(\, h(i), \beta \,) + \frac{1}{L} \sum_j max(\, h(j) - \beta, 0 \,) \quad (5.8)$$

Dengan β menyatakan *clip limit*. Pembatasan ini mencegah penguatan berlebihan pada area yang homogen, sebab tanpa pembatasan tersebut derau halus pada jaringan yang seragam akan ikut diperkuat menjadi tekstur palsu. Pada bab ini digunakan *clip limit* sebesar 0,02, yaitu nilai yang tergolong konservatif.

Langkah terakhir adalah penapisan *Gaussian* untuk meredam derau berfrekuensi tinggi yang mungkin diperkuat oleh CLAHE.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2 + y^2) / 2\sigma^2} \quad (5.9)$$

Nilai σ yang digunakan sebesar 0,5 tergolong sangat kecil dan hanya memengaruhi ketetanggaan piksel terdekat. Pemilihan nilai kecil ini penting karena penapisan yang terlalu kuat akan mengaburkan batas lesi, padahal ketajaman batas justru merupakan informasi yang dibutuhkan. Urutan CLAHE yang diikuti penapisan *Gaussian* karena itu merupakan pasangan yang saling melengkapi: yang pertama memperkuat kontras beserta deraunya, yang kedua meredam derau tanpa mengembalikan kontras yang telah diperoleh.

Fungsi Kerugian dan Penanganan Ketidakseimbangan

Permasalahan pada bab ini bersifat biner sehingga lapisan keluaran menggunakan satu neuron dengan aktivasi sigmoid sebagaimana telah dibahas pada Bab 1 melalui Persamaan 1.5. Fungsi kerugian yang digunakan adalah *binary cross-entropy*, namun dengan tiga penyesuaian yang perlu diuraikan.

Penyesuaian pertama bersifat teknis. Implementasi yang digunakan adalah *BCEWithLogitsLoss*, yaitu penggabungan fungsi sigmoid dan *binary cross-entropy* ke dalam satu operasi. Penggabungan ini bukan sekadar kepraktisan, melainkan menyangkut kestabilan numerik: menghitung sigmoid terlebih dahulu lalu mengambil logaritmanya berpotensi menghasilkan luapan aritmetika ketika nilai *logit* sangat besar, sedangkan bentuk gabungannya dapat dihitung secara stabil melalui identitas *log-sum-exp*.

Penyesuaian kedua adalah pembobotan kelas. Pada konfigurasi yang menggunakan seluruh irisan, jumlah irisan tanpa lesi jauh melebihi jumlah irisan berlesi, sehingga kontribusi kelas minoritas terhadap fungsi kerugian perlu diperbesar melalui bobot positif.

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[\, w\, y_i\, log(p_i) + (1 - y_i)\, log(1 - p_i)\,] \quad (5.10)$$

Dengan w menyatakan bobot kelas positif yang lazimnya ditetapkan sebagai rasio jumlah sampel negatif terhadap positif. Penyesuaian ketiga adalah *label smoothing*, yaitu menggeser label target dari nilai ekstrem nol dan satu menuju nilai yang sedikit lebih moderat.

$$y_{smooth} = y\,(1 - \varepsilon) + \frac{\varepsilon}{2} \quad (5.11)$$

Dengan ε menyatakan parameter penghalusan yang pada bab ini bernilai 0,08. Tujuannya adalah mencegah model menjadi terlalu yakin. Tanpa penghalusan, proses optimasi terus mendorong *logit* menuju nilai tak hingga demi mengejar probabilitas tepat satu, yang justru merusak kalibrasi keluaran sebagaimana telah dibahas pada Bab 4. Pada konteks medis, penghalusan label juga memiliki pembenaran tersendiri: label *ground truth* itu sendiri tidak sepenuhnya pasti, sebab irisan yang hanya memuat beberapa piksel lesi berada pada wilayah abu-abu antara kedua kelas.

Mixup sebagai Regularisasi

Mixup merupakan teknik regularisasi yang membentuk sampel pelatihan baru sebagai kombinasi linear dari dua sampel yang dipilih secara acak, beserta kombinasi linear labelnya.

$$x_{mix} = \lambda\, x_i + (1 - \lambda)\, x_j, \quad y_{mix} = \lambda\, y_i + (1 - \lambda)\, y_j \tag{5.12}$$

Dengan λ diambil dari distribusi Beta (α, α). Nilai α yang kecil, seperti 0,15 yang digunakan pada bab ini, membuat λ cenderung mendekati nol atau satu sehingga sebagian besar sampel hasil pencampuran masih sangat menyerupai salah satu sampel aslinya. Gagasan di balik teknik ini adalah memaksa model berperilaku linear di antara sampel-sampel pelatihan, sehingga keputusan yang dihasilkannya berubah secara halus dan tidak melompat tajam pada wilayah antara.

Penerapan *mixup* pada citra medis memerlukan kehati-hatian. Mencampur citra otak dengan lesi dan citra otak tanpa lesi menghasilkan citra yang secara anatomis mustahil, yaitu dua otak yang saling tembus pandang. Model yang dilatih pada citra semacam itu mempelajari sesuatu yang tidak pernah ada di dunia nyata. Pembenarannya terletak pada sifat regularisasi: tujuan *mixup* memang bukan menghasilkan citra yang realistis, melainkan memaksa perilaku model menjadi lebih teratur di antara titik-titik data. Nilai α yang kecil membatasi seberapa jauh pencampuran itu berlangsung, sehingga risikonya tetap terkendali.

AdamW dan Penjadwalan *Learning Rate*

Optimizer yang digunakan adalah AdamW, yaitu varian Adam dengan pemisahan *weight decay*. Adam memperbarui parameter berdasarkan estimasi momen pertama dan kedua dari gradien.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\, g_t \qquad (5.13)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)\, {g_t}^2 \qquad (5.14)$$

Kedua estimasi tersebut dikoreksi terhadap bias awal hasil koreksinya dinyatakan dengan tanda prima kemudian digunakan untuk memperbarui parameter. Perbedaan AdamW terletak pada penambahan suku peluruhan bobot secara terpisah, bukan melalui penambahan pada gradien.

$$\theta_{t+1} = \theta_t - \alpha \left(\frac{m\prime_t}{\sqrt{v\prime_t} + \epsilon} + \lambda\, \theta_t \right) \qquad (5.15)$$

Pemisahan ini penting karena pada Adam konvensional, penambahan suku peluruhan ke dalam gradien menyebabkan besarnya peluruhan ikut diskalakan oleh estimasi momen kedua. Akibatnya parameter dengan gradien besar justru mengalami peluruhan yang lebih lemah, yang merupakan kebalikan dari maksud regularisasi. AdamW mengembalikan perilaku peluruhan bobot sebagaimana mestinya.

Nilai *learning rate* dijadwalkan menggunakan *CosineAnnealingLR*, yang menurunkannya mengikuti kurva kosinus dari nilai awal menuju nilai minimum sepanjang sejumlah *epoch* yang ditetapkan.

$$\alpha_t = \alpha_{min} + \frac{1}{2} \left(\alpha_{max} - \alpha_{min} \right) \left(1 + cos\left(\frac{t\,\pi}{T_{max}} \right) \right) \qquad (5.16)$$

Bentuk kosinus memberikan penurunan yang lambat pada awal, curam di tengah, lalu melandai kembali menjelang akhir. Pola ini sesuai dengan kebutuhan proses optimasi: langkah besar dipertahankan cukup lama untuk menjelajah, lalu diperkecil secara halus agar model dapat menetap pada minimum tanpa berosilasi.

Perlu ditambahkan bahwa pada bab ini *learning rate* tidak seragam untuk seluruh bagian jaringan. Bagian *feature extractor* yang memuat bobot *ImageNet* menerima nilai sepersepuluh dari bagian CBAM dan kepala klasifikasi. Alasannya sama seperti yang telah dibahas pada Bab 2: bagian yang sudah memuat representasi

berguna cukup disesuaikan secara halus, sedangkan bagian yang dimulai dari bobot acak memerlukan langkah yang lebih besar.

Test-Time Augmentation

Test-Time Augmentation (TTA) merupakan teknik yang menerapkan augmentasi bukan pada saat pelatihan, melainkan pada saat inferensi. Sebuah citra uji ditransformasikan menjadi beberapa versi, masing-masing diprediksi secara terpisah, lalu hasilnya dirata-ratakan.

$$p_{TTA}(x) = \frac{1}{K}\sum_{k=1}^{K} f(T_k(x)) \qquad (5.17)$$

dengan p indeks TTA menyatakan probabilitas hasil perata-rataan, T indeks k menyatakan transformasi ke-k yang harus bersifat menjaga label, dan f menyatakan model. Perata-rataan atas beberapa prediksi menekan ragam sehingga keluaran menjadi lebih stabil. Perlu ditegaskan bahwa TTA tidak menjamin peningkatan pada seluruh metrik. Perata-rataan cenderung menarik probabilitas menuju nilai tengah, sehingga apabila ambang keputusan tetap dipertahankan pada 0,5 maka sebagian prediksi positif yang semula berada sedikit di atas ambang dapat bergeser ke bawahnya. Akibatnya *recall* menurun sementara *precision* meningkat. Fenomena inilah yang akan teramati pada Subbab 5.4.6.

Metrik Evaluasi Segmentasi

Metrik untuk tahap klasifikasi berupa *accuracy*, *precision*, *recall*, *F1-score*, dan AUC-ROC telah diuraikan pada Bab 1 dan Bab 4. Untuk tahap segmentasi digunakan *Dice Similarity Coefficient* yang telah dibahas pada Bab 2 melalui Persamaan 2.2, dan dituliskan kembali di sini dalam notasi yang sesuai.

$$DSC = \frac{2\,|S \cap G|}{|S| + |G|} \qquad (5.18)$$

Dengan S menyatakan himpunan *voxel* hasil segmentasi model dan G menyatakan himpunan *voxel* pada *ground truth*. Metrik pendamping berupa *Intersection over Union* dinyatakan sebagai berikut.

$$IoU = \frac{|S \cap G|}{|S \cup G|} (5.19)$$

Satu sifat DSC yang sangat menentukan pada kasus stroke perlu digarisbawahi, yaitu ketergantungannya pada ukuran objek. Pada lesi berukuran besar, kesalahan sebesar beberapa piksel pada tepinya hampir tidak menggeser nilai DSC. Pada lesi berukuran sangat kecil yang hanya memuat beberapa puluh *voxel*, kesalahan yang sama dapat menurunkan DSC secara drastis. Konsekuensinya, rata-rata DSC atas sekumpulan pasien tidak mencerminkan performa yang setara bagi semua ukuran lesi, dan pelaporan rata-rata tanpa disertai sebaran maupun stratifikasi menurut ukuran lesi berpotensi menyesatkan.

Estimasi Volume Lesi

Setelah masker segmentasi diperoleh, volume lesi dihitung dengan mengalikan jumlah *voxel* lesi dengan ukuran fisik satu *voxel* pada ketiga sumbu.

$$V = N_{voxel} \times S_x \times S_y \times S_z \qquad (5.20)$$

Dengan V menyatakan volume dalam milimeter kubik, N indeks *voxel* menyatakan jumlah *voxel* yang tersegmentasi sebagai lesi, serta S menyatakan ukuran *voxel* pada masing-masing sumbu dalam satuan milimeter. Nilai tersebut kemudian dikonversi ke satuan mililiter.

$$V_{mL} = \frac{V_{mm^3}}{1000} \quad (5.21)$$

Perhatikan bahwa nilai S tidak boleh diasumsikan, melainkan harus dibaca dari metadata citra. Sebuah pemindaian MRI otak yang lazim memiliki resolusi dalam bidang sekitar 1 milimeter namun ketebalan irisan antara 3 sampai 6 milimeter, sehingga *voxel*nya sangat tidak isotropik. Menganggap *voxel* berbentuk kubus akan menghasilkan kesalahan volume hingga beberapa kali lipat. Inilah alasan seluruh perhitungan pada bab ini dilakukan menggunakan pustaka SimpleITK yang membaca informasi tersebut langsung dari berkas.

Ruang Fisik pada Berkas *NIfTI*

Konsep terakhir pada bagian teori ini jarang dibahas pada buku *deep learning*, namun justru menjadi penyebab kegagalan paling serius pada bab ini. Berkas citra medis volumetrik seperti *NIfTI* tidak hanya menyimpan larik nilai *voxel*, tetapi juga sebuah matriks transformasi

afin yang memetakan indeks *voxel* menuju koordinat fisik dalam satuan milimeter pada ruang nyata.

$$[\, x, y, z, 1 \,]^{\mathrm{T}} = A \cdot [\, i, j, k, 1 \,]^{\mathrm{T}} \qquad (5.22)$$

dengan i, j, dan k menyatakan indeks *voxel*, x, y, dan z menyatakan koordinat fisik, serta A menyatakan matriks afin berukuran 4×4 yang memuat informasi mengenai orientasi, ukuran *voxel*, dan titik asal. Matriks inilah yang memungkinkan sebuah pemindaian diketahui posisi anatomisnya secara pasti, terlepas dari bagaimana lariknya tersusun di dalam berkas.

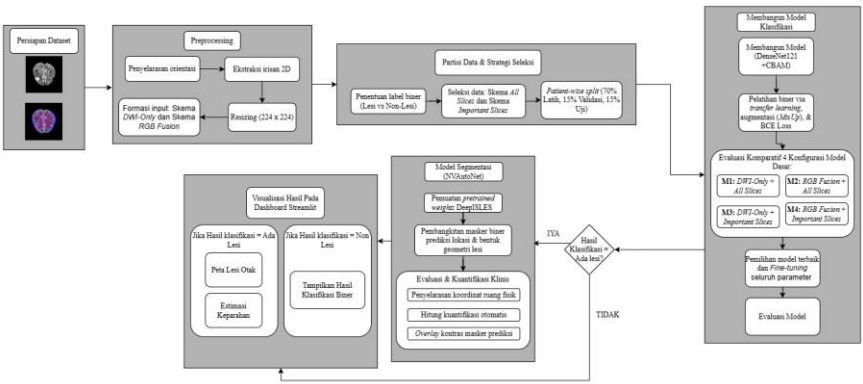
Konsekuensinya sangat mendasar: dua berkas dengan larik *voxel* yang sama persis dapat merepresentasikan objek pada posisi ruang yang sama sekali berbeda apabila matriks afinnya berbeda. Sebaliknya, dua berkas yang merepresentasikan objek pada posisi yang sama dapat memiliki susunan larik yang berbeda. Apabila perbandingan antara hasil segmentasi dan *ground truth* dilakukan pada tingkat indeks larik tanpa memeriksa matriks afinnya, maka yang dibandingkan sesungguhnya adalah dua objek pada posisi ruang yang berlainan. Nilai DSC yang dihasilkan akan mendekati nol, dan tidak akan ada pesan galat apa pun yang muncul.

Penyelarasan dilakukan melalui proses *resampling* ke ruang fisik yang sama, yaitu memetakan salah satu berkas ke dalam sistem koordinat berkas lainnya menggunakan matriks afin keduanya. Untuk data berupa masker biner, interpolasi yang digunakan harus berjenis *nearest neighbour*, sebab interpolasi linear akan menghasilkan nilai antara yang merusak sifat biner label. Rincian penerapannya beserta dampaknya terhadap hasil dibahas pada Subbab Kegagalan yang Bukan Kegagalan Model: Kasus *Space Mismatch*.

➢ Metodologi

Alur Metodologi

Keseluruhan sistem disusun sebagai satu alur kerja berurutan yang bercabang pada tahap keputusan klasifikasi. Alur tersebut ditunjukkan pada Gambar 5.5.



Gambar 5.5 Diagram Alir Metodologi

Proses diawali dengan persiapan dataset, dilanjutkan dengan prapemrosesan yang meliputi penyelarasan orientasi, ekstraksi irisan dua dimensi, pengubahan ukuran, serta pembentukan dua skema masukan yaitu *DWI-Only* dan *RGB Fusion*. Tahap berikutnya berupa penentuan label biner, penerapan dua strategi seleksi irisan, dan pembagian data secara *patient-wise*. Empat konfigurasi dataset yang terbentuk dievaluasi menggunakan model *DenseNet121-CBAM* untuk menentukan konfigurasi terbaik.

Titik keputusan terletak pada percabangan setelah klasifikasi. Apabila sebuah irisan dinyatakan mengandung lesi, irisan tersebut diteruskan ke tahap segmentasi menggunakan *NVAutoNet*, dilanjutkan dengan penyelarasan koordinat ruang fisik, kuantifikasi volume, dan penumpangan masker pada citra asli. Apabila sebuah irisan dinyatakan tidak mengandung lesi, prosesnya berhenti pada hasil klasifikasi biner. Keduanya bermuara pada dasbor *Streamlit* yang menyajikan hasil kepada pengguna.

Perhatian: Percabangan yang Bersifat Satu Arah
Perhatikan arah anak panah pada Gambar 5.5. Tidak terdapat jalur yang mengembalikan irisan bernilai negatif untuk diperiksa ulang. Sebuah irisan yang keliru dinyatakan tidak mengandung lesi oleh tahap pertama tidak akan pernah mencapai tahap segmentasi, dan lesi di dalamnya tidak akan pernah terhitung dalam estimasi volume.

Konsekuensi matematisnya tegas: *recall* pada tahap klasifikasi merupakan batas atas bagi kemampuan sistem secara keseluruhan dalam menemukan lesi. Betapapun sempurnanya model segmentasi, ia tidak dapat menemukan sesuatu yang tidak pernah diteruskan kepadanya.

Inilah pertukaran mendasar dari rancangan dua tahap. Penghematan komputasi diperoleh dengan cara tidak memproses sebagian data, dan sebagian data yang tidak diproses itu tidak seluruhnya benar-benar kosong. Rancangan semacam ini menjadi sah apabila ambang keputusan pada tahap pertama disetel untuk mengutamakan *recall* dengan sengaja menerima lebih banyak alarm palsu demi memastikan tidak ada lesi yang terlewat, sebab alarm palsu masih dapat disaring oleh tahap kedua sedangkan lesi yang terlewat tidak dapat dipulihkan.

Dataset dan Variabel Metodologi

Data utama yang digunakan adalah dataset publik ISLES 2022 (*Ischemic Stroke Lesion Segmentation Challenge*). Dataset ini menyediakan data dalam format *NIfTI* berupa volume tiga dimensi, dengan setiap kasus pasien memuat sekuens DWI, ADC, dan FLAIR beserta masker penanda lesi yang dianotasi oleh ahli neuroradiologi. Komponen data pada Metodologi ini dibagi ke dalam tiga variabel operasional sebagaimana dirangkum pada Tabel 5.1.

Tabel 5.1 Variabel Operasional Metodologi

Variabel	**Bentuk Data**	**Keterangan**
Prediktor	Irisan 2D dari sekuens DWI, ADC, FLAIR	Masukan model klasifikasi
Target klasifikasi	Label biner lesi dan non-lesi	Diturunkan dari masker ground truth
Target segmentasi	Masker biner tiga dimensi	Anotasi ahli neuroradiologi

Selain dataset publik, Metodologi ini merencanakan penggunaan data uji independen berupa citra MRI otak pasien stroke iskemik yang diperoleh melalui kerja sama dengan RSUD Haji Provinsi Jawa Timur, Surabaya. Data tersebut berasal dari mesin pemindai yang berbeda dari data pelatihan, sehingga sesuai untuk menguji ketangguhan model terhadap variasi karakteristik akuisisi di lapangan. Seluruh data pasien telah melalui proses anonimisasi sesuai prosedur kaji laik etik rumah sakit.

Catatan: Nilai Validasi Eksternal
Rencana validasi eksternal ini merupakan kekuatan rancangan Metodologi yang membedakannya dari keempat bab sebelumnya, yang seluruhnya hanya menggunakan dataset publik tunggal.
Alasannya penting untuk dipahami. Sebuah model yang dilatih pada ISLES 2022 mempelajari bukan hanya ciri lesi stroke, tetapi juga ciri khas rangkaian pemindai yang menghasilkan dataset tersebut pola derau, karakteristik kontras, protokol akuisisi, dan bahkan komposisi populasi pasiennya. Ketika diterapkan pada data dari rumah sakit lain, sebagian ciri yang dipelajari itu tidak lagi berlaku. Penurunan performa merupakan hal yang lazim terjadi dan tidak selalu menandakan model yang buruk.
Yang perlu ditegaskan adalah bahwa validasi eksternal hanya bermakna apabila dilaporkan secara utuh, termasuk ketika hasilnya mengecewakan. Angka performa pada data rumah sakit yang lebih rendah daripada data publik justru merupakan informasi paling berharga bagi calon pengguna sistem, sebab angka itulah yang mendekati kondisi penggunaan nyata.

Prapemrosesan Citra MRI

Tahap prapemrosesan bertujuan menyeragamkan orientasi, dimensi, dan karakteristik intensitas seluruh volume MRI. Proses dilakukan pada ketiga modalitas secara berurutan melalui langkah-langkah berikut.

a. Penyelarasan orientasi. Seluruh volume diubah ke sistem koordinat RAS menggunakan fungsi *as_closest_canonical* dari pustaka NiBabel, sehingga seluruh data memiliki orientasi

spasial yang konsisten terlepas dari perbedaan posisi kepala maupun perangkat saat akuisisi.

b. Ekstraksi irisan. Setiap volume diuraikan menjadi kumpulan irisan aksial dua dimensi. Irisan dengan cakupan jaringan otak kurang dari 5% disingkirkan karena tidak memuat informasi anatomi yang memadai. Apabila jumlah irisan antarmodalitas berbeda, dilakukan interpolasi agar posisi anatomis setiap irisan bersesuaian.
c. Pengubahan ukuran dan normalisasi. Setiap irisan diubah menjadi 224 × 224 piksel sesuai masukan *DenseNet121*, kemudian dinormalisasi menggunakan Persamaan 5.6 setelah terlebih dahulu dipotong pada persentil ke-1 dan ke-99 sesuai Persamaan 5.5.
d. Peningkatan kontras. CLAHE diterapkan dengan *clip limit* sebesar 0,02 untuk memperjelas area lesi terhadap jaringan sekitarnya, kemudian disusul penapisan *Gaussian* dengan σ sebesar 0,5 untuk meredam derau yang ikut diperkuat.

Pembacaan berkas *NIfTI* beserta penyelarasan orientasinya ditunjukkan pada Kode 5.1.

```
def load_nifti(path):
    try:
        img = nib.as_closest_canonical(nib.load(path))
        data = img.get_fdata(dtype=np.float32)
        if data.ndim == 4:
            data = data[:, :, :, -1]
        return data
    except:
        return None
```

Kode 5.1 Pembacaan dan Penyelarasan Orientasi Berkas NIfTI

Perhatikan penanganan volume berdimensi empat pada kode tersebut. Sebagian berkas DWI menyimpan beberapa volume dengan nilai pembobotan difusi yang berbeda, dan volume terakhir umumnya merepresentasikan b-value tinggi yang paling sensitif terhadap infark akut. Pemilihan volume terakhir karena itu bukan keputusan sembarangan, melainkan pemilihan sekuens yang paling informatif secara klinis.

Rangkaian prapemrosesan pada tingkat irisan ditunjukkan pada Kode 5.2.

```
def preprocess_slice(sl, size=224):
    # Step 1: Resize
    img_s1 = resize(img, (size, size))
    # Step 2: Normalisasi min-max
    img_s2 = (img_s1 - img_s1.min()) / (img_s1.max() - img_s1.min())
    # Step 3a: CLAHE
    img_s3 = exposure.equalize_adapthist(img_s2, clip_limit=0.02)
    # Step 3b: Gaussian Filter
    img_s3 = ndi.gaussian_filter(img_s3, sigma=0.5)
    return img_s3
```

Kode 5.2 Rangkaian Prapemrosesan pada Tingkat Irisan

Pembentukan Dataset dan Strategi Seleksi Irisan

Pelabelan dilakukan secara otomatis berdasarkan masker *ground truth*. Setiap irisan dihitung rasio area lesinya sebagai perbandingan jumlah piksel bernilai positif terhadap total piksel citra, kemudian dibandingkan terhadap suatu ambang. Irisan dengan rasio di atas ambang diberi label lesi, sedangkan sisanya diberi label non-lesi.

```
for z in range(n):
    dwi_sl = dwi_vol[:, :, z]
    # Filter slice hampir kosong
    brain_cov = np.count_nonzero(dwi_sl) / dwi_sl.size
    if brain_cov < 0.05:
        continue
    mask_sl = mask_vol[:, :, z]
    lesi_ratio = np.sum(mask_sl > 0.5) / mask_sl.size
    label = "lesi" if lesi_ratio >= LESI_THRESH else "non-lesi"
```

Kode 5.3 Pelabelan Irisan Berdasarkan Rasio Area Lesi

Penggunaan ambang, dan bukan sekadar keberadaan satu piksel positif, bertujuan memisahkan irisan yang benar-benar memuat infark dari irisan yang hanya memiliki derau anotasi berukuran sangat kecil. Perlu disadari bahwa penetapan ambang ini

merupakan keputusan yang berdampak langsung terhadap definisi kedua kelas. Ambang yang terlalu rendah memasukkan irisan tepi lesi yang secara visual nyaris tidak dapat dibedakan dari jaringan normal ke dalam kelas positif, sehingga model dipaksa mempelajari perbedaan yang tidak tampak. Ambang yang terlalu tinggi menyingkirkan lesi kecil dari kelas positif, sehingga model tidak pernah belajar mengenalinya.

Selanjutnya dibentuk dua skema masukan. Pada skema *DWI-Only*, citra DWI direplikasi menjadi tiga kanal identik. Pada skema *RGB Fusion*, ketiga modalitas digabungkan menjadi satu citra tiga kanal.

```
rgb = np.stack(
    [dwi_proc[z], adc_proc[z], flair_proc[z]],
    axis=0
)
```

Kode 5.4 Pembentukan Citra *RGB Fusion*

Kedua skema masukan tersebut kemudian dikombinasikan dengan dua strategi seleksi irisan. Strategi *All Slices* menggunakan seluruh irisan yang lolos seleksi cakupan otak. Strategi *Important Slices* hanya memilih irisan yang dinilai paling representatif: pada kelas lesi dipilih irisan dengan rasio lesi terbesar, sedangkan pada kelas non-lesi dipilih irisan di bagian tengah volume yang paling banyak memuat struktur otak utama.

```
# Lesi terbesar
slices_lesi_sorted = sorted(slices_lesi, key=lambda x: x[3],
                            reverse=True)
imp_lesi = slices_lesi_sorted[:MAX_LESI_IMP]

# Slice tengah untuk non-lesi
mid = len(slices_nolesi) // 2
half = MAX_NOLESI_IMP // 2
imp_nolesi = slices_nolesi[max(0, mid - half):
                           min(len(slices_nolesi), mid + half)]
```

Kode 5.5 Pemilihan *Important Slice*

Kombinasi kedua skema masukan dan kedua strategi seleksi menghasilkan empat konfigurasi dataset sebagaimana disajikan pada Tabel 5.2.

Tabel 5.2 Empat Konfigurasi Dataset yang Dibandingkan

Model	**Jenis Masukan**	**Strategi Irisan**	**Keterangan**
M1	DWI-*Only*	*All Slices*	Satu modalitas, seluruh irisan
M2	*RGB Fusion*	*All Slices*	Tiga modalitas, seluruh irisan
M3	DWI-*Only*	*Important Slices*	Satu modalitas, irisan terpilih
M4	*RGB Fusion*	*Important Slices*	Tiga modalitas, irisan terpilih

Perhatian: Seleksi Irisan Mengubah Himpunan Uji, Bukan Hanya Himpunan Latih

Strategi *Important Slices* memiliki pembenaran yang masuk akal untuk data pelatihan: memusatkan perhatian model pada irisan yang informatif dan mengurangi dominasi irisan kosong. Persoalannya, seleksi yang sama juga diterapkan pada data pengujian. Akibatnya M1 dan M2 diuji pada himpunan yang memuat seluruh irisan termasuk irisan dengan lesi berukuran sangat kecil yang sulit dikenali, sedangkan M3 dan M4 diuji pada himpunan yang kelas positifnya hanya terdiri atas irisan dengan lesi terbesar, yaitu kasus yang paling mudah.

Konsekuensinya, keempat model tidak diuji pada permasalahan yang setara. Perbandingan angka di antara M1 dan M3 karena itu tidak dapat dibaca sebagai perbandingan kualitas model, melainkan sebagian mencerminkan perbedaan tingkat kesulitan himpunan ujinya. Hal ini akan ditunjukkan secara kuantitatif pada Subbab 5.4.5.

Rancangan yang benar adalah menerapkan seleksi hanya pada data pelatihan, sementara seluruh model dievaluasi pada satu himpunan uji yang sama dan mencerminkan distribusi nyata. Dengan cara itu, pertanyaan yang dijawab menjadi jelas: apakah

melatih pada irisan terpilih menghasilkan model yang lebih baik ketika diuji pada kondisi sebenarnya.

Pembagian Data

Pembagian data dilakukan menggunakan pendekatan *patient-wise split* dengan rasio 70% untuk pelatihan, 15% untuk validasi, dan 15% untuk pengujian. Pembagian dilakukan pada tingkat pasien sehingga seluruh irisan milik satu pasien hanya berada pada satu kelompok data.

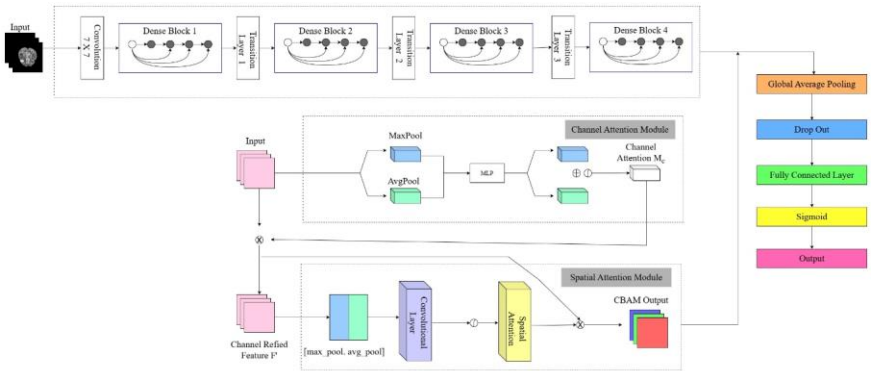
```
idx_train, idx_temp = train_test_split(all_idx,
                                       test_size=0.30,
                                       random_state=SEED)
idx_val, idx_test = train_test_split(idx_temp,
                                     test_size=0.50,
                                     random_state=SEED)
```

Kode 5.6 Pembagian Data secara *Patient-wise*

Keputusan membagi pada tingkat pasien merupakan hal yang benar dan patut ditekankan. Sebagaimana telah dibahas pada Bab 2, dataset MRI memuat puluhan irisan berdekatan dari setiap pasien yang saling menyerupai. Pembagian acak pada tingkat irisan akan menempatkan irisan ke-14 dan ke-15 milik pasien yang sama pada kelompok yang berbeda, sehingga model dinilai pada citra yang nyaris identik dengan yang telah dipelajarinya. Hasil evaluasinya akan tampak sangat baik namun tidak mencerminkan performa pada pasien baru.

Arsitektur *DenseNet121*-CBAM

Model klasifikasi dibangun dengan menempatkan CBAM di antara *feature extractor DenseNet121* dan kepala klasifikasi. Arsitektur lengkapnya ditunjukkan pada Gambar 5.6.



Gambar 5.6 Arsitektur *DenseNet121* dengan Integrasi Modul CBAM

Citra masukan berukuran 224 × 224 diproses melalui konvolusi awal 7×7, kemudian melewati empat *dense block* yang dihubungkan oleh tiga *transition layer*. Peta fitur keluaran dengan 1024 kanal diteruskan ke modul CBAM, yang menerapkan *channel attention* dilanjutkan *spatial attention* sesuai Persamaan 5.2 sampai 5.4. Hasilnya diringkas melalui *global average pooling*, kemudian melewati kepala klasifikasi berupa *dropout*, lapisan terhubung penuh dengan 512 neuron beserta *batch normalization* dan ReLU, *dropout* kedua, dan akhirnya satu neuron keluaran. Implementasinya ditunjukkan pada Kode 5.7.

```
class DenseNet121CBAM(nn.Module):
    def __init__(self, drop=DROPOUT, freeze=True):
        super().__init__()
        base = models.densenet121(
            weights=DenseNet121_Weights.IMAGENET1K_V1)
        self.features = base.features
        self.relu = nn.ReLU(inplace=True)
        self.cbam = CBAM(1024)
        self.gap = nn.AdaptiveAvgPool2d(1)
        self.head = nn.Sequential(
            nn.Dropout(drop),
            nn.Linear(1024, 512),
            nn.BatchNorm1d(512),
            nn.ReLU(True),
            nn.Dropout(drop * 0.5),
            nn.Linear(512, 1))
```

Kode 5.7 Implementasi Arsitektur *DenseNet121-CBAM*

Modul CBAM sendiri diimplementasikan sebagai penerapan berurutan dari kedua submodul perhatian.

```
class CBAM(nn.Module):
    def __init__(self, c, r=16, k=7):
        super().__init__()
        self.ca = ChannelAttention(c, r)
        self.sa = SpatialAttention(k)

    def forward(self, x):
        x = x * self.ca(x)
        x = x * self.sa(x)
        return x
```

Kode 5.8 Implementasi Modul CBAM

Penempatan CBAM setelah seluruh *dense block*, dan bukan disisipkan di antara setiap blok, merupakan keputusan yang perlu dicatat. Penempatan di ujung berarti mekanisme perhatian bekerja pada peta fitur dengan resolusi spasial paling rendah, yaitu 7×7 untuk masukan berukuran 224×224. Setiap sel pada peta perhatian spasial karena itu mewakili wilayah seluas 32×32 piksel pada citra asli. Untuk lesi berukuran besar hal ini memadai, tetapi untuk lesi kecil seluruh lesi mungkin hanya menempati satu sel. Penempatan alternatif berupa penyisipan CBAM setelah setiap *dense block* akan memberikan perhatian pada beberapa tingkat resolusi sekaligus, dengan konsekuensi bertambahnya beban komputasi.

Pada tahap awal pelatihan, sebagian lapisan *DenseNet121* dibekukan untuk mempertahankan representasi *ImageNet*.

```
if freeze:
    for n, p in self.features.named_parameters():
        if any(n.startswith(x) for x in [
            "conv0", "norm0",
            "denseblock1", "transition1",
            "denseblock2", "transition2"]):
            p.requires_grad = False
```

Kode 5.9 Pembekuan Lapisan Awal *DenseNet121*

Lapisan yang dibekukan mencakup konvolusi awal hingga *dense block* kedua, yaitu bagian yang mempelajari fitur dasar berupa tepi, tekstur, dan pola umum. Fitur semacam ini bersifat universal dan tetap berlaku pada citra MRI meskipun dipelajari dari citra alami. Sebaliknya, *dense block* ketiga dan keempat yang mempelajari pola tingkat tinggi dibiarkan terbuka agar dapat menyesuaikan diri dengan karakteristik khas citra otak.

Konfigurasi Pelatihan

Konfigurasi lengkap pelatihan model klasifikasi disajikan pada Tabel 5.3.

Tabel 5.3 Konfigurasi Pelatihan Model *DenseNet121-CBAM*

Parameter	**Nilai**	**Alasan Pemilihan**
Arsitektur	*DenseNet121* + CBAM	Efisien parameter dengan perhatian ganda
Transfer learning	*Pretrained ImageNet*	Mempercepat dan menstabilkan konvergensi
Ukuran masukan	224 × 224 piksel	Sesuai masukan bawaan *DenseNet121*
Batch size	32	Menyeimbangkan kestabilan dan memori
Optimizer	*AdamW*	Peluruhan bobot terpisah dari gradien
Scheduler	*CosineAnnealingLR*	Penurunan *learning rate* yang halus
Fungsi kerugian	B*CEWithLogitsLoss* + Label *Smoothing*	Stabil secara numerik dan mencegah *overconfidence*
Label *smoothing*	$\varepsilon = 0{,}08$	Mengakui ketidakpastian pada label tepi
Mixup	$\alpha = 0{,}15$	Regularisasi antar-sampel yang terkendali
Dropout	0,35	Regularisasi pada kepala klasifikasi

Epoch maksimum	50–60	Batas atas, dibatasi *early stopping*
Early stopping	*Patience* = 10 pada F1 validasi	Menghentikan saat performa stagnan
Penyeimbangan kelas	*Weighted Random Sampler*	Mengatasi dominasi kelas non-lesi
Augmentasi	*Flip*, rotasi, *zoom*, *brightness*, *blur*	Meningkatkan ketangguhan model

Augmentasi yang diterapkan pada data pelatihan ditunjukkan pada Kode 5.10.

```
TRAIN_TF = transforms.Compose([
    transforms.RandomHorizontalFlip(0.5),
    transforms.RandomVerticalFlip(0.3),
    transforms.RandomRotation(20),
    transforms.RandomAffine(0, scale=(0.88, 1.12)),
    transforms.ColorJitter(brightness=0.2, contrast=0.15),
    transforms.GaussianBlur(3, sigma=(0.1, 0.8)),
    transforms.Normalize(IMAGENET_MEAN, IMAGENET_STD)
])
```

Kode 5.10 Augmentasi Data Pelatihan

Sebagian besar transformasi tersebut sesuai dengan karakteristik citra otak. *Horizontal flip* menghasilkan citra yang menyerupai lesi pada hemisfer berlawanan dan masih realistis secara anatomis. Rotasi hingga 20 derajat mengakomodasi variasi kemiringan kepala saat pemindaian. Penyesuaian kecerahan dan kontras meniru variasi antarperangkat. Namun *vertical flip* dengan peluang 0,3 patut dipertanyakan: irisan aksial otak memiliki orientasi anterior-posterior yang bermakna, dengan lobus frontal di bagian atas dan serebelum atau lobus oksipital di bagian bawah. Pembalikan vertikal menghasilkan susunan yang tidak pernah ditemui pada praktik. Meskipun demikian, karena tugas yang dihadapi berupa deteksi keberadaan lesi dan bukan penentuan lokasi anatomisnya, dampaknya kemungkinan terbatas.

Pengaturan *optimizer* dengan *learning rate* berbeda untuk setiap bagian jaringan ditunjukkan pada Kode 5.11.

```
opt = optim.AdamW([
   {"params": model.features.parameters(), "lr": LR_P1 * 0.1},
   {"params": model.cbam.parameters(),     "lr": LR_P1},
   {"params": model.head.parameters(),     "lr": LR_P1}
], weight_decay=1e-3)

sched = optim.lr_scheduler.CosineAnnealingLR(
   opt, T_max=FREEZE_EP, eta_min=1e-6)
```

Kode 5.11 Konfigurasi *Optimizer* dan *Scheduler*

Perhatikan pula bahwa *early stopping* pada bab ini memantau *F1-score* validasi, bukan *validation loss*. Pilihan ini tepat untuk permasalahan dengan kelas tidak seimbang, sebab *loss* dapat membaik melalui peningkatan keyakinan pada kelas mayoritas tanpa perbaikan apa pun pada kemampuan mendeteksi kelas minoritas.

Segmentasi Menggunakan *NVAutoNet*

Tahap segmentasi menggunakan *NVAutoNet*, salah satu komponen kerangka kerja DeepISLES yang merupakan gabungan metode berperforma terbaik pada kompetisi ISLES 2022. DeepISLES sendiri menggabungkan tiga algoritma heterogen melalui mekanisme pemungutan suara mayoritas pada tingkat piksel, yaitu SEALS yang berbasis modifikasi nnU-Net, NVAUTO yang berbasis MONAI Auto3DSeg, dan SWAN yang berbasis arsitektur *Factorizer*. Integrasi ketiganya dilaporkan meningkatkan *Dice Coefficient* sebesar 7,4% dibandingkan penggunaan model tunggal.

Bab ini memanfaatkan komponen NVAUTO saja, dengan pertimbangan kestabilannya dalam memproses modalitas DWI dan ADC secara simultan serta beban komputasinya yang lebih ringan daripada keseluruhan gabungan. Model digunakan dalam bentuk bobot terlatih tanpa pelatihan ulang. Konfigurasi inferensinya disajikan pada Tabel 5.4.

Tabel 5.4 Konfigurasi Inferensi Segmentasi *NVAutoNet*

Parameter	**Konfigurasi**
Kerangka kerja	*Deep*ISLES
Model	*NVAutoNet*
Jenis model	*Pretrained TorchScript*
Jumlah *checkpoint*	15 model ensemble
Modalitas masukan	ADC dan DWI
Dimensi masukan	3D volumetrik
Pustaka	MONAI dan *PyTorch*
Perangkat	GPU CUDA
Keluaran	Masker lesi biner

```
model = torch.jit.load(checkpoint_path).to(device)
model.eval()

self.checkpoints = sorted(
   glob.glob(os.path.join(weights_dir, "model*.ts")))
print(f"Loaded {len(self.checkpoints)} checkpoint files")
```

Kode 5.12 Pemuatan Model *NVAutoNet* Terlatih

Catatan: Ketegangan antara Klasifikasi 2D dan Segmentasi 3D

Terdapat satu ketidakserasian yang perlu dinyatakan secara terbuka pada rancangan ini. Model klasifikasi bekerja pada irisan dua dimensi secara terpisah, sedangkan *NVAutoNet* merupakan model tiga dimensi yang mengandalkan konteks antariris untuk menghasilkan segmentasi yang konsisten.

Ketika hanya irisan positif yang diteruskan lalu disusun kembali menjadi volume, susunan yang terbentuk berpotensi tidak kontinu, misalnya irisan ke-8, ke-9, ke-12, dan ke-13 tanpa irisan ke-10 dan ke-11 di antaranya. Bagi model tiga dimensi, susunan semacam itu menyajikan hubungan spasial yang keliru, seolah irisan ke-9 dan ke-12 bersebelahan secara anatomis.

Klaim penghematan memori juga perlu ditakar ulang. Model tiga dimensi umumnya memproses volume melalui *sliding window* dengan ukuran petak tetap, sehingga kebutuhan memori puncaknya ditentukan oleh ukuran petak dan bukan oleh jumlah

irisan. Yang dihemat sesungguhnya adalah waktu inferensi, bukan memori puncak.
Penyelesaian yang lebih baik adalah memanfaatkan hasil klasifikasi untuk menentukan *rentang* irisan yang perlu diproses dari irisan positif pertama hingga terakhir beserta beberapa irisan penyangga di kedua ujungnya sehingga kontinuitas volume tetap terjaga sekaligus tetap memangkas sebagian besar volume yang tidak relevan.

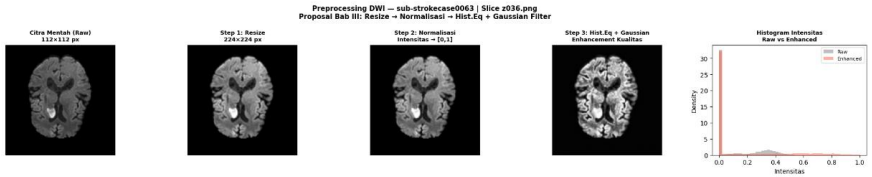
Kuantifikasi Volume Lesi

Tahap akhir menghitung volume lesi dari masker segmentasi menggunakan Persamaan 5.20 dan 5.21. Informasi ukuran *voxel* dibaca dari metadata citra menggunakan pustaka SimpleITK, sehingga hasil pengukuran tetap sahih meskipun citra berasal dari perangkat dengan resolusi spasial yang berbeda. Hasil perhitungan kemudian ditumpangkan pada citra MRI asli untuk mem-visualisasikan lokasi dan bentuk lesi.

➢ Hasil dan Ujicoba

Hasil Prapemrosesan

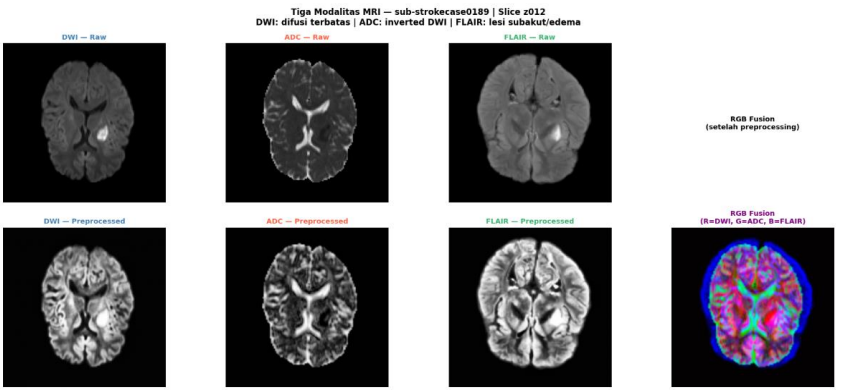
Pengaruh setiap tahapan prapemrosesan terhadap kualitas visual citra MRI ditunjukkan pada Gambar 5.7.



Gambar 5.7 Perubahan Citra MRI pada Setiap Tahapan Prapemrosesan

Setelah penerapan CLAHE dan penapisan *Gaussian*, area lesi tampak jauh lebih kontras dibandingkan citra mentah, dan detail jaringan otak menjadi lebih jelas. Histogram distribusi intensitas juga memperlihatkan sebaran yang lebih merata dibandingkan citra awal, yang menandakan bahwa rentang dinamis citra telah dimanfaatkan secara lebih penuh. Perbandingan karakteristik ketiga

modalitas beserta hasil penggabungannya ditunjukkan pada Gambar 5.8.



Gambar 5.8 Perbandingan Tiga Modalitas MRI dan Hasil *RGB Fusion*

Gambar tersebut memperlihatkan dengan jelas apa yang telah dibahas secara teoretis pada Subbab 5.2.2. Pada baris atas, DWI mentah menampilkan lesi sebagai bercak terang pada hemisfer kiri, ADC menampilkan area yang bersesuaian sebagai bercak gelap, sedangkan FLAIR menampilkan perubahan yang lebih samar. Setelah prapemrosesan pada baris bawah, ketiganya menunjukkan peningkatan kontras yang nyata.

Panel paling kanan menampilkan hasil *RGB Fusion*, dan panel inilah yang paling patut diamati secara cermat. Citra yang dihasilkan didominasi warna biru dan ungu, dengan area lesi tampak sebagai bercak merah muda. Warna biru yang mendominasi berasal dari kanal FLAIR yang bernilai tinggi pada sebagian besar jaringan otak. Yang lebih penting, pada area lesi terlihat bahwa kanal merah bernilai tinggi sementara kanal hijau bernilai rendah persis kondisi saling berlawanan antara DWI dan ADC yang telah diantisipasi sebelumnya. Bagi mata manusia, kombinasi ini menghasilkan warna khas yang mudah dikenali. Bagi jaringan yang dilatih pada citra alami, kombinasi semacam ini justru merupakan pola yang tidak pernah ditemuinya, sebab pada citra alami ketiga kanal cenderung bergerak searah.

Distribusi Dataset pada Empat Konfigurasi

Hasil pembagian data pada tingkat pasien disajikan pada Tabel 5.5.

Tabel 5.5 Pembagian Data Berdasarkan Pasien

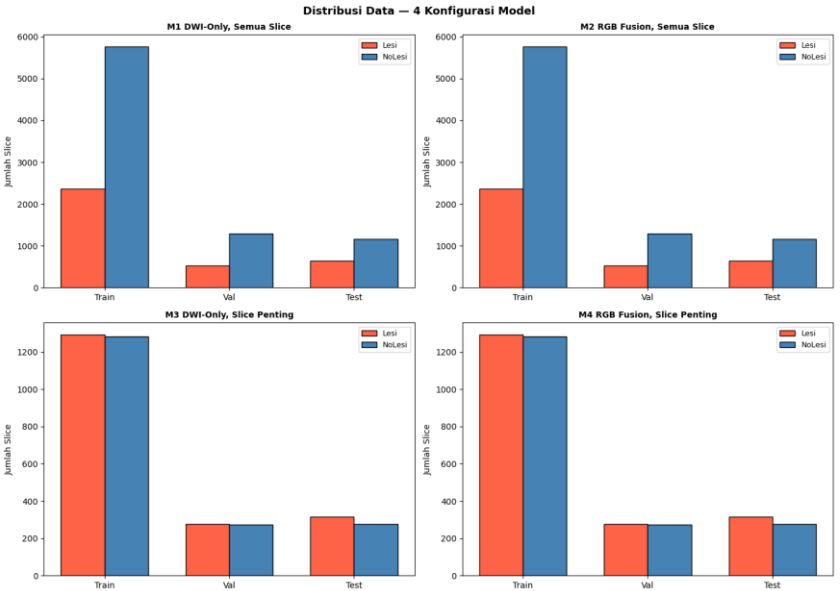
Jenis Data	**Jumlah Pasien**	**Proporsi**
Training	161	70%
Validation	34	15%
Testing	35	15%
Total	**230**	**100%**

Jumlah irisan yang dihasilkan pada masing-masing konfigurasi berbeda secara mencolok sebagaimana disajikan pada Tabel 5.6.

Tabel 5.6 Distribusi Jumlah Irisan pada Empat Konfigurasi

Model	***Training***	***Validation***	***Testing***	**Distribusi Kelas *Training***
M1 DWI-All	8.126	1.805	1.799	Lesi 2.364 : Non-lesi 5.762
M2 RGB-All	8.126	1.805	1.799	Lesi 2.364 : Non-lesi 5.762
M3 DWI-Important	2.575	549	593	Lesi 1.292 : Non-lesi 1.283
M4 RGB-Important	2.575	549	593	Lesi 1.292 : Non-lesi 1.283

Perbedaan ini bersifat mendasar dan bukan sekadar perbedaan jumlah. Konfigurasi *All Slices* memuat lebih dari tiga kali lipat data dengan perbandingan kelas sekitar 1 berbanding 2,4 yang condong ke arah non-lesi. Konfigurasi *Important Slices* memuat data jauh lebih sedikit namun dengan perbandingan kelas yang nyaris seimbang sempurna. Visualisasi distribusinya ditunjukkan pada Gambar 5.9.



Gambar 5.9 Distribusi Jumlah Irisan pada Empat Konfigurasi Dataset

Angka pada kolom pengujian merupakan bagian yang paling menentukan bagi pembacaan hasil pada subbab-subbab berikutnya. Model M1 dan M2 diuji pada 1.799 irisan, sedangkan M3 dan M4 diuji pada 593 irisan. Kedua himpunan tersebut tidak hanya berbeda ukuran, tetapi juga berbeda komposisi dan tingkat kesulitannya, sebagaimana akan diuraikan secara kuantitatif pada Subbab 5.4.5.

Perbandingan Empat Konfigurasi Model

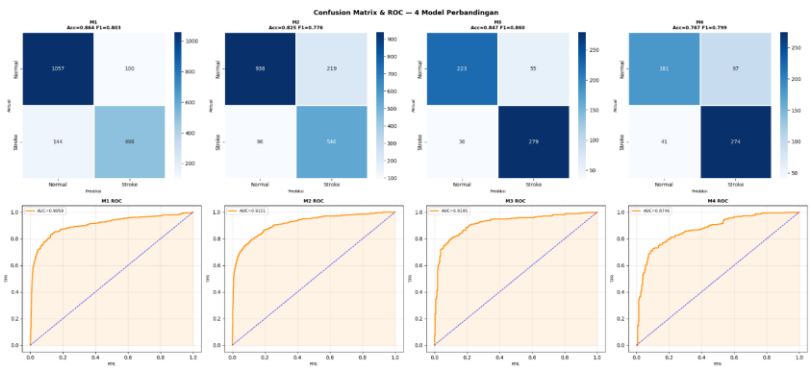
Hasil evaluasi keempat konfigurasi model disajikan pada Tabel 5.7.

Tabel 5.7 Hasil Evaluasi Empat Konfigurasi *DenseNet121-CBAM*

Model	**Masukan**	**Strategi Irisan**	***Accuracy***	**F1-Score**	**AUC-ROC**
M1	DWI-*Only*	*All Slices*	0,8644	0,8032	0,9059
M2	*RGB Fusion*	*All Slices*	0,8249	0,7761	0,9151
M3	**DWI-*Only***	***Important Slices***	**0,8466**	**0,8598**	**0,9195**
M4	*RGB Fusion*	*Important Slices*	0,7673	0,7988	0,8746

Terdapat dua pola yang segera terlihat. Pola pertama, konfigurasi berbasis *DWI-Only* secara konsisten mengungguli *RGB Fusion* pada strategi seleksi irisan mana pun. Pola kedua, konfigurasi dengan *Important Slices* menghasilkan *F1-score* yang lebih tinggi meskipun *accuracy*-nya justru lebih rendah. Kedua pola tersebut akan dianalisis secara terpisah karena memiliki penjelasan yang berbeda.

Model M3 dengan *DWI-Only* dan *Important Slices* memperoleh *F1-score* tertinggi sebesar 0,8598 dan AUC tertinggi sebesar 0,9195, sehingga dipilih sebagai konfigurasi terbaik. Model M1 sesungguhnya memperoleh *accuracy* tertinggi sebesar 0,8644, namun selisih *F1-score* yang mencapai 0,057 dinilai lebih menentukan karena mencerminkan keseimbangan yang lebih baik antara kemampuan mendeteksi lesi dan menghindari alarm palsu. Visualisasi *confusion matrix* dan kurva ROC keempat model ditunjukkan pada Gambar 5.10.



Gambar 5.10 *Confusion Matrix* dan Kurva ROC Empat Konfigurasi Model

Rincian angka pada *confusion matrix* memungkinkan perhitungan metrik yang jauh lebih informatif daripada yang tercantum pada Tabel 5.7, dan hasilnya disajikan pada Tabel 5.8.

Tabel 5.8 Rincian *Confusion Matrix* dan Metrik Turunannya

Model	TN	FP	FN	TP	*Precision*	*Recall*	Specificity
M1	1.057	100	144	498	0,833	0,776	0,914
M2	938	219	96	546	0,714	0,851	0,811
M3	223	55	36	279	0,835	0,886	0,802
M4	181	97	41	274	0,739	0,870	0,651

Tabel tersebut mengungkap sesuatu yang tersembunyi di balik angka rerata. Pada model M1 yang memperoleh *accuracy* tertinggi, sebanyak 144 dari 642 irisan berlesi keliru dinyatakan tidak berlesi, setara *recall* sebesar 0,776. Pada rancangan dua tahap yang telah dibahas, keseratus empat puluh empat irisan tersebut tidak akan pernah mencapai tahap segmentasi. Angka *accuracy* sebesar 0,8644 yang tampak paling baik pada Tabel 5.7 karena itu justru menyembunyikan kelemahan yang paling berbahaya bagi sistem ini.

Sebaliknya, model M2 memperoleh *recall* sebesar 0,851 yang lebih tinggi daripada M1, meskipun *accuracy*-nya lebih rendah. Perbedaannya terletak pada perilaku: M2 jauh lebih longgar dalam menyatakan positif, dengan 219 alarm palsu dibandingkan 100 pada M1. Pada konteks penapisan bertahap, perilaku semacam ini sesungguhnya lebih dikehendaki, sebab alarm palsu masih dapat disaring oleh tahap segmentasi yang menghasilkan masker kosong, sedangkan lesi yang terlewat tidak dapat dipulihkan.

Mengapa *RGB Fusion* Justru Berperforma Lebih Rendah

Hasil bahwa penggabungan tiga modalitas berperforma lebih rendah daripada penggunaan satu modalitas saja bertentangan dengan intuisi dan karena itu layak dianalisis secara khusus. Tiga penjelasan dapat diajukan, dan ketiganya kemungkinan berperan bersama-sama.

Penjelasan pertama berkaitan dengan korelasi antarkanal. Sebagaimana dibahas pada Subbab Representasi Masukan: DWI-*Only* dan RGB *Fusion*, bobot *ImageNet* dipelajari dari citra alami yang ketiga kanal warnanya berkorelasi kuat. Filter pada lapisan awal *DenseNet* dibentuk berdasarkan keteraturan tersebut. Pada citra *RGB Fusion*, kanal merah dan hijau justru berlawanan pada area lesi, sehingga filter yang mengharapkan keselarasan antarkanal menghasilkan respons yang lemah tepat di lokasi yang paling penting. Model harus mempelajari kembali perilaku lapisan awal,

padahal lapisan-lapisan tersebut justru dibekukan selama tahap awal pelatihan.

Penjelasan kedua berkaitan dengan pengenceran sinyal. DWI merupakan sekuens yang paling sensitif terhadap infark akut. Ketika ditempatkan sebagai salah satu dari tiga kanal, kontribusinya terhadap keputusan model bersaing dengan dua kanal lain yang sebagian informasinya tidak relevan atau bahkan kosong. Pada pasien hiperakut, kanal FLAIR sengaja tidak menunjukkan apa pun di lokasi lesi, sehingga sepertiga masukan model tidak membawa informasi diagnostik apa pun untuk kasus tersebut.

Penjelasan ketiga berkaitan dengan penyelarasan antarmodalitas. Ketiga sekuens diperoleh melalui akuisisi yang terpisah, sehingga pergerakan kepala pasien di antara akuisisi menimbulkan pergeseran spasial. Apabila DWI dan ADC bergeser beberapa piksel satu sama lain, maka pada citra gabungan tepi lesi memuat wilayah dengan kombinasi kanal yang keliru. Bab ini menerapkan penyelarasan orientasi kanonik dan interpolasi jumlah irisan, namun tidak menerapkan registrasi antarmodalitas yang sesungguhnya. Untuk citra keabuan tunggal ketidaksejajaran semacam ini tidak berdampak, tetapi untuk citra gabungan tiga kanal dampaknya langsung terasa.

Pelajaran: Menambah Modalitas Tidak Sama dengan Menambah Informasi

Terdapat kecenderungan untuk beranggapan bahwa memberikan lebih banyak data kepada model pasti menghasilkan performa yang lebih baik. Hasil pada bab ini menunjukkan sebaliknya, dan alasannya bersifat mendasar.

Penggabungan modalitas memberi manfaat apabila ketiga syarat berikut terpenuhi: modalitas tambahan memuat informasi yang tidak terkandung pada modalitas utama, informasi tersebut tersaji dalam bentuk yang dapat dimanfaatkan arsitektur yang digunakan, dan seluruh modalitas telah tersejajarkan secara spasial. Apabila salah satu syarat tidak terpenuhi, modalitas tambahan berubah menjadi derau yang mengencerkan sinyal utama.

Pendekatan yang lebih menjanjikan bukanlah menumpuk modalitas ke dalam kanal warna, melainkan menggunakan jalur pengekstraksi fitur terpisah untuk setiap modalitas lalu menggabungkan representasinya pada tingkat fitur, bukan pada tingkat piksel. Dengan cara itu, setiap modalitas diproses menggunakan filter yang sesuai dengan karakteristiknya sendiri.

Analisis Kritis terhadap Perbandingan Antarkonfigurasi

Bagian ini memeriksa apakah perbandingan pada Tabel 5.7 sesungguhnya sahih. Pemeriksaan dimulai dari komposisi himpunan uji masing-masing konfigurasi sebagaimana dirangkum pada Tabel 5.9.

Tabel 5.9 Komposisi Himpunan Uji pada Kedua Strategi Seleksi

Aspek	**M1 dan M2 (*All Slices*)**	**M3 dan M4 (*Important Slices*)**
Jumlah irisan uji	1.799	593
Irisan berlesi	642 (35,7%)	315 (53,1%)
Irisan non-lesi	1.157 (64,3%)	278 (46,9%)
Karakteristik kelas positif	Seluruh irisan berlesi	Hanya irisan dengan lesi terbesar
Karakteristik kelas negatif	Seluruh irisan tanpa lesi	Hanya irisan tengah volume
Baseline kelas mayoritas	64,3%	53,1%

Tabel tersebut memperlihatkan bahwa kedua himpunan uji berbeda dalam hampir setiap aspek. Perbedaan yang paling menentukan terletak pada karakteristik kelas positifnya. Himpunan uji M1 dan M2 memuat seluruh irisan berlesi, termasuk irisan tepi yang hanya menyerempet ujung lesi dan karena itu memuat area infark yang sangat kecil. Himpunan uji M3 dan M4 hanya memuat irisan dengan rasio lesi terbesar, yaitu irisan tempat lesi tampak paling jelas dan paling mudah dikenali.

Konsekuensinya, *recall* M3 sebesar 0,886 yang lebih tinggi daripada *recall* M1 sebesar 0,776 tidak dapat langsung dibaca sebagai bukti bahwa M3 lebih baik dalam menemukan lesi. Sebagian

besar selisih tersebut kemungkinan mencerminkan kenyataan bahwa M3 hanya diminta menemukan lesi yang besar, sedangkan M1 diminta menemukan lesi berukuran apa pun. Membandingkan keduanya menyerupai membandingkan hasil dua ujian dengan tingkat kesulitan yang berbeda tanpa menyesuaikan skornya.

Perbedaan *baseline* kelas mayoritas juga memengaruhi pembacaan *accuracy*. Pada himpunan uji M1, sebuah model yang selalu menebak non-lesi memperoleh akurasi 64,3%; pada himpunan uji M3, model yang sama memperoleh 53,1%. Dengan demikian, *accuracy* M1 sebesar 0,8644 berada 22,1 poin di atas *baseline*-nya, sedangkan *accuracy* M3 sebesar 0,8466 berada 31,5 poin di atas *baseline*-nya. Dibaca dengan cara ini, keunggulan M3 justru menjadi lebih besar daripada yang tampak pada Tabel 5.7, namun angka ini pun masih belum memperhitungkan perbedaan kesulitan kelas positifnya.

Perhatian: Cara Membandingkan yang Benar

Rancangan yang sahih untuk menjawab pertanyaan Metodologi ini sesungguhnya sederhana. Strategi seleksi irisan diterapkan *hanya* pada data pelatihan, sedangkan keempat model dievaluasi pada satu himpunan uji yang sama yang memuat seluruh irisan sesuai distribusi nyata.

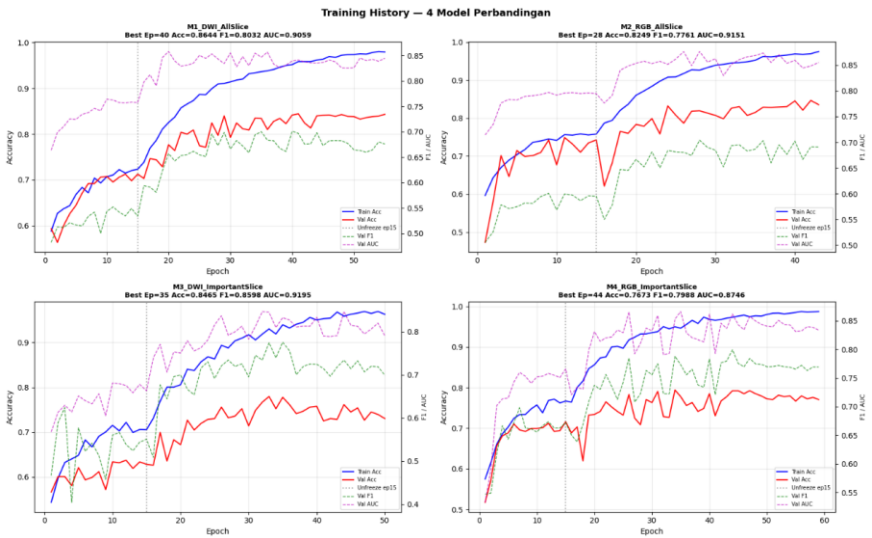
Dengan rancangan tersebut, pertanyaan yang dijawab menjadi tegas: apakah melatih pada irisan terpilih menghasilkan model yang lebih baik ketika menghadapi kondisi sebenarnya. Pertanyaan itulah yang relevan bagi penerapan, sebab pada praktik nyata sistem akan menerima seluruh irisan dari sebuah pemindaian dan tidak memiliki cara untuk mengetahui sebelumnya irisan mana yang memuat lesi terbesar.

Perlu ditambahkan bahwa terdapat kemungkinan hasilnya berbalik. Model yang hanya pernah melihat lesi berukuran besar selama pelatihan berpeluang gagal mengenali lesi kecil, sehingga pada himpunan uji yang realistis performanya justru dapat lebih rendah daripada model yang dilatih pada seluruh irisan. Kemungkinan ini tidak dapat dikesampingkan tanpa pengujian, dan pengujian itulah yang belum dilakukan.

Satu catatan teknis perlu ditambahkan. Nilai *F1-score* M3 tercantum sebagai 0,8595 pada tabel hasil laporan asli, sedangkan perhitungan langsung dari *confusion matrix* pada Gambar 5.10 menghasilkan 0,8598, yang juga merupakan angka yang tercantum pada narasi dan pada dasbor sistem. Selisih sebesar 0,0003 ini tidak berarti secara substantif, tetapi konsistensi pelaporan angka merupakan hal yang perlu dijaga karena pembaca tidak memiliki cara untuk mengetahui angka mana yang benar tanpa menghitung ulang sendiri.

Riwayat Pelatihan dan Evaluasi Akhir dengan TTA

Perkembangan performa keempat model selama pelatihan ditunjukkan pada Gambar 5.11.



Gambar 5.11 Riwayat Pelatihan Empat Konfigurasi Model

Seluruh model menunjukkan peningkatan performa seiring bertambahnya *epoch*, dengan kurva pelatihan dan validasi yang bergerak searah. Model M3 memiliki kurva validasi yang paling stabil dengan peningkatan *F1-score* dan AUC yang konsisten, serta selisih antara kurva pelatihan dan validasi yang relatif kecil sehingga menandakan kemampuan generalisasi yang baik. Penerapan augmentasi, *dropout*, *mixup*, dan *early stopping* tampak berhasil menekan *overfitting*.

Model M3 selanjutnya dievaluasi kembali menggunakan *Test-Time Augmentation* sesuai Persamaan 5.17. Perbandingan hasil sebelum dan sesudah penerapan TTA disajikan pada Tabel 5.10.

Tabel 5.10 Perbandingan Hasil Model M3 Sebelum dan Sesudah TTA

Metrik	**M3 tanpa TTA**	**M3 dengan TTA**	**Selisih**
Accuracy	0,8466	0,8555	+0,0089
F1-Score	0,8598	0,8006	−0,0592
AUC-ROC	0,9195	0,9158	−0,0037

Perhatian: TTA Menaikkan Akurasi namun Menurunkan F1 secara Berarti

Angka pada Tabel 5.10 memerlukan penjelasan karena bertentangan dengan alasan yang lazim dikemukakan untuk menerapkan TTA. Akurasi memang naik sebesar 0,89 poin persentase, tetapi *F1-score* turun sebesar 5,92 poin persentase dan AUC turun tipis. Penurunan *F1* sebesar itu jauh lebih besar daripada kenaikan akurasinya.

Penjelasan yang paling mungkin adalah pergeseran ambang efektif. Perata-rataan atas beberapa prediksi menarik nilai probabilitas menuju bagian tengah rentang. Prediksi positif yang semula berada sedikit di atas ambang 0,5 dapat bergeser ke bawahnya, sehingga sebagian *true positive* berubah menjadi *false negative*. Akibatnya *recall* menurun, dan karena *F1* sangat peka terhadap *recall*, nilainya ikut turun tajam. Akurasi tetap naik karena penurunan pada kelas positif diimbangi perbaikan pada kelas negatif yang jumlahnya lebih besar.

Apabila dugaan ini benar, penyelesaiannya sederhana dan tidak memerlukan perubahan model: nilai ambang perlu ditala ulang pada data validasi *setelah* TTA diterapkan, bukan sebelum. Hal ini merupakan penerapan langsung dari prinsip pencarian ambang optimal yang telah dibahas pada Bab 2.

Yang perlu dihindari adalah menyatakan bahwa penerapan TTA menghasilkan performa yang paling stabil tanpa menyebutkan bahwa metrik utamanya justru menurun. Ketika sebuah teknik

memperbaiki satu metrik namun memperburuk metrik lain, keduanya harus dilaporkan bersama-sama, dan pemilihan konfigurasi akhir harus disertai alasan yang jelas mengenai metrik mana yang diprioritaskan dan mengapa.

Pada konteks sistem ini, alasan tersebut sesungguhnya sudah jelas. Karena rancangan dua tahap membuat *recall* klasifikasi menjadi batas atas kemampuan sistem menemukan lesi, konfigurasi yang menurunkan *recall* justru berlawanan dengan kebutuhan. Konfigurasi tanpa TTA dengan *F1-score* 0,8598 karena itu lebih sesuai, kecuali apabila TTA disertai penalaan ulang ambang keputusan.

Hasil Segmentasi Lesi

Evaluasi segmentasi dilakukan menggunakan 50 data pengujian dari dataset ISLES 2022 dengan hasil sebagaimana disajikan pada Tabel 5.11.

Tabel 5.11 Hasil Evaluasi Segmentasi *NVAutoNet*

Metrik	**Nilai**
Mean Dice Score	0,8347
Mean Precision	0,8699
Mean Recall	0,8221
Standar deviasi Dice	0,1575
Kasus dengan Dice ≥ 0,7	46 dari 50
Kasus dengan Dice ≥ 0,5	47 dari 50
Kasus dengan Dice ≤ 0,5	3 dari 50

Nilai rerata *Dice Score* sebesar 0,8347 menunjukkan tingkat kesesuaian yang tinggi antara masker prediksi dan *ground truth*. Nilai *precision* sebesar 0,8699 yang lebih tinggi daripada *recall* sebesar 0,8221 menandakan bahwa model cenderung konservatif: area yang ditandai sebagai lesi umumnya memang lesi, namun sebagian tepi lesi tidak tertandai.

Kolom standar deviasi sebesar 0,1575 justru merupakan angka yang paling penting pada tabel tersebut, dan paling mudah terlewat. Nilai sebesar ini menandakan sebaran yang sangat lebar. Apabila distribusinya mendekati normal, maka sekitar sepertiga kasus memiliki *Dice* di luar rentang 0,68 sampai 0,99. Dengan kata lain, rerata sebesar 0,8347 tidak menggambarkan performa yang dialami kebanyakan pasien secara seragam, melainkan merupakan rerata dari mayoritas kasus yang sangat baik dan sejumlah kecil kasus yang gagal total.

Beberapa kasus ekstrem disajikan pada Tabel 5.12.

Tabel 5.12 Kasus dengan Performa Segmentasi Ekstrem

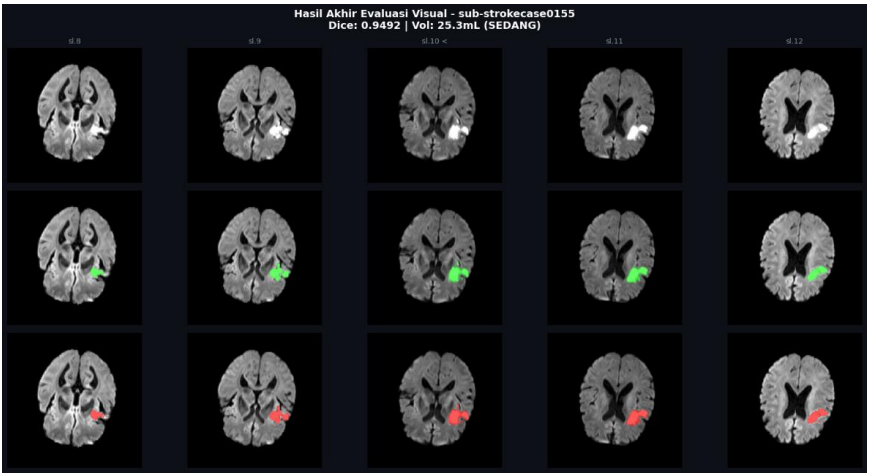
Kasus	***Dice***	***Precision***	***Recall***	**Volume Lesi**
sub-strokecase0155	0,9492	0,9553	0,9432	25,31 mL
sub-strokecase0063	0,9446	0,9137	0,9776	6,49 mL
sub-strokecase0151	0,0000	—	—	sangat kecil

Kasus sub-strokecase0151 dengan *Dice* sebesar nol layak diperhatikan secara khusus. Nilai nol pada metrik ini bersifat mutlak dan hanya terjadi ketika tidak ada satu pun *voxel* yang beririsan antara prediksi dan *ground truth*. Terdapat dua kemungkinan penyebabnya: model tidak menemukan lesi sama sekali sehingga masker prediksinya kosong, atau model menemukan sesuatu namun pada lokasi yang sepenuhnya keliru. Keduanya memiliki implikasi yang sangat berbeda, dan membedakannya menuntut pemeriksaan visual terhadap kasus tersebut, pemeriksaan yang belum dilakukan.

Perlu pula ditegaskan bahwa pada kasus dengan lesi sangat kecil, nilai *Dice* memang cenderung rendah bahkan ketika segmentasinya secara klinis dapat diterima. Sebuah lesi berukuran 20 *voxel* yang diprediksi dengan pergeseran beberapa *voxel* dapat menghasilkan *Dice* mendekati nol, padahal lokasi yang ditunjukkan sesungguhnya benar. Hal ini merupakan keterbatasan metrik, bukan semata kegagalan model, dan menjadi alasan mengapa evaluasi segmentasi pada lesi kecil sebaiknya dilengkapi metrik berbasis

jarak seperti *Hausdorff distance* atau sekadar tingkat keberhasilan deteksi.

Visualisasi hasil segmentasi pada kasus dengan performa terbaik ditunjukkan pada Gambar 5.12.



Gambar 5.12 Visualisasi Hasil Segmentasi pada Kasus sub-strokecase0155

Gambar tersebut menampilkan tiga baris. Baris pertama memuat citra DWI asli pada beberapa irisan terpilih, dengan lesi tampak sebagai area hiperintens pada satu hemisfer. Baris kedua menampilkan *ground truth* hasil anotasi dokter yang ditandai warna hijau, sedangkan baris ketiga menampilkan hasil prediksi *NVAutoNet* yang ditandai warna merah. Model mampu mengikuti bentuk dan lokasi lesi secara konsisten pada beberapa irisan, dengan kesesuaian tertinggi pada irisan yang memuat area lesi paling dominan.

Yang patut diapresiasi adalah kontinuitas segmentasi antariris. Bentuk lesi berubah secara halus dari satu irisan ke irisan berikutnya, mengikuti bentuk tiga dimensi lesi yang sesungguhnya. Kontinuitas semacam ini merupakan keunggulan model tiga dimensi yang tidak dapat dicapai oleh segmentasi dua dimensi per irisan, dan sekaligus menegaskan kembali pentingnya menjaga keutuhan volume sebagaimana dibahas pada Subbab 5.3.8. Perbedaan yang tersisa umumnya berada pada tepi lesi, tempat intensitas jaringan menyerupai jaringan normal di sekitarnya.

Kegagalan yang Bukan Kegagalan Model: Kasus *Space Mismatch*

Nilai *Dice Score* sebesar 0,8347 yang dilaporkan pada subbab sebelumnya tidak diperoleh pada percobaan pertama. Pada evaluasi awal, nilai yang dihasilkan hanya sebesar 0,0866. Angka tersebut praktis menyatakan bahwa masker prediksi dan *ground truth* nyaris tidak beririsan sama sekali, yaitu tingkat kegagalan yang setara dengan menebak secara acak.

Situasi semacam ini menempatkan peneliti pada persimpangan yang menentukan. Tafsiran yang paling mudah adalah menyimpulkan bahwa model tidak sesuai untuk data ini, lalu beralih mencari model lain atau melakukan pelatihan ulang. Tafsiran itu keliru, dan kekeliruannya akan menghabiskan waktu berminggu-minggu tanpa hasil.

Petunjuk yang membedakan keduanya terletak pada besarnya penyimpangan. *NVAutoNet* merupakan komponen dari kerangka kerja yang memenangi kompetisi ISLES 2022 dan telah dilaporkan mencapai *Dice* di atas 0,8 pada dataset yang sama persis dengan yang digunakan di sini. Sebuah model yang telah terbukti pada data tertentu tidak mungkin turun menjadi 0,0866 pada data yang sama karena alasan yang berkaitan dengan kemampuan model. Selisih sebesar itu bukan penurunan performa, melainkan pertanda bahwa ada sesuatu yang salah pada proses perbandingannya.

Penelusuran mengarah pada ketidakselarasan ruang fisik sebagaimana telah diuraikan pada Subbab 5.2.13. Berkas prediksi dan berkas *ground truth* memiliki matriks afin yang berbeda, sehingga ketika keduanya dibandingkan pada tingkat indeks larik, yang diperbandingkan sesungguhnya adalah dua objek yang berada pada posisi ruang berlainan. Penyelesaiannya berupa penyelarasan ruang fisik menggunakan pustaka SimpleITK, yaitu memetakan salah satu berkas ke dalam sistem koordinat berkas lainnya sebelum metrik dihitung. Dampaknya disajikan pada Tabel 5.13.

Tabel 5.13 Dampak Penyelarasan Ruang Fisik terhadap Nilai *Dice*

Kondisi Evaluasi	***Mean Dice Score***	**Tafsiran**
Sebelum penyelarasan	0,0866	Tampak seperti kegagalan total model
Setelah penyelarasan	0,8347	Performa sesuai laporan kompetisi ISLES

Pelajaran Utama: Kesalahan yang Tidak Menimbulkan Pesan Galat

Kasus ini merupakan temuan paling berharga pada bab ini, dan barangkali pada keseluruhan buku ini, meskipun tidak berkaitan dengan arsitektur maupun akurasi.

Perhatikan sifat kesalahannya. Program berjalan sampai selesai. Tidak ada pengecualian yang dilemparkan. Tidak ada peringatan yang muncul. Ukuran larik keduanya bersesuaian sehingga operasi perhitungan berlangsung tanpa hambatan. Satu-satunya gejala adalah sebuah angka yang buruk dan angka yang buruk selalu memiliki penjelasan yang tampak masuk akal berupa model yang kurang baik.

Kelas kesalahan semacam ini khas pada pekerjaan citra medis dan jarang dijumpai pada dataset citra umum. Berkas JPEG hanya menyimpan larik piksel; berkas *NIfTI* dan DICOM menyimpan larik piksel *beserta* informasi posisinya di ruang nyata. Mengabaikan informasi kedua tersebut menghasilkan program yang berjalan tanpa keluhan namun membandingkan hal yang salah.

Kebiasaan yang perlu dibangun: ketika sebuah metrik jauh menyimpang dari yang dilaporkan pada literatur untuk model dan data yang sama, curigai alur pemrosesan terlebih dahulu sebelum menyalahkan model. Penurunan sebesar sepuluh atau dua puluh persen dapat dijelaskan oleh perbedaan data. Penurunan menjadi sepersepuluh nilai yang diharapkan hampir selalu menandakan kesalahan teknis.

Pemeriksaan yang murah dan cepat: cetak matriks afin kedua berkas dan bandingkan; tampilkan satu irisan prediksi dan satu irisan *ground truth* secara berdampingan; periksa apakah keduanya memiliki ukuran *voxel* dan titik asal yang sama. Tiga langkah tersebut memerlukan waktu beberapa menit dan dapat menghemat berminggu-minggu pekerjaan yang salah arah.

Terdapat satu implikasi lanjutan yang perlu dinyatakan. Karena penyelarasan ruang fisik memengaruhi seluruh perhitungan berbasis *voxel*, maka estimasi volume lesi yang dibahas pada subbab berikutnya juga bergantung pada penyelarasan yang benar. Volume

yang dihitung dari masker yang tidak tersejajarkan akan keliru bukan hanya pada lokasinya, tetapi juga pada besarannya apabila ukuran *voxel* kedua berkas berbeda.

Estimasi Volume dan Penentuan Tingkat Keparahan

Volume lesi dihitung dari masker segmentasi menggunakan Persamaan 5.20 dan 5.21, dengan ukuran *voxel* dibaca langsung dari metadata citra. Pada kasus sub-strokecase0155, volume yang diperoleh sebesar 25,31 mL, sedangkan pada kasus sub-strokecase0063 sebesar 6,49 mL. Kedua angka tersebut selanjutnya digunakan sebagai indikator kuantitatif tingkat keparahan infark.

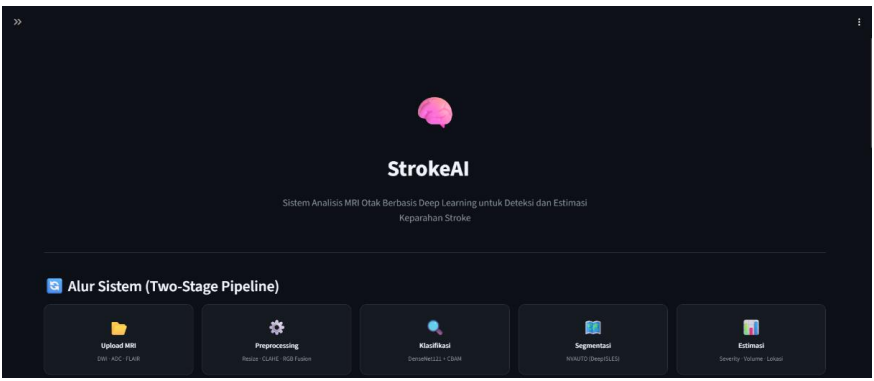
Keandalan angka volume bergantung pada kualitas segmentasi yang mendasarinya, namun hubungan keduanya tidak sesederhana yang tampak. Sebagaimana telah dibahas pada Bab 2, kesalahan pada dua arah dapat saling meniadakan sehingga volume total tetap mendekati benar meskipun *Dice*-nya rendah; sebaliknya, prediksi yang secara sistematis sedikit lebih sempit dari lesi sebenarnya menghasilkan taksiran volume yang bias ke bawah meskipun *Dice*-nya tinggi. Pada kasus ini, nilai *precision* rerata sebesar 0,8699 yang lebih tinggi daripada *recall* rerata sebesar 0,8221 justru menandakan kecenderungan yang kedua: model cenderung menandai area yang lebih sempit daripada lesi sebenarnya, sehingga volume yang dilaporkan berpeluang lebih kecil daripada volume yang sesungguhnya.

> **Perhatian: Volume Belum Cukup untuk Menyatakan Tingkat Keparahan**
> Volume infark memang merupakan salah satu penentu prognosis pada stroke iskemik, tetapi ia bukan satu-satunya dan bahkan bukan yang terpenting pada semua kasus.
> Lokasi lesi sering kali lebih menentukan daripada ukurannya. Infark berukuran kecil pada batang otak dapat mengancam nyawa karena mengenai pusat pengaturan pernapasan dan kesadaran, sedangkan infark yang jauh lebih besar pada lobus frontal nondominan dapat menimbulkan gejala yang relatif ringan. Sebuah sistem yang menyatakan keparahan semata-mata berdasarkan volume akan menilai kedua kasus tersebut secara terbalik.

Faktor kedua adalah waktu. Sebagaimana dibahas pada Subbab 5.2.1, volume infark merupakan potret sesaat dari proses yang masih berlangsung. Volume 20 mL pada dua jam setelah awitan memiliki makna yang sangat berbeda dari volume 20 mL pada dua puluh empat jam setelah awitan.
Faktor ketiga adalah keberadaan penumbra, yaitu jaringan yang masih dapat diselamatkan dan justru menjadi sasaran terapi. Informasi ini tidak terkandung pada volume infark dan memerlukan pencitraan perfusi.
Praktik yang lebih bertanggung jawab adalah melaporkan volume beserta lokasi anatomisnya dan waktu pemindaian sejak awitan, lalu menyerahkan penafsiran tingkat keparahan kepada klinisi. Apabila kategori keparahan tetap hendak ditampilkan, ambangnya harus berasal dari literatur klinis yang dirujuk secara eksplisit, bukan ditetapkan sendiri oleh perancang sistem.

Integrasi Sistem dalam Dasbor Interaktif

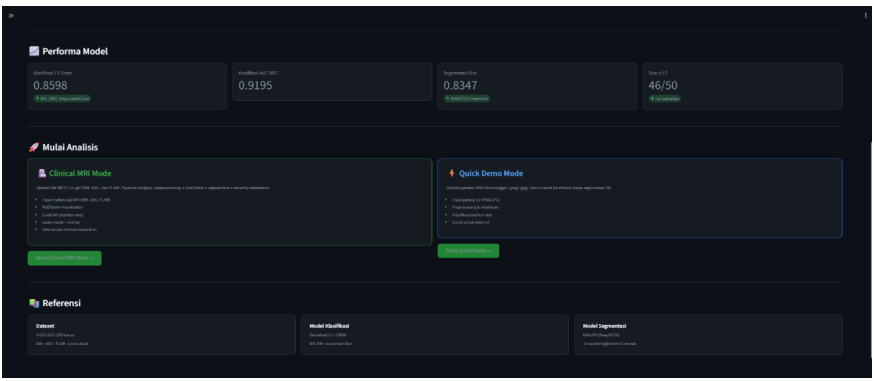
Seluruh alur kerja diintegrasikan ke dalam antarmuka berbasis *Streamlit* bernama StrokeAI. Tampilan halaman utamanya ditunjukkan pada Gambar 5.13.



Gambar 5.13 Tampilan Beranda Aplikasi StrokeAI dan Alur Kerja Sistem

Halaman beranda menampilkan identitas aplikasi beserta visualisasi alur dua tahap yang memetakan lima tahapan pemrosesan secara berurutan, yaitu pengunggahan citra MRI, prapemrosesan,

klasifikasi, segmentasi, dan estimasi. Penyajian alur secara eksplisit merupakan keputusan rancangan yang baik karena memberi pengguna gambaran mengenai apa yang sedang dilakukan sistem terhadap datanya. Panel performa model ditunjukkan pada Gambar 5.14.



Gambar 5.14 Panel Metrik Performa Model dan Pemilihan Mode Analisis

Panel tersebut menampilkan *F1-score* klasifikasi sebesar 0,8598, AUC-ROC sebesar 0,9195, *Mean Dice Score* segmentasi sebesar 0,8347, serta keterangan bahwa 46 dari 50 sampel validasi melampaui ambang *Dice* 0,7. Menampilkan metrik performa secara terbuka kepada pengguna merupakan praktik yang patut ditiru, sebab klinisi berhak mengetahui tingkat keandalan alat yang digunakannya. Perlu dicatat bahwa angka yang ditampilkan adalah 0,8598, yaitu hasil tanpa TTA, sedangkan kesimpulan laporan menyebut 0,8006 sebagai hasil akhir dengan TTA. Ketidakselarasan ini perlu dirapikan agar pengguna memperoleh informasi yang konsisten.

Sistem menyediakan dua mode analisis. *Clinical MRI Mode* menerima berkas volumetrik *NIfTI* dari pemindai rumah sakit dan menjalankan keseluruhan alur termasuk segmentasi tiga dimensi. *Quick Demo Mode* menerima satu irisan dua dimensi dalam format gambar biasa dan hanya menjalankan tahap klasifikasi beserta penampilan peta perhatian, tanpa mengaktifkan mesin segmentasi yang berat. Pemisahan ini merupakan rancangan yang tepat karena membedakan penggunaan sesungguhnya dari peragaan antarmuka.

Catatan: Satu Angka yang Sebaiknya Turut Ditampilkan
Panel performa menampilkan *F1-score*, AUC, dan *Dice*. Terdapat satu angka lagi yang sesungguhnya paling relevan bagi pengguna klinis namun belum ditampilkan, yaitu *recall* tahap klasifikasi.
Alasannya berkaitan langsung dengan sifat rancangan dua tahap. Nilai *recall* menyatakan berapa persen irisan berlesi yang berhasil diteruskan ke tahap segmentasi, dan karena tidak ada jalur pemeriksaan ulang, angka itulah yang menentukan berapa persen lesi yang berpeluang luput dari sistem secara keseluruhan.
Bagi seorang dokter yang hendak memutuskan seberapa jauh ia dapat bersandar pada keluaran sistem, pernyataan bahwa sistem meneruskan sekitar 89% irisan berlesi jauh lebih berguna daripada nilai AUC sebesar 0,9195. Yang pertama menjawab pertanyaan mengenai apa yang mungkin terlewat; yang kedua menjawab pertanyaan mengenai kemampuan pemisahan model pada seluruh nilai ambang, yang bukan merupakan pertanyaan klinis.

Pembahasan dan Keterbatasan

Hasil pada bab ini perlu dibaca bersama sejumlah keterbatasan berikut, yang dikelompokkan menurut tahapan sistem.
Keterbatasan pada rancangan eksperimen:

a. Strategi seleksi irisan diterapkan pada data pengujian sehingga keempat konfigurasi tidak dievaluasi pada permasalahan yang setara, dan perbandingan angkanya sebagian mencerminkan perbedaan tingkat kesulitan himpunan uji.
b. Terdapat ketidakselarasan pelaporan jumlah pasien antara bagian metodologi yang menyebut 175, 37, dan 38 pasien dengan bagian hasil yang menyebut 161, 34, dan 35 pasien. Selisih 20 pasien ini perlu diklarifikasi karena memengaruhi seluruh angka yang diturunkan darinya.
c. Bagian kesimpulan menyebut pembagian data 80 berbanding 20 secara acak, sedangkan bagian metodologi menyebut 70, 15, dan 15 secara *patient-wise*. Perbedaan ini bukan sekadar penulisan, sebab pembagian acak pada tingkat irisan akan

menimbulkan kebocoran data sedangkan pembagian *patient-wise* tidak.

d. Penerapan TTA menurunkan *F1-score* sebesar 5,92 poin persentase namun dilaporkan sebagai konfigurasi yang paling stabil, tanpa penalaan ulang ambang keputusan setelah TTA diterapkan.

Keterbatasan pada tahap klasifikasi:

a. Nilai *recall* pada konfigurasi terbaik berkisar 0,886 pada himpunan uji terpilih, yang berarti sekitar satu dari sembilan irisan berlesi tidak diteruskan ke tahap segmentasi dan lesinya tidak akan pernah terhitung.
b. Ambang keputusan dipertahankan pada nilai bawaan 0,5 tanpa penalaan, padahal sifat rancangan dua tahap menuntut ambang yang lebih rendah agar mengutamakan *recall*.
c. Modul CBAM ditempatkan hanya pada keluaran akhir dengan resolusi 7 × 7, sehingga setiap sel peta perhatian mewakili wilayah 32 × 32 piksel dan berpotensi terlalu kasar bagi lesi berukuran kecil.
d. Peta perhatian CBAM tidak diverifikasi terhadap lokasi lesi sebenarnya, padahal verifikasi semacam ini merupakan salah satu manfaat utama penggunaan mekanisme perhatian.

Keterbatasan pada tahap segmentasi dan kuantifikasi:

a. Standar deviasi *Dice* sebesar 0,1575 menunjukkan sebaran yang lebar, sehingga rerata sebesar 0,8347 tidak mencerminkan pengalaman yang seragam bagi seluruh pasien.
b. Terdapat kasus dengan *Dice* sebesar nol yang belum diperiksa secara visual untuk membedakan antara masker kosong dan masker pada lokasi keliru.
c. Penyusunan ulang irisan positif menjadi volume berpotensi menghasilkan susunan yang tidak kontinu, padahal *NVAutoNet* merupakan model tiga dimensi yang mengandalkan konteks antariris.
d. Penentuan tingkat keparahan hanya bertumpu pada volume tanpa memperhitungkan lokasi anatomis maupun waktu sejak awitan gejala.

Keterbatasan pada validasi:

a. Validasi eksternal menggunakan data RSUD Haji telah direncanakan dan memperoleh persetujuan etik, namun hasilnya belum dilaporkan sehingga kemampuan generalisasi model terhadap data klinis nyata belum diketahui.
b. Evaluasi segmentasi hanya dilakukan pada 50 kasus, yang tergolong sedikit untuk memperoleh taksiran rerata yang stabil mengingat besarnya sebaran nilai.

Sejumlah arah pengembangan dapat ditempuh, dan urutannya sengaja disusun menurut perbandingan antara dampak dan biaya. Yang paling murah sekaligus paling berdampak adalah meng-evaluasi ulang keempat konfigurasi pada satu himpunan uji yang sama, menala ambang keputusan pada data validasi dengan mengutamakan *recall*, serta melaporkan *recall* klasifikasi sebagai metrik utama pada dasbor. Ketiganya tidak menuntut pelatihan ulang maupun perubahan arsitektur.

Tingkat berikutnya menuntut sedikit pekerjaan tambahan, yaitu menyelesaikan dan melaporkan validasi eksternal pada data rumah sakit, memeriksa secara visual seluruh kasus dengan *Dice* rendah, memverifikasi peta perhatian CBAM terhadap lokasi lesi, serta mengganti penyusunan ulang irisan positif dengan pemilihan rentang irisan yang menjaga kontinuitas volume. Tingkat yang paling ambisius mencakup penggabungan modalitas pada tingkat fitur alih-alih tingkat piksel, penyisipan CBAM pada beberapa tingkat resolusi, penerapan registrasi antarmodalitas yang sesungguhnya, serta perluasan penentuan keparahan dengan memasukkan lokasi anatomis lesi.

➢ Ringkasan

Bab ini telah menguraikan pembangunan sistem dua tahap untuk deteksi dan estimasi keparahan stroke iskemik pada citra MRI, mulai dari prapemrosesan volume tiga dimensi hingga dasbor interaktif. Beberapa hal penting dapat dirangkum sebagai berikut.

a. Konfigurasi terbaik diperoleh pada M3 yang menggunakan *DWI-Only* dengan strategi *Important Slices*, dengan *F1-score* sebesar 0,8598 dan AUC sebesar 0,9195. Namun perbandingan antarkonfigurasi tidak sepenuhnya sahih karena strategi seleksi

irisan turut diterapkan pada himpunan uji, sehingga keempat model tidak menghadapi permasalahan yang setara.

b. Penggabungan tiga modalitas melalui *RGB Fusion* justru berperforma lebih rendah daripada penggunaan DWI saja, yang dapat dijelaskan oleh hilangnya korelasi antarkanal yang diharapkan bobot *ImageNet*, pengenceran sinyal DWI, dan ketiadaan registrasi antarmodalitas.
c. Penerapan *Test-Time Augmentation* menaikkan akurasi sebesar 0,89 poin persentase namun menurunkan *F1-score* sebesar 5,92 poin persentase, sehingga pada sistem yang mengutamakan *recall* konfigurasi tanpa TTA lebih sesuai kecuali disertai penalaan ulang ambang keputusan.
d. Segmentasi menggunakan *NVAutoNet* menghasilkan *Mean Dice Score* sebesar 0,8347 dengan standar deviasi 0,1575, yang menandakan sebaran lebar dan menuntut pelaporan yang distratifikasi menurut ukuran lesi.
e. Nilai *Dice* awal sebesar 0,0866 disebabkan oleh ketidak-selarasan ruang fisik antara berkas prediksi dan *ground truth*, bukan oleh kegagalan model, dan meningkat menjadi 0,8347 setelah dilakukan penyelarasan menggunakan SimpleITK.
f. Rancangan dua tahap menjadikan *recall* tahap klasifikasi sebagai batas atas kemampuan sistem menemukan lesi, sebab tidak terdapat jalur yang mengembalikan irisan bernilai negatif untuk diperiksa ulang.
g. Estimasi volume lesi dapat dihitung secara otomatis dari masker segmentasi, tetapi penentuan tingkat keparahan menuntut informasi tambahan berupa lokasi anatomis dan waktu sejak awitan yang belum tercakup pada sistem ini.

Gagasan yang paling penting untuk dibawa dari bab ini berkaitan dengan cara menyikapi angka yang mengecewakan. Ketika evaluasi segmentasi menghasilkan nilai 0,0866, tersedia dua jalan: menyimpulkan bahwa model tidak memadai, atau menelusuri apakah yang dibandingkan memang benar. Jalan pertama tampak lebih masuk akal dan hampir selalu lebih mudah ditempuh. Jalan kedua menuntut kesediaan untuk meragukan alur kerja sendiri sebelum meragukan model orang lain, dan pada kasus ini jalan itulah yang benar. Sepuluh kali lipat perbaikan diperoleh bukan dari

arsitektur yang lebih baik, melainkan dari satu langkah penyelarasan koordinat yang seluruhnya bersifat teknis.

Buku ini ditutup dengan pengamatan yang menghubungkan kelima bab. Bab 1 memperlihatkan bahwa model dapat benar karena alasan yang keliru. Bab 2 memperlihatkan bahwa satu model dapat tampak baik atau buruk bergantung pada cara pembacaan metriknya. Bab 3 memperlihatkan bahwa sebuah angka dapat sepenuhnya benar secara perhitungan namun menyesatkan tanpa pembanding. Bab 4 memperlihatkan bahwa perbandingan yang tampak adil dapat runtuh oleh hal-hal yang tidak muncul pada tabel konfigurasi. Bab ini memperlihatkan bahwa kegagalan yang paling menyerupai kegagalan model justru dapat berasal dari satu baris kode yang tidak pernah ditulis.

Kelimanya menunjuk pada satu keterampilan yang sama, yaitu kesediaan untuk memeriksa kembali apa yang sesungguhnya sedang diukur sebelum menerima angka yang dihasilkannya.

➢ Daftar Pustaka

Abumihsan, A., dkk. (2025). Hybrid *Deep Learning* Model Combining DenseNet121, MobileNetV3, and CBAM for Ischemic Stroke Detection Using CT Images. Journal of Medical Imaging and Health Informatics.

Bojsen, J. A., dkk. (2024). *Artificial Intelligence* in Stroke Imaging: A Narrative Review of Automated Analysis on MRI. European Journal of Radiology.

Chattopadhay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018). *Grad-CAM*++: Improved Visual Explanations for Deep Convolutional Networks. Dalam IEEE Winter Conference on Applications of Computer Vision (WACV), hlm. 839–847.

Hernandez Petzsche, M. R., dkk. (2022). ISLES 2022: A Multi-Center Magnetic Resonance Imaging Stroke Lesion Segmentation Dataset. Scientific Data, 9(1), 762.

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. Dalam Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), hlm. 4700–4708.

Isensee, F., Jaeger, P. F., Kohl, S. A. A., Petersen, J., & Maier-Hein, K. H. (2021). nnU-Net: A Self-Configuring Method for *Deep Learning*-Based Biomedical Image Segmentation. Nature Methods, 18(2), 203–211.

Kingma, D. P., & Ba, J. (2017). Adam: A Method for Stochastic Optimization. Dalam International Conference on Learning Representations (ICLR).

Lansberg, M. G., dkk. (2000). Comparison of Diffusion-Weighted MRI and CT in Acute Stroke. Neurology, 54(8), 1557–1561.

Loshchilov, I., & Hutter, F. (2019). Decoupled Weight Decay Regularization. Dalam International Conference on Learning Representations (ICLR).

Loshchilov, I., & Hutter, F. (2017). SGDR: Stochastic Gradient Descent with Warm Restarts. Dalam International Conference on Learning Representations (ICLR).

Pizer, S. M., dkk. (1987). Adaptive Histogram Equalization and Its Variations. Computer Vision, Graphics, and Image Processing, 39(3), 355–368.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Dalam Medical Image Computing and Computer-Assisted Intervention (MICCAI), hlm. 234–241.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. Dalam Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), hlm. 2818–2826.

Tedyanto, E. H., dkk. (2022). Magnetic Resonance Imaging in Acute Ischemic Stroke. Cureus, 14(7).

Thomalla, G., dkk. (2018). MRI-Guided Thrombolysis for Stroke with Unknown Time of Onset. New England Journal of Medicine, 379(7), 611–622.

Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). CBAM: Convolutional Block Attention Module. Dalam Proceedings of the European Conference on Computer Vision (ECCV), hlm. 3–19.

World Stroke Organization. (2023). Global Stroke Fact Sheet. Jenewa: WSO.

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). Mixup: Beyond Empirical Risk Minimization. Dalam International Conference on Learning Representations (ICLR).

BAB 6
PREDIKSI KOMPLIKASI MEDIS PASIEN HEMODIALISIS MENGGUNAKAN *XGBOOST* DENGAN OPTIMALISASI *ARTIFICIAL BEE COLONY*

Setelah membaca bab ini, pembaca diharapkan mampu:

a. Menjelaskan mekanisme komplikasi intradialitik pada pasien hemodialisis serta parameter klinis yang menjadi penanda risikonya.
b. Menguraikan prinsip kerja *XGBoost*, mulai dari fungsi objektif berregularisasi hingga penurunan bobot daun optimal melalui ekspansi Taylor orde dua.
c. Menjelaskan cara kerja algoritma *Artificial Bee Colony* dan alasan penggunaannya untuk penalaan *hyperparameter* dibandingkan Grid Search maupun Random Search.
d. Menerapkan SMOTE untuk menangani ketidakseimbangan kelas serta mengenali kesalahan urutan penerapan yang menimbulkan kebocoran data.
e. Membandingkan skema validasi *hold-out* dan *k-fold cross-validation*, serta menggunakan selisih di antara keduanya sebagai alat diagnosis.
f. Membaca *confusion matrix* multikelas untuk menemukan kekeliruan yang paling merugikan secara klinis.
g. Menafsirkan *feature* importance secara kritis, termasuk mengenali bias yang melekat pada metode penghitungannya.

➢ Pendahuluan

Penyakit Ginjal Kronik merupakan masalah kesehatan yang menarik perhatian dunia. *World Health Organization* pada tahun 2021 mencatat lebih dari 843,6 juta individu di seluruh dunia menderita penyakit ini, atau sekitar 15% populasi global. Angka kematiannya mencapai 1,2 juta jiwa pada tahun 2019, meningkat sebesar 31,7% dibandingkan kondisi tahun 2010, sehingga menempatkannya pada peringkat kedua belas sebagai penyebab utama kematian global. Di Indonesia, Riset Kesehatan Dasar mencatat lonjakan yang sangat

tajam, dari 11.689 penderita gagal ginjal kronik pada tahun 2013 menjadi 713.783 penderita pada tahun 2018.

Ketika fungsi ginjal menurun hingga tahap akhir, pasien memerlukan terapi pengganti fungsi ginjal. Di antara tiga pilihan yang tersedia hemodialisis, dialisis peritoneal, dan transplantasi ginjal hemodialisis merupakan yang paling banyak dijalani. Laporan Registri Ginjal Indonesia menunjukkan peningkatan jumlah pasien hemodialisis dari 1.885 orang pada tahun 2007 menjadi sekitar 21 ribu pada tahun 2013, dan melonjak menjadi 130 ribu pada tahun 2020.

Hemodialisis bekerja dengan mengalirkan darah pasien melalui ginjal buatan untuk membuang zat sisa metabolisme dan kelebihan cairan. Meskipun demikian, terapi ini tidak dapat menggantikan seluruh fungsi ginjal, dan justru proses terapinya sendiri dapat menimbulkan berbagai komplikasi. Laporan Indonesia Renal Registry tahun 2020 menempatkan hipertensi intradialitik sebagai penyulit terbanyak dengan proporsi mencapai 30%, disusul hipotensi yang meningkat tajam dari 14% menjadi 27% dibandingkan laporan sebelumnya. Laporan yang sama mencatat bahwa penyebab kematian terbanyak pada pasien hemodialisis adalah komplikasi kardiovaskular dengan proporsi 42%.

Permasalahan pada bab ini berbeda secara mendasar dari lima bab sebelumnya, dan perbedaannya terletak pada jenis data yang dianalisis. Kelima bab terdahulu seluruhnya bekerja pada citra rontgen dada, MRI otak, citra mikroskopis, citra OCT retina, dan kembali MRI otak. Bab ini bekerja pada data tabular berupa rekam medis: usia, jenis kelamin, tekanan darah, kadar hemoglobin, kreatinin, ureum, dan sejumlah parameter klinis lainnya. Pergeseran ini membawa konsekuensi yang menyeluruh, dan konsekuensi pertamanya adalah bahwa jaringan saraf dalam bukan lagi pilihan yang paling tepat.

Pada data tabular, metode berbasis pohon keputusan yang diperkuat melalui *gradient boosting* secara konsisten mengungguli jaringan saraf dalam pada berbagai perbandingan empiris. Alasannya berkaitan dengan sifat datanya. Citra memiliki struktur spasial yang kuat dan hubungan antarpiksel yang teratur, dan konvolusi dirancang khusus untuk memanfaatkan keteraturan itu. Data tabular tidak memiliki keteraturan semacam itu: kolom-

kolomnya tidak memiliki urutan yang bermakna, satuannya berbeda-beda, sebagian bersifat kategorik, dan hubungan antara fitur dan target sering kali berbentuk ambang yang tegas, misalnya tekanan darah sistolik di atas nilai tertentu. Pohon keputusan memodelkan ambang secara alamiah karena memang bekerja dengan pemisahan bernilai ambang.

Konsekuensi kedua berkaitan dengan jumlah data. Bab ini menggunakan 387 baris data pasien, yaitu jumlah yang sangat kecil dibandingkan puluhan ribu citra pada bab-bab sebelumnya. Jumlah sekecil ini mengubah hampir seluruh pertimbangan metodologis: pembagian data menjadi lebih rentan terhadap keacakan, validasi silang menjadi lebih penting daripada pembagian tunggal, penanganan ketidakseimbangan kelas menjadi lebih menentukan, dan risiko kebocoran data menjadi jauh lebih besar.

Bab ini membahas pembangunan model prediksi komplikasi hemodialisis menggunakan *Extreme Gradient Boosting* (*XGBoost*) yang *hyperparameter*-nya ditala menggunakan algoritma *Artificial Bee Colony* (ABC). Delapan parameter utama dioptimalkan, dan sebanyak 72 model diuji melalui kombinasi data seimbang dan tidak seimbang dengan dua skema pembagian data. Selain menguraikan proses dan hasilnya, bab ini akan menelusuri satu pertanyaan yang muncul dari hasilnya sendiri: mengapa akurasi pada skema *hold-out* mencapai 96% sedangkan pada validasi silang hanya 87%, dan apa arti selisih sebesar sembilan poin persentase itu. Penelusuran tersebut merupakan inti pelajaran pada bab ini.

➢ Pengantar Teori

Bagian ini menguraikan landasan teori yang diperlukan pada bab ini. Karena permasalahan yang dihadapi bukan lagi pengolahan citra, sebagian besar konsep pada bab-bab sebelumnya tidak lagi relevan dan digantikan oleh konsep yang khas bagi data tabular serta metode berbasis pohon.

Hemodialisis dan Komplikasi Intradialitik

Hemodialisis bekerja dengan mengalirkan darah pasien melalui suatu membran semipermeabel yang dikenal sebagai dialiser, tempat zat sisa metabolisme berdifusi ke dalam cairan dialisat dan kelebihan cairan tubuh ditarik keluar melalui perbedaan tekanan. Satu sesi

berlangsung sekitar empat jam dan umumnya dilakukan tiga kali seminggu. Dalam rentang waktu tersebut, sejumlah cairan yang terkumpul selama dua sampai tiga hari harus dikeluarkan sekaligus, dan kecepatan pengeluaran inilah yang menjadi sumber sebagian besar komplikasi.

Tiga komplikasi yang menjadi fokus bab ini beserta mekanismenya diuraikan pada Tabel 6.1.

Tabel 6.1 Tiga Komplikasi Intradialitik yang Dianalisis

Komplikasi	**Mekanisme Singkat**	**Manifestasi Klinis**
Hipertensi intradialitik	Aktivasi sistem renin-angiotensin dan pengeluaran vasokonstriktor selama sesi	Tekanan darah justru naik selama atau setelah dialisis
Hipotensi intradialitik	Penarikan cairan lebih cepat daripada kemampuan pengisian ulang dari ruang antarsel	Pusing, mual, kram, hingga kehilangan kesadaran
Gangguan gastrointestinal	Penurunan perfusi saluran cerna dan penumpukan zat uremik	Mual, muntah, nyeri perut, perdarahan saluran cerna

Catatan Klinis: Hipotensi Merupakan yang Paling Berbahaya

Ketiga komplikasi tersebut tidak setara bobot risikonya, dan hal ini akan menjadi penting ketika membaca hasil pada Subbab 6.4.7.

Hipertensi intradialitik berkembang relatif perlahan dan umumnya masih memberi waktu bagi tenaga medis untuk merespons. Gangguan gastrointestinal menimbulkan ketidaknyamanan yang nyata namun jarang mengancam nyawa secara langsung.

Hipotensi intradialitik berbeda. Penurunan tekanan darah dapat terjadi dalam hitungan menit dan berujung pada kehilangan

kesadaran di tengah sesi, ketika pasien sedang terhubung dengan mesin melalui akses vaskular. Selain risiko langsung tersebut, episode hipotensi berulang dikaitkan dengan penurunan perfusi otak dan jantung yang bersifat kumulatif, sehingga meningkatkan risiko stroke dan kejadian kardiovaskular pada jangka panjang.
Konsekuensinya bagi sistem prediksi bersifat langsung: kegagalan mengenali risiko hipotensi merupakan jenis kesalahan yang paling merugikan, dan sistem semacam ini seharusnya dievaluasi dengan memberikan bobot lebih besar pada kelas tersebut.

Parameter klinis yang digunakan sebagai prediktor pada bab ini beserta maknanya dirangkum pada Tabel 6.2. Memahami makna klinis setiap variabel bukan sekadar pelengkap, sebab hasil analisis kepentingan fitur pada Subbab 6.4.9 hanya dapat dinilai kewajarannya apabila pembaca mengetahui apa yang sesungguhnya diukur oleh setiap variabel.

Tabel 6.2 Variabel Klinis pada Data Rekam Medis Pasien Hemodialisis

Variabel	**Tipe**	**Makna Klinis**
Jenis kelamin	Kategorik	Faktor risiko dasar
Usia	Numerik	Berkaitan dengan kekakuan pembuluh dan cadangan fisiologis
Lama hemodialisis	Numerik	Durasi menjalani terapi dalam bulan
Tekanan sistolik	Numerik	Tekanan puncak saat jantung berkontraksi (mmHg)
Tekanan diastolik	Numerik	Tekanan saat jantung berelaksasi (mmHg)
Hemoglobin	Numerik	Penanda anemia, lazim rendah pada gagal ginjal
Kreatinin	Numerik	Zat sisa otot; penanda utama fungsi ginjal
Ureum	Numerik	Zat sisa protein; penanda beban uremik

SI-TIBC	Numerik	Saturasi transferin; penanda status zat besi
EPO 3000 dan EPO 2000	Kategorik	Pemberian *eritropoietin* untuk mengatasi anemia
Tron Sucrose	Numerik	Suplemen zat besi intravena
Riwayat penyakit	Kategorik	Penyakit penyerta pada pasien
Riwayat keluarga	Kategorik	Penyakit penyerta pada keluarga

Satu hal perlu dicatat mengenai variabel EPO dan Tron Sucrose. Ketiganya bukan merupakan pengukuran kondisi pasien, melainkan catatan mengenai tindakan yang telah diberikan kepada pasien. Variabel semacam ini disebut variabel terapi, dan keberadaannya dalam model prediksi menuntut kehati-hatian: pemberian *eritropoietin* merupakan akibat dari kondisi anemia, bukan penyebabnya. Model yang menemukan hubungan antara pemberian EPO dan komplikasi sesungguhnya sedang menangkap jejak keputusan klinis dokter, bukan hubungan fisiologis.

Pohon Keputusan dan *Gradient Boosting*

Pohon keputusan bekerja dengan membagi ruang fitur secara berulang menggunakan aturan bernilai ambang. Pada setiap simpul, algoritma memilih satu fitur beserta satu nilai ambang yang paling memisahkan data menurut suatu ukuran ketidakmurnian. Proses berlanjut hingga tercapai kriteria penghentian, dan setiap daun memberikan prediksi berupa kelas atau nilai tertentu.

Keunggulan pohon keputusan terletak pada kesesuaiannya dengan sifat data tabular. Pohon tidak memerlukan penskalaan fitur karena hanya membandingkan nilai terhadap ambang; pohon menangani fitur numerik dan kategorik secara berdampingan tanpa perlakuan khusus; dan pohon secara alamiah memodelkan interaksi antarfitur melalui percabangan bertingkat. Kelemahannya adalah ketidakstabilan: perubahan kecil pada data dapat menghasilkan struktur pohon yang sangat berbeda.

Gradient boosting mengatasi kelemahan tersebut dengan membangun banyak pohon secara berurutan, di mana setiap pohon baru dilatih untuk memperbaiki kesalahan gabungan pohon-pohon sebelumnya. Prediksi akhir merupakan penjumlahan keluaran seluruh pohon.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in F \quad (6.1)$$

Dengan K menyatakan jumlah pohon, f indeks k menyatakan pohon ke-k, dan F menyatakan himpunan seluruh struktur pohon regresi dan klasifikasi yang mungkin dibentuk. Perbedaan mendasar antara *boosting* dan *bagging* pendekatan yang digunakan pada *Random Forest* terletak pada urutan pembangunannya. Pada *bagging*, seluruh pohon dibangun secara paralel dan saling bebas, lalu hasilnya dirata-ratakan. Pada *boosting*, setiap pohon bergantung pada hasil pohon sebelumnya, sehingga model secara bertahap memusatkan perhatian pada kasus yang masih sulit.

XGBoost: Fungsi Objektif dan Penurunannya

XGBoost yang dikembangkan Tianqi Chen pada tahun 2014 merupakan penyempurnaan *gradient boosting* dengan dua tambahan utama, yaitu regularisasi eksplisit pada fungsi objektif dan penggunaan turunan orde kedua. Fungsi objektifnya dinyatakan sebagai berikut.

$$O = \sum_{i=1}^{n} L(y_i, F(x_i)) + \sum_{k=1}^{K} \Omega(f_k) \quad (6.2)$$

Suku pertama merupakan fungsi kerugian yang mengukur selisih antara nilai sebenarnya dan prediksi, sedangkan suku kedua merupakan penalti terhadap kompleksitas setiap pohon. Kehadiran suku kedua inilah yang membedakan *XGBoost* dari *gradient boosting* konvensional, dan bentuknya diuraikan sebagai berikut.

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} {w_j}^2 \quad (6.3)$$

Dengan T menyatakan jumlah daun pada pohon, w indeks j menyatakan bobot daun ke-j, γ menyatakan penalti untuk setiap tambahan daun, dan λ menyatakan koefisien regularisasi L2 terhadap bobot daun. Kedua parameter tersebut bekerja pada aspek yang berbeda: γ menekan jumlah daun sehingga pohon menjadi lebih dangkal, sedangkan λ menekan besarnya bobot pada setiap daun sehingga prediksi menjadi lebih moderat.

Pada iterasi ke-t, fungsi objektifnya menjadi sebagai berikut, dengan pohon-pohon sebelumnya dianggap tetap.

$$Obj^{(t)} = \sum_{i=1}^{n} L(y_i, \hat{y}^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (6.4)$$

Bentuk tersebut belum dapat dioptimalkan secara langsung karena fungsi kerugiannya dapat berbentuk apa saja. *XGBoost* mengatasinya dengan mendekati fungsi kerugian menggunakan ekspansi Taylor sampai orde kedua.

$$Obj^{(t)} \approx \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^{\,2}(x_i)] + \Omega(f_t) \quad (6.5)$$

Dengan g indeks i dan h indeks i berturut-turut menyatakan turunan pertama dan kedua dari fungsi kerugian terhadap prediksi. Penggunaan turunan kedua merupakan keunggulan penting *XGBoost* dibandingkan *gradient boosting* konvensional yang hanya memanfaatkan turunan pertama, sebab informasi kelengkungan memungkinkan penentuan besar langkah yang lebih tepat pada setiap iterasi. Dengan mengelompokkan sampel menurut daun tempatnya jatuh, bentuk tersebut dapat disederhanakan menjadi penjumlahan atas daun.

$$Obj^{(t)} = \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^{\,2}] + \gamma T \quad (6.6)$$

Bentuk ini merupakan fungsi kuadrat terhadap bobot daun, sehingga nilai optimalnya dapat diperoleh secara analitik dengan menyamakan turunannya terhadap nol.

$$w_j^{\,*} = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \quad (6.7)$$

Substitusi kembali menghasilkan nilai objektif optimal bagi suatu struktur pohon tertentu.

$$Obj^{*} = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \quad (6.8)$$

Persamaan 6.8 memiliki makna praktis yang penting. Nilai tersebut berperan sebagai skor kualitas bagi sebuah struktur pohon, sehingga ketika algoritma mempertimbangkan pemisahan pada suatu simpul, ia dapat menghitung berapa besar perbaikan yang

dihasilkan pemisahan itu. Apabila perbaikannya lebih kecil daripada γ, pemisahan tidak dilakukan. Dengan demikian γ berfungsi sebagai ambang minimum manfaat sebuah pemisahan, dan inilah mekanisme pemangkasan yang bekerja secara otomatis di dalam *XGBoost*.
Kedelapan parameter yang ditala pada bab ini beserta perannya dirangkum pada Tabel 6.3.

Tabel 6.3 Delapan *Hyperparameter XGBoost* yang Ditala

Parameter	**Peran**	**Arah Pengaruh**
n_estimators	Jumlah pohon yang dibangun	Terlalu banyak berisiko *overfitting*
learning_rate (η)	Bobot kontribusi setiap pohon baru	Nilai kecil memerlukan lebih banyak pohon
max_depth	Kedalaman maksimum setiap pohon	Makin dalam makin kompleks
min_child_weight	Bobot minimum pada satu daun	Nilai kecil membolehkan daun bersampel sedikit
gamma (γ)	Ambang minimum manfaat pemisahan	Nilai besar menghasilkan pohon lebih dangkal
subsample	Proporsi sampel per pohon	Nilai kecil menambah keragaman
colsample_bytree	Proporsi fitur per pohon	Nilai kecil mengurangi korelasi antarpohon
reg_lambda (λ)	Regularisasi L2 pada bobot daun	Nilai besar memoderasi prediksi

Delapan parameter dengan rentang nilai yang luas menghasilkan ruang pencarian yang sangat besar. Apabila setiap parameter hanya diuji pada sepuluh nilai saja, jumlah kombinasi yang harus dievaluasi mencapai sepuluh pangkat delapan. Kenyataan inilah yang menjadikan penalaan secara menyeluruh tidak mungkin dilakukan dan menuntut pendekatan pencarian yang lebih cerdas.

Algoritma *Artificial Bee Colony*

Artificial Bee Colony merupakan algoritma optimasi yang terinspirasi dari perilaku koloni lebah madu dalam mencari sumber makanan. Dalam kerangka algoritma ini, setiap posisi sumber makanan merepresentasikan satu kandidat solusi pada konteks bab ini berupa satu kombinasi nilai kedelapan *hyperparameter* sedangkan jumlah nektar pada sumber tersebut merepresentasikan kualitas solusinya.

Koloni dibagi menjadi tiga peran yang bekerja bergantian pada setiap iterasi. *Lebah pekerja* mengeksplorasi wilayah di sekitar sumber makanan yang sudah diketahui, mencari perbaikan lokal. *Lebah pengamat* memilih sumber makanan yang akan dieksplorasi lebih lanjut berdasarkan probabilitas yang sebanding dengan kualitasnya, sehingga solusi yang menjanjikan memperoleh perhatian lebih besar. *Lebah penjelajah* meninggalkan sumber makanan yang tidak membaik selama sejumlah percobaan tertentu dan menggantinya dengan sumber baru yang dipilih secara acak.

Populasi awal dibangkitkan secara acak di dalam batas ruang pencarian.

$$x_{ij} = u_j^L + rand(0,1) \cdot (u_j^U - u_j^L) \quad (6.9)$$

Dengan x indeks ij menyatakan nilai parameter ke-j pada solusi ke-i, serta u beraksen L dan U berturut-turut menyatakan batas bawah dan batas atas ruang pencarian pada dimensi tersebut. Pada fase lebah pekerja, solusi baru dibentuk dengan menggeser satu dimensi solusi lama ke arah selisihnya terhadap solusi tetangga yang dipilih secara acak.

$$v_{ij} = x_{ij} + \varphi (x_{ij} - x_{kj}) \quad (6.10)$$

Dengan φ merupakan bilangan acak berdistribusi seragam pada rentang −1 sampai 1. Persamaan ini merupakan inti mekanisme ABC dan patut diperhatikan dengan cermat. Besar langkah perubahan sebanding dengan selisih antara dua solusi, sehingga ketika populasi masih tersebar luas pada awal pencarian, langkahnya besar dan bersifat menjelajah; ketika populasi mulai memusat menjelang akhir, selisihnya mengecil dan langkahnya menjadi halus. Dengan kata lain, peralihan dari eksplorasi menuju eksploitasi terjadi secara otomatis tanpa perlu dijadwalkan.

Pada fase lebah pengamat, probabilitas sebuah sumber makanan dipilih ditentukan secara sebanding dengan kualitasnya.

$$P_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} (6.11)$$

Dengan fit menyatakan nilai kebugaran solusi dan SN menyatakan jumlah sumber makanan. Fase lebah penjelajah tidak memiliki persamaan tersendiri, melainkan bekerja melalui aturan sederhana: apabila suatu sumber makanan tidak membaik selama sejumlah percobaan yang ditetapkan oleh parameter *limit*, sumber tersebut ditinggalkan dan digantikan oleh solusi baru yang dibangkitkan menggunakan Persamaan 6.9.

Catatan: Mengapa ABC dan Bukan *Grid Search*

Perbandingan antara ABC dan metode pencarian tradisional dapat dipahami melalui satu perbedaan mendasar, yaitu apakah pengetahuan yang telah diperoleh dimanfaatkan atau tidak.

Grid Search mengevaluasi seluruh kombinasi pada suatu kisi yang telah ditetapkan. Metode ini menjamin cakupan yang lengkap namun biayanya tumbuh secara eksponensial terhadap jumlah parameter. Pada delapan parameter, pendekatan ini praktis mustahil.

Random Search mengambil kombinasi secara acak dan pada praktiknya sering mengungguli Grid Search karena tidak membuang upaya pada parameter yang ternyata tidak berpengaruh. Namun setiap percobaannya sepenuhnya bebas dari percobaan sebelumnya, informasi mengenai wilayah yang menjanjikan tidak dimanfaatkan sama sekali.

ABC memanfaatkan informasi itu. Solusi yang baik menjadi titik awal pencarian berikutnya melalui Persamaan 6.10, sedangkan mekanisme lebah penjelajah mencegah pencarian terjebak pada satu wilayah saja. Inilah alasan pendekatan populasi umumnya lebih efisien pada ruang pencarian berdimensi tinggi.

Perlu ditambahkan bahwa keunggulan ini tidak gratis. ABC sendiri memiliki tiga parameter yang harus ditetapkan jumlah lebah, jumlah iterasi maksimum, dan *limit*, sehingga persoalan penalaan tidak hilang melainkan berpindah ke tingkat yang lebih tinggi. Bedanya, performa ABC relatif tidak peka terhadap

ketiga parameter tersebut, sehingga nilai bawaan yang wajar umumnya sudah memadai.

Fungsi kebugaran yang digunakan pada bab ini adalah galat klasifikasi, sehingga proses optimasi berupaya meminimumkannya.

$$fitness = 1 - \frac{1}{m}\sum_{s=1}^{m} Acc_s \qquad (6.12)$$

Dengan m menyatakan jumlah lipatan pada validasi silang atau bernilai satu pada skema *hold-out*. Pemilihan data tempat nilai kebugaran ini dihitung merupakan keputusan yang jauh lebih menentukan daripada yang tampak, dan hal ini akan dibahas kembali pada Subbab 6.4.5.

Synthetic Minority Oversampling Technique

Ketidakseimbangan kelas menimbulkan persoalan yang telah dibahas pada Bab 2 dan Bab 3: model cenderung memusatkan perhatian pada kelas mayoritas karena hal itu sudah cukup untuk menekan fungsi kerugian. Terdapat tiga pendekatan umum untuk mengatasinya, yaitu mengurangi kelas mayoritas, memperbanyak kelas minoritas, atau memberikan bobot berbeda pada fungsi kerugian. SMOTE termasuk pendekatan kedua, namun dengan cara yang lebih cerdas daripada sekadar menggandakan data.

Jumlah data sintetis yang perlu dibangkitkan ditentukan berdasarkan persentase pengulangan yang dikehendaki.

$$N\% = \frac{jumlah\ data\ kelas\ mayoritas}{jumlah\ data\ kelas\ minoritas} \times 100\% \qquad (6.13)$$

Untuk setiap data pada kelas minoritas, dicari sejumlah tetangga terdekat menggunakan jarak Euclidean.

$$d(x, y) = \sqrt{\sum_{p=1}^{n}(x_p - y_p)^2} \qquad (6.14)$$

Data sintetis kemudian dibangkitkan sebagai titik acak pada garis yang menghubungkan sebuah data asli dengan salah satu tetangga terdekatnya.

$$x_{syn} = x_i + (x_{knn} - x_i) \cdot \delta \qquad (6.15)$$

Dengan δ merupakan bilangan acak pada rentang nol sampai satu. Apabila δ mendekati nol, data sintetis nyaris identik dengan data asli; apabila mendekati satu, data sintetis nyaris identik dengan tetangganya; dan apabila berada di sekitar 0,5, data sintetis terletak di tengah keduanya. Gagasan di balik pendekatan ini adalah bahwa wilayah di antara dua data kelas minoritas kemungkinan besar juga merupakan wilayah kelas minoritas, sehingga membangkitkan data di sana lebih bermakna daripada sekadar menggandakan data yang ada.

Penggunaan jarak Euclidean pada Persamaan 6.14 mengandung satu tuntutan yang mudah terlewat, yaitu bahwa seluruh fitur harus berada pada skala yang sebanding. Pada data rekam medis, kadar ureum berkisar pada ratusan sedangkan hemoglobin berkisar pada satuan; tanpa penskalaan, perhitungan jarak akan sepenuhnya didominasi oleh fitur bernilai besar. Selain itu, penerapan SMOTE pada fitur kategorik hasil *label encoding* menghasilkan nilai antara yang tidak memiliki makna data sintetis dengan nilai jenis kelamin sebesar 0,4 tidak merepresentasikan pasien mana pun.

Skema Validasi dan Metrik Evaluasi

Dua skema pembagian data digunakan pada bab ini. Skema *hold-out* membagi data satu kali menjadi bagian pelatihan dan pengujian dengan rasio tertentu. Skema *k-fold cross-validation* membagi data menjadi k bagian, lalu melatih dan menguji model sebanyak k kali dengan setiap bagian bergiliran menjadi data uji. Akurasi akhir merupakan rata-rata atas seluruh lipatan.

$$Acc_{CV} = \frac{1}{k}\sum_{f=1}^{k} Acc_f \qquad (6.16)$$

Perbedaan keduanya menjadi sangat penting pada data berukuran kecil seperti pada bab ini. Dengan 387 baris data, skema *hold-out* 75:25 menyisakan sekitar 97 data uji saja, sehingga hasilnya sangat bergantung pada kebetulan pembagian. Validasi silang menggunakan seluruh data secara bergiliran sebagai data uji, sehingga taksirannya jauh lebih stabil. Sebagai konsekuensinya, ketika kedua skema menghasilkan angka yang berbeda jauh, taksiran dari validasi silang jauh lebih layak dipercaya.

Metrik evaluasi yang digunakan berupa *accuracy*, *precision*, *recall*, dan *F1-score* yang seluruhnya telah dibahas pada Bab 1 dan diperluas ke ranah multikelas pada Bab 4 melalui Persamaan 4.3 dan 4.4. Pada bab ini digunakan rerata makro yang memperlakukan setiap kelas secara setara, sehingga kelas berperforma buruk tetap terlihat pengaruhnya pada angka rerata.

Metrik terakhir yang digunakan adalah AUC-ROC yang diperluas ke ranah multikelas melalui pendekatan satu-lawan-sisanya. Bagi setiap kelas, seluruh kelas lain digabungkan menjadi kelas negatif, lalu kurva ROC dan luas di bawahnya dihitung seperti pada kasus biner. Perlu diperhatikan bahwa AUC dihitung dari nilai probabilitas dan tidak bergantung pada ambang keputusan, sehingga sebuah model dapat memiliki AUC tinggi namun tetap menghasilkan banyak kekeliruan pada ambang bawaan sebesar 0,5.

Kepentingan Fitur dan Biasnya

Salah satu keunggulan metode berbasis pohon dibandingkan jaringan saraf dalam adalah ketersediaan ukuran kepentingan fitur secara langsung. Terdapat tiga cara menghitungnya, dan ketiganya dapat menghasilkan peringkat yang berbeda untuk model yang sama.

a. *Weight* atau *F-score* menghitung berapa kali sebuah fitur digunakan sebagai dasar pemisahan pada seluruh pohon.
b. *Gain* menghitung rata-rata perbaikan fungsi objektif yang dihasilkan pemisahan berdasarkan fitur tersebut, sesuai Persamaan 6.8.
c. *Cover* menghitung rata-rata jumlah sampel yang terpengaruh oleh pemisahan berdasarkan fitur tersebut.

Perhatian: Ukuran Weight Bias terhadap Fitur Kontinu
Ukuran *weight* merupakan yang paling sering ditampilkan karena merupakan bawaan pada banyak implementasi, sekaligus yang paling mudah disalahtafsirkan.
Persoalannya terletak pada jumlah kemungkinan pemisahan. Fitur kontinu seperti kadar ureum dengan ratusan nilai berbeda menyediakan ratusan titik ambang yang mungkin, sehingga fitur tersebut berpeluang besar terpilih berulang kali di berbagai simpul. Fitur biner seperti jenis kelamin hanya menyediakan

satu titik pemisahan, sehingga betapapun kuat pengaruhnya, ia hanya dapat digunakan sekali pada setiap jalur pohon.
Akibatnya, peringkat berdasarkan *weight* sebagian mencerminkan kardinalitas fitur dan bukan semata kekuatan pengaruhnya. Sebuah fitur biner yang sangat menentukan dapat tampak berada di peringkat bawah, sementara fitur kontinu yang pengaruhnya lemah namun bervariasi dapat tampak menonjol.
Praktik yang lebih baik adalah melaporkan ukuran *gain* yang secara langsung mengukur seberapa besar perbaikan yang dihasilkan setiap fitur, atau menggunakan nilai SHAP yang memberikan atribusi konsisten pada tingkat prediksi individual. Hal ini akan menjadi relevan ketika membaca Gambar 6.9 pada Subbab 6.4.9.

Biaya Kekeliruan dan Aturan Keputusan

Konsep terakhir pada bagian teori ini jarang dibahas pada buku pembelajaran mesin umum, namun sangat menentukan pada penerapan klinis. Aturan keputusan bawaan pada klasifikasi multikelas adalah memilih kelas dengan probabilitas tertinggi.

$$\hat{y} = arg\,max\,P(k \mid x),\ \ k \in \{0,1,2,3\} \quad (6.17)$$

Aturan ini memperlakukan seluruh jenis kekeliruan sebagai setara. Pada permasalahan bab ini, asumsi tersebut jelas tidak berlaku: memprediksi pasien berisiko hipotensi sebagai tanpa komplikasi memiliki konsekuensi yang jauh lebih berat daripada memprediksi pasien tanpa komplikasi sebagai berisiko hipertensi. Kerangka yang lebih tepat adalah keputusan berbasis biaya, dengan mendefinisikan matriks biaya C yang menyatakan kerugian akibat memprediksi kelas i padahal kelas sebenarnya adalah j, lalu memilih kelas yang meminimumkan harapan biayanya.

$$\hat{y} = arg\,min\,\sum_{j=0}^{K-1} C(i,j) \cdot P(j \mid x) \quad (6.18)$$

Penyusunan matriks biaya semacam ini bukan pekerjaan perancang sistem, melainkan pekerjaan klinisi. Nilai yang dimasukkan mencerminkan penilaian medis mengenai seberapa besar kerugian setiap jenis kekeliruan, dan penilaian itu hanya dapat diberikan oleh pihak yang memahami konsekuensi klinisnya. Yang menjadi tanggung jawab perancang sistem adalah menyediakan

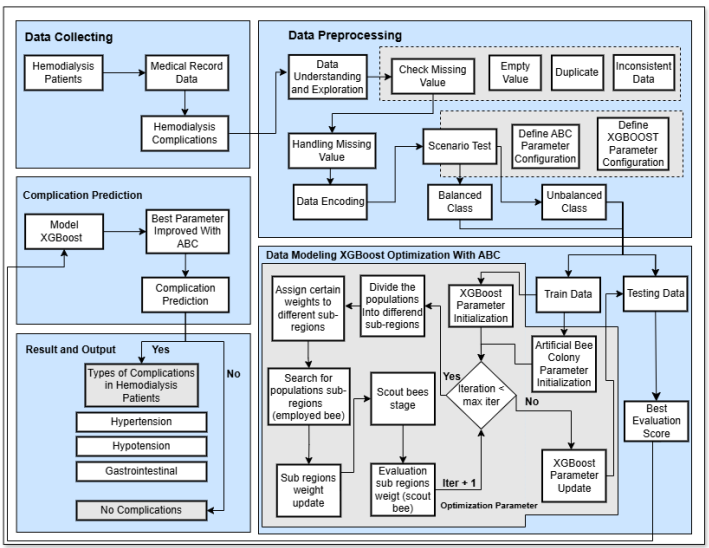
mekanismenya, serta menyatakan secara terbuka bahwa penggunaan aturan bawaan pada Persamaan 6.17 sesungguhnya merupakan penetapan matriks biaya secara diam-diam yaitu matriks yang menganggap seluruh kekeliruan bernilai sama.

Terdapat cara yang lebih sederhana untuk mencapai tujuan serupa, yaitu memberikan bobot berbeda pada setiap kelas saat pelatihan. Namun keduanya bekerja pada tahap yang berbeda dan sebaiknya tidak dipertukarkan. Pembobotan kelas memengaruhi apa yang dipelajari model, sedangkan matriks biaya memengaruhi bagaimana keluaran model ditafsirkan. Keunggulan pendekatan kedua adalah bahwa ia dapat disesuaikan kapan saja tanpa melatih ulang model sifat yang sangat berguna ketika prioritas klinis berubah, misalnya ketika sebuah unit hemodialisis mengalami peningkatan kejadian hipotensi dan ingin menaikkan kewaspadaan untuk sementara waktu.

➢ Metodologi

Alur Metodologi

Keseluruhan Metodologi disusun sebagai lima tahap berurutan, yaitu pengumpulan data, prapemrosesan, pemodelan, evaluasi, dan implementasi sistem. Alur lengkapnya ditunjukkan pada Gambar 6.1.



Gambar 6.1 Alur Metodologi Prediksi Komplikasi Hemodialisis

Blok *Data Collecting* memuat pengumpulan rekam medis pasien hemodialisis beserta catatan komplikasinya. Blok *Data Preprocessing* mencakup pemahaman data, pemeriksaan dan penanganan nilai kosong, pelabelan data kategorik, serta pembentukan dua kondisi dataset berupa kelas seimbang dan tidak seimbang. Blok *Data Modeling* memperlihatkan integrasi *XGBoost* dengan ABC, tempat inisialisasi parameter kedua algoritma dilanjutkan dengan siklus lebah pekerja, lebah pengamat, dan lebah penjelajah hingga tercapai jumlah iterasi maksimum. Blok terakhir memuat pemilihan parameter terbaik, prediksi komplikasi, serta penyajian keluaran berupa empat kategori, yaitu hipertensi, hipotensi, gastrointestinal, dan tanpa komplikasi.

Sumber dan Variabel Data

Data yang digunakan merupakan data sekunder berupa rekam medis pasien hemodialisis pada sebuah rumah sakit umum daerah yang berstatus rujukan regional, dengan periode pencatatan tahun 2022 sampai 2025. Dataset terdiri atas 387 baris dengan 17 kolom yang mencakup identitas tersamarkan, karakteristik demografis, parameter laboratorium, catatan terapi, riwayat penyakit, serta jenis komplikasi yang dialami. Seluruh identitas pasien telah disamarkan sesuai perjanjian kode etik kesehatan.

Variabel target berupa jenis komplikasi dengan empat kategori. Variabel prediktor terdiri atas gabungan variabel numerik dan kategorik sebagaimana telah diuraikan pada Tabel 6.2.

Catatan: Ukuran Data dan Konsekuensinya

Jumlah 387 baris data merupakan angka yang perlu dipegang sepanjang membaca bab ini, sebab hampir seluruh keputusan metodologis dan seluruh keterbatasan hasilnya berpangkal dari sini.

Sebagai perbandingan, Bab 4 menggunakan 24.000 citra dan Bab 5 menggunakan lebih dari 8.000 irisan. Dengan 387 baris yang terbagi ke dalam empat kelas, kelas terkecil hanya memuat 58 sampel. Setelah pembagian 75:25, kelas tersebut menyisakan sekitar 15 sampel pada data uji.

Konsekuensinya bersifat langsung. Sebuah metrik yang dihitung dari 15 sampel memiliki ketelitian sekitar 7 poin persentase per

sampel, sehingga selisih beberapa poin antara dua model tidak dapat dibedakan dari keacakan. Validasi silang menjadi bukan sekadar pilihan yang lebih baik, melainkan keharusan. Dan setiap bentuk kebocoran data, sekecil apa pun, akan memberikan dampak yang jauh lebih besar daripada pada dataset berukuran besar.

Penanganan Nilai Kosong

Pemeriksaan awal menemukan sejumlah nilai kosong pada beberapa atribut. Penanganannya dibedakan menurut tipe data: atribut numerik diisi menggunakan nilai rata-rata, sedangkan atribut kategorik diisi menggunakan modus.

```
# Imputasi atribut numerik dengan rata-rata
data['Sistolik'].fillna(data['Sistolik'].mean(), inplace=True)
data['usia'].fillna(data['usia'].mean(), inplace=True)
data['Hemoglobin'].fillna(data['Hemoglobin'].mean(),
inplace=True)
data['Ureum'].fillna(data['Ureum'].mean(), inplace=True)
data['Kreatinin'].fillna(data['Kreatinin'].mean(), inplace=True)

# Imputasi atribut kategorik dengan modus
data['EPO_3000'].fillna(data['EPO_3000'].mode().iloc[0],
inplace=True)
data['EPO_2000'].fillna(data['EPO_2000'].mode().iloc[0],
inplace=True)
```

Kode 6.1 Imputasi Nilai Kosong

Pendekatan ini sederhana dan lazim digunakan, namun mengandung dua konsekuensi yang perlu disadari. Yang pertama, imputasi dengan rata-rata mengurangi ragam data dan menarik seluruh nilai yang hilang menuju pusat distribusi. Pada variabel klinis, hal ini berarti seorang pasien dengan nilai laboratorium yang tidak tercatat akan diperlakukan seolah-olah memiliki nilai rata-rata, padahal ketiadaan catatan itu sendiri kadang bermakna.

Yang kedua, dan lebih menentukan, imputasi dilakukan menggunakan statistik dari keseluruhan data termasuk data yang kelak menjadi data uji. Praktik yang benar adalah menghitung nilai rata-rata hanya dari data pelatihan, lalu menerapkan nilai tersebut pada data uji. Meskipun pengaruh kebocoran melalui imputasi umumnya kecil, prinsip yang sama akan muncul kembali dalam bentuk yang jauh lebih berdampak pada Subbab 6.3.5.

Pelabelan Data Kategorik

Seluruh atribut kategorik diubah menjadi representasi numerik agar dapat diproses oleh *XGBoost*. Variabel target memperoleh pemetaan khusus, sedangkan variabel prediktor kategorik lainnya diproses menggunakan *LabelEncoder*.

```
# Pemetaan label untuk variabel target
komplikasi_mapping = {
   'tidak ada': 0, 'hipertensi': 1,
   'hipotensi': 2, 'gastrointestinal': 3
}
data['Jenis_Komplikasi'] = data['Jenis_Komplikasi'].map(
   komplikasi_mapping)

# Encoding atribut kategorik lainnya
categorical_columns = ['Jenis_Kelamin', 'EPO_3000',
'EPO_2000',
                'Riwayat_penyakit',
                'Riwayat_penyakit_keluarga']
label_encoder = LabelEncoder()
for col in categorical_columns:
   data[col] = label_encoder.fit_transform(data[col])
```

Kode 6.2 Pelabelan Data Kategorik

Penggunaan *label encoding* pada variabel kategorik nominal seperti riwayat penyakit sesungguhnya menimbulkan pertanyaan, sebab metode ini menetapkan nilai numerik berurutan pada kategori yang tidak memiliki urutan. Diabetes yang dipetakan menjadi 1 dan hipertensi yang dipetakan menjadi 2 seolah menyiratkan bahwa hipertensi bernilai lebih besar. Pada model linier, hal ini merupakan kesalahan serius. Pada model berbasis pohon, dampaknya jauh lebih

ringan karena pohon dapat memisahkan nilai mana pun melalui rangkaian pemisahan bertingkat, meskipun hal itu memerlukan lebih banyak simpul daripada yang seharusnya. Alternatif berupa *one-hot encoding* akan menghilangkan persoalan tersebut dengan konsekuensi bertambahnya jumlah kolom.

Penyeimbangan Data dengan SMOTE

Distribusi kelas pada data awal tidak seimbang, dengan 142 sampel tanpa komplikasi, 93 sampel hipertensi, 93 sampel gastrointestinal, dan hanya 58 sampel hipotensi. SMOTE diterapkan untuk menyetarakan jumlah sampel pada seluruh kelas.

```
# Pisahkan fitur dan target
X = data3.iloc[:, :-1]
Y = data3['Jenis_Komplikasi']

# Terapkan SMOTE untuk menyeimbangkan kelas minoritas
smote = SMOTE(random_state=42)
X_resampled, Y_resampled = smote.fit_resample(X, Y)
```

Kode 6.3 Penyeimbangan Kelas Menggunakan SMOTE

Setelah proses ini, seluruh kelas memiliki 142 sampel sehingga total data menjadi 568 baris. Sekilas langkah ini tampak lurus dan tidak bermasalah. Namun terdapat satu hal yang menentukan segalanya, yaitu kapan langkah ini dilakukan relatif terhadap pembagian data dan hal itu dibahas pada kotak berikut.

Peringatan: Urutan SMOTE dan Pembagian Data Menentukan Kesahihan Seluruh Hasil

Perhatikan urutan penyajian pada laporan yang mendasari bab ini: penyeimbangan data disajikan pada Subbab 4.2.4, sedangkan pembagian data disajikan pada Subbab 4.2.5 setelahnya. Apabila urutan penulisan mencerminkan urutan pelaksanaan, maka SMOTE diterapkan pada keseluruhan data sebelum data dibagi.

Mengapa hal ini bermasalah. SMOTE membangkitkan data sintetis melalui interpolasi antara sebuah data asli dan tetangga terdekatnya, sebagaimana Persamaan 6.15. Apabila proses ini

dijalankan sebelum pembagian, sangat mungkin sebuah data sintetis dibangkitkan dari pasangan data yang kelak terpisah ke bagian pelatihan dan bagian pengujian. Akibatnya, data pelatihan memuat informasi yang berasal dari data uji.
Gejalanya sangat khas. Akurasi pada data uji menjadi jauh lebih tinggi daripada yang wajar, dan selisihnya paling mencolok justru pada kelas minoritas yaitu kelas yang paling banyak memperoleh data sintetis.
Urutan yang benar adalah membagi data terlebih dahulu, lalu menerapkan SMOTE hanya pada bagian pelatihan. Data uji harus mempertahankan distribusi aslinya agar hasil evaluasi mencerminkan kondisi nyata, sebagaimana telah ditegaskan pada Bab 3. Pada validasi silang, SMOTE harus diterapkan ulang di dalam setiap lipatan, dan hal ini paling mudah dijamin melalui penggunaan *Pipeline* dari pustaka imbalanced-learn.
Kotak ini sengaja tidak menyatakan bahwa kekeliruan tersebut pasti terjadi, sebab potongan kode yang tersedia tidak memperlihatkan keseluruhan alur. Yang dinyatakan adalah bahwa hal ini merupakan titik pertama yang harus diperiksa, dan Subbab 6.4.5 akan menunjukkan bukti tidak langsung yang mengarah ke sana.

Skema Pembagian Data

Dua skema pembagian data digunakan secara berdampingan. Skema *hold-out* diterapkan dengan dua rasio, yaitu 75:25 dan 80:20. Skema validasi silang diterapkan dengan empat nilai k, yaitu 3, 5, 7, dan 10.

```
# Hold-out 80 : 20
X_train, X_test, Y_train, Y_test = train_test_split(
   X, Y, test_size=0.2, random_state=42)

# Hold-out 75 : 25
X_train, X_test, Y_train, Y_test = train_test_split(
   X, Y, test_size=0.25, random_state=42)
```

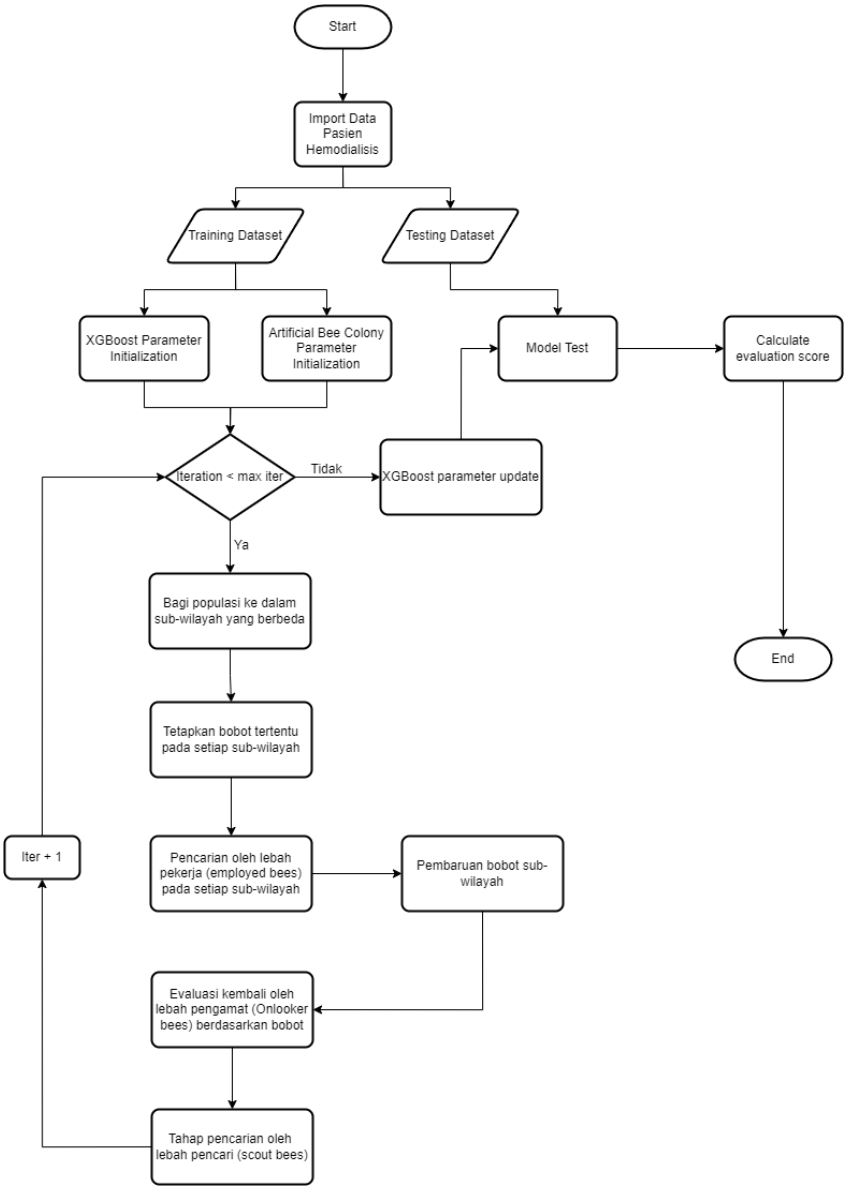
Kode 6.4 Pembagian Data dengan Metode *Hold-Out*

Kombinasi dua skema pembagian, empat nilai k, dua kondisi keseimbangan data, dan lima skenario pengujian menghasilkan total 72 model yang dievaluasi. Rancangan yang menyeluruh semacam ini merupakan kekuatan Metodologi ini, sebab memungkinkan pengamatan pola secara lintas skema dan justru pola lintas skema itulah yang kelak menjadi alat diagnosis pada Subbab 6.4.5.

Satu hal perlu dicatat mengenai pembagian data. Fungsi *train_test_split* pada potongan kode di atas tidak menggunakan parameter *stratify*, sehingga proporsi kelas pada bagian pelatihan dan pengujian tidak dijamin sama dengan proporsi pada data asli. Pada data berukuran kecil dengan kelas minoritas berjumlah 58 sampel, ketiadaan stratifikasi dapat menghasilkan pembagian yang cukup timpang secara kebetulan.

Integrasi *XGBoost* dengan *Artificial Bee Colony*

Proses integrasi kedua algoritma berlangsung sebagai berikut. Setiap sumber makanan dalam koloni merepresentasikan satu vektor berdimensi delapan yang memuat nilai kedelapan *hyperparameter XGBoost*. Untuk mengevaluasi kualitas sebuah sumber makanan, model *XGBoost* dilatih menggunakan kombinasi parameter tersebut, lalu galat klasifikasinya dihitung sesuai Persamaan 6.12 dan digunakan sebagai nilai kebugaran. Siklus lebah pekerja, lebah pengamat, dan lebah penjelajah dijalankan hingga tercapai jumlah iterasi maksimum, dan kombinasi parameter dengan kebugaran terbaik digunakan untuk membentuk model akhir. Alur lengkapnya ditunjukkan pada Gambar 6.2.



Gambar 6.2 Alur Algoritma *XGBoost* dengan Optimalisasi *Artificial Bee Colony*

Konfigurasi parameter ABC yang digunakan meliputi jumlah lebah pekerja, jumlah iterasi maksimum, dan nilai *limit*. Beberapa kombinasi nilai diuji melalui lima skenario pengujian untuk masing-masing skema pembagian data.

Catatan: Biaya Komputasi Optimasi Berbasis Populasi
Satu hal yang jarang diperhitungkan pada awal perancangan adalah bahwa setiap evaluasi kebugaran pada ABC menuntut pelatihan sebuah model *XGBoost* secara utuh.
Dengan 30 lebah dan 30 iterasi, jumlah pelatihan model yang dijalankan mencapai sekitar 900 kali. Apabila skema validasinya berupa validasi silang lima lipatan, angka itu berlipat menjadi 4.500 pelatihan. Inilah sebabnya sebagian percobaan pada bab ini memerlukan waktu lebih dari satu jam, sebagaimana akan terlihat pada Tabel 6.7.
Konsekuensi praktisnya adalah bahwa waktu pelatihan harus diperlakukan sebagai salah satu kriteria pemilihan, bukan sekadar catatan tambahan. Sebuah model yang memerlukan 68 menit untuk mencapai akurasi 96% dan sebuah model lain yang memerlukan 9 menit untuk mencapai 95% bukanlah pilihan yang setara, terutama apabila sistem kelak perlu dilatih ulang secara berkala seiring bertambahnya data pasien.

Rancangan Skenario Pengujian
Rancangan pengujian pada bab ini disusun sebagai percobaan faktorial atas tiga faktor. Faktor pertama berupa skema pembagian data dengan enam tingkat, yaitu dua rasio *hold-out* dan empat nilai k pada validasi silang. Faktor kedua berupa kondisi keseimbangan data dengan dua tingkat. Faktor ketiga berupa konfigurasi parameter ABC dengan lima skenario. Komposisinya dirangkum pada Tabel 6.5.

Tabel 6.5 Komposisi Skenario Pengujian

Faktor	**Tingkat**	**Jumlah**
Skema pembagian data	*Hold-out* 75:25 dan 80:20; K = 3, 5, 7, 10	6
Kondisi keseimbangan	Data asli dan data hasil SMOTE	2
Konfigurasi ABC	Lima kombinasi jumlah lebah, iterasi, dan limit	5
Total model *XGBoost*-ABC	6 × 2 × 5 (sebagian kombinasi tidak dijalankan)	± 60
Model pembanding tanpa optimalisasi	6 × 2	12
Total keseluruhan	—	**72**

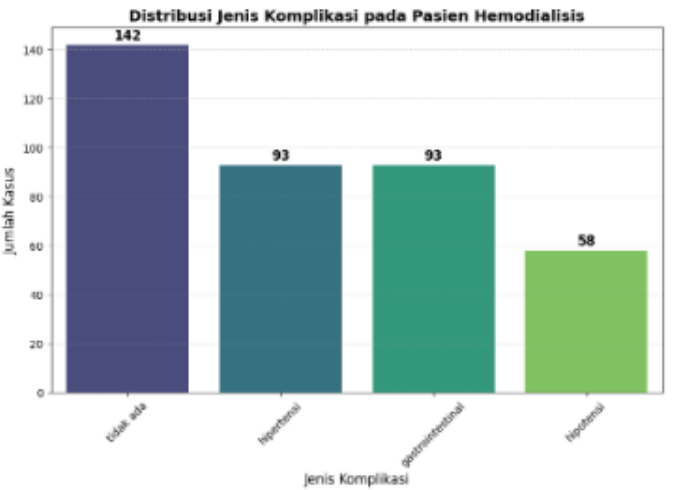
Rancangan faktorial semacam ini merupakan kekuatan utama Metodologi ini dan patut dicontoh. Keunggulannya bukan terletak pada banyaknya model yang diuji, melainkan pada kemungkinan mengamati pola secara lintas faktor. Muncul pada satu skenario dapat disebabkan oleh kebetulan; muncul secara konsisten pada seluruh tingkat suatu faktor jauh lebih layak dipercaya. Kemampuan membedakan keduanya inilah yang akan dimanfaatkan secara langsung pada Subbab 6.4.5.

Perlu ditambahkan satu catatan mengenai apa yang sesungguhnya dibandingkan pada faktor ketiga. Kelima konfigurasi ABC bukan merupakan lima model yang berbeda, melainkan lima kali pencarian atas ruang parameter yang sama dengan anggaran pencarian yang berbeda-beda. Karena ABC bersifat stokastik, kelima hasilnya juga mencerminkan keragaman akibat keacakan, bukan semata pengaruh konfigurasi. Menafsirkan selisih di antara kelimanya sebagai pengaruh konfigurasi karena itu perlu dilakukan dengan hati-hati, kecuali apabila setiap konfigurasi dijalankan beberapa kali dengan nilai benih acak yang berbeda.

➢ Hasil dan Ujicoba

Eksplorasi Data dan Hubungan Antarvariabel

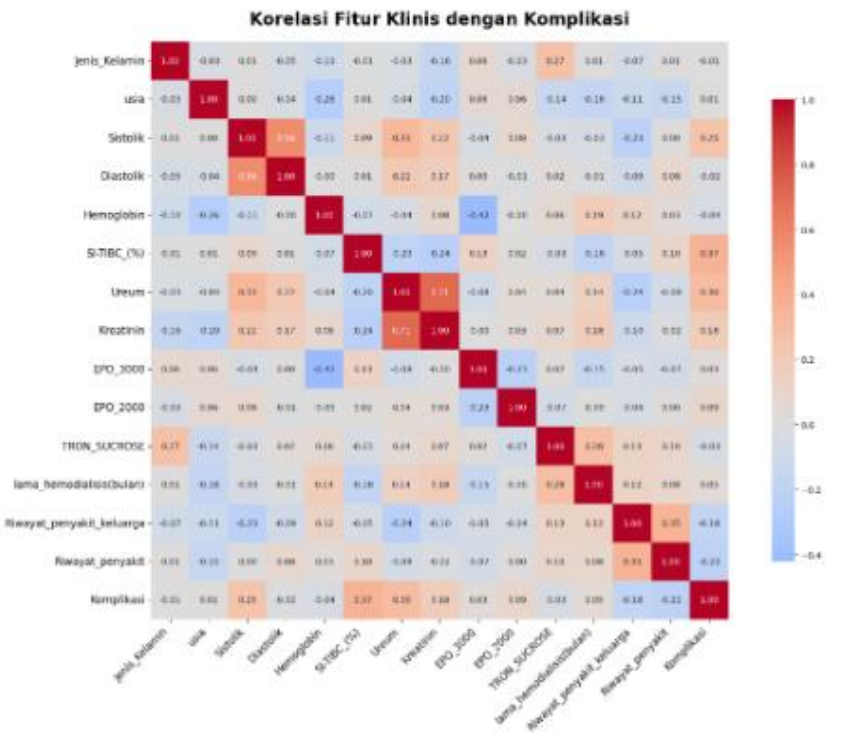
Distribusi jenis komplikasi pada data awal ditunjukkan pada Gambar 6.3.



Gambar 6.3 Distribusi Jenis Komplikasi pada Data Awal

Kelas tanpa komplikasi mendominasi dengan 142 sampel atau sekitar 36,7% dari keseluruhan data, disusul hipertensi dan gastrointestinal yang masing-masing berjumlah 93 sampel atau 24,1%, dan terakhir hipotensi dengan 58 sampel atau 15,0%. Ketidakseimbangan ini tergolong sedang perbandingan antara kelas terbesar dan terkecil sekitar 2,4 berbanding 1 dan jauh lebih ringan daripada yang dijumpai pada Bab 3.

Perlu dicatat bahwa kelas dengan sampel paling sedikit justru merupakan kelas hipotensi, yaitu komplikasi yang paling berbahaya sebagaimana dibahas pada Subbab 6.2.1. Kondisi ini merupakan kombinasi yang tidak menguntungkan: kelas yang paling penting dikenali justru merupakan kelas yang paling sedikit datanya. Hubungan antarvariabel klinis dan komplikasi ditunjukkan melalui matriks korelasi pada Gambar 6.4.



Gambar 6.4 Matriks Korelasi Fitur Klinis dengan Komplikasi

Kolom terakhir pada matriks tersebut memuat korelasi setiap variabel terhadap keberadaan komplikasi, dan angkanya dirangkum pada Tabel 6.4.

Tabel 6.4 Korelasi Variabel Klinis terhadap Keberadaan Komplikasi

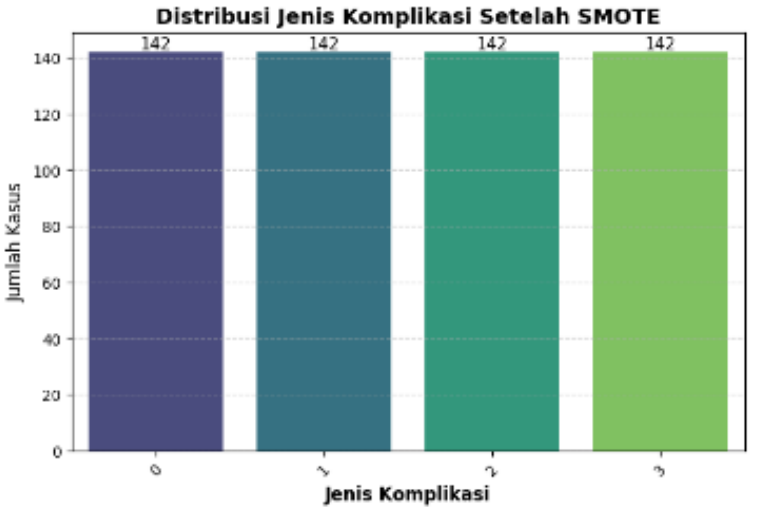
Variabel	**Korelasi**	**Tafsiran Klinis**
SI-TIBC	+0,37	Gangguan status zat besi berkaitan dengan komplikasi
Ureum	+0,30	Beban uremik tinggi meningkatkan risiko
Sistolik	+0,25	Tekanan darah tinggi sebagai penanda risiko
Kreatinin	+0,18	Fungsi ginjal residual yang lebih buruk
Riwayat keluarga	−0,18	Arah negatif, perlu penelusuran lebih lanjut
Riwayat penyakit	−0,22	Arah negatif, perlu penelusuran lebih lanjut

Dua hal patut diperhatikan pada matriks korelasi tersebut. Yang pertama, korelasi antara ureum dan kreatinin mencapai 0,71 angka yang sangat tinggi dan sepenuhnya dapat dijelaskan secara fisiologis, sebab keduanya merupakan zat sisa metabolisme yang sama-sama menumpuk ketika fungsi ginjal menurun. Multikolinearitas semacam ini tidak mengganggu model berbasis pohon sebagaimana ia mengganggu model linier, namun berdampak pada penafsiran kepentingan fitur: ketika dua fitur membawa informasi yang hampir sama, pohon akan memilih salah satunya secara sembarang sehingga kepentingan yang seharusnya terbagi tampak terpusat pada satu fitur saja.

Yang kedua, korelasi antara riwayat penyakit dan komplikasi bernilai negatif sebesar −0,22, yang bertentangan dengan intuisi klinis bahwa penyakit penyerta seharusnya meningkatkan risiko. Penjelasan yang paling mungkin bersifat teknis: variabel ini merupakan hasil *label encoding* atas kategori nominal, sehingga korelasi Pearson yang dihitung darinya tidak memiliki makna. Nilai numerik yang diberikan kepada setiap kategori bersifat sembarang, sehingga arah korelasinya pun sembarang. Hal ini merupakan

pengingat bahwa matriks korelasi hanya sahih dibaca pada variabel numerik dan variabel biner, bukan pada kategori nominal berkode.

Distribusi kelas setelah penerapan SMOTE ditunjukkan pada Gambar 6.5.



Gambar 6.5 Distribusi Kelas Setelah Penerapan SMOTE

Seluruh kelas kini memuat 142 sampel dengan total 568 data. Kelas hipotensi mengalami penambahan terbesar, yaitu 84 sampel sintetis atau sekitar 145% dari jumlah aslinya. Angka ini penting untuk diingat: lebih dari separuh sampel pada kelas hipotensi setelah penyeimbangan merupakan data buatan, bukan data pasien yang sesungguhnya.

Hasil *XGBoost* Tanpa Optimalisasi

Sebagai pembanding dasar, *XGBoost* dijalankan terlebih dahulu menggunakan parameter bawaan tanpa penalaan. Hasilnya disajikan pada Tabel 6.6.

Tabel 6.6 Akurasi *XGBoost* Tanpa Optimalisasi

Skema Pembagian	**Data Tidak Seimbang**	**Data Seimbang**	**Selisih**
Hold-out 80 : 20	91%	94%	+3
Hold-out 75 : 25	87%	93%	+6
K-Fold, K = 3	84%	84%	0

K-Fold, K = 5	84%	84%	0
K-Fold, K = 7	83%	83%	0
K-Fold, K = 10	79%	79%	0

Akurasi tertinggi sebesar 94% diperoleh pada skema *hold-out* 80:20 dengan data seimbang. Terdapat pula pola yang konsisten pada validasi silang, yaitu penurunan akurasi seiring bertambahnya jumlah lipatan, dari 84% pada k sama dengan 3 menjadi 79% pada k sama dengan 10. Pola ini wajar pada data berukuran kecil: semakin banyak lipatan, semakin sedikit data uji pada setiap lipatan sehingga taksirannya semakin bergejolak, sekaligus semakin banyak data pelatihan sehingga model semakin kompleks dan berisiko *overfitting*.

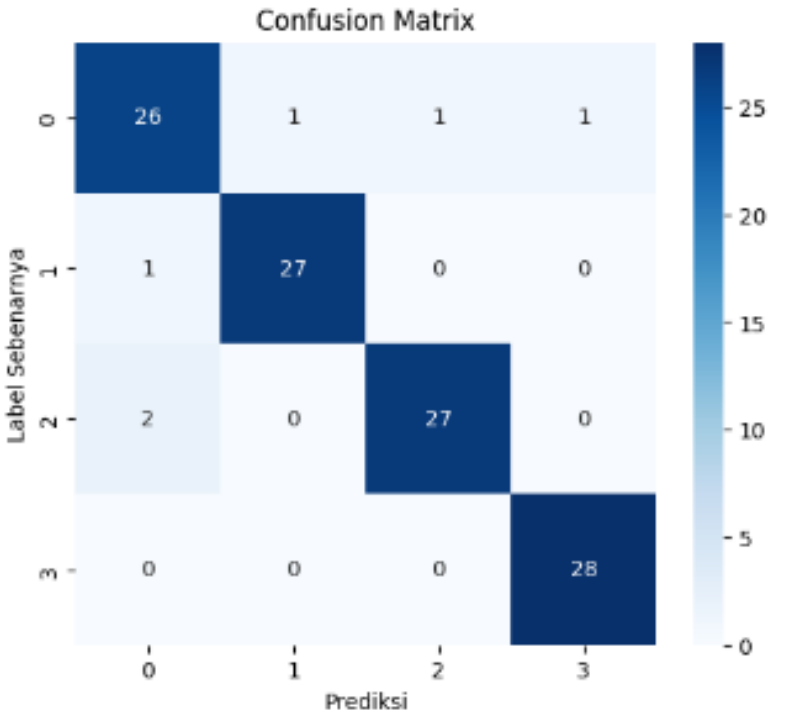
Kolom selisih pada Tabel 6.6 merupakan bagian yang paling patut diperhatikan, dan pengamatannya akan diteruskan pada Subbab 6.4.5. Pada skema *hold-out*, penyeimbangan data menaikkan akurasi sebesar tiga sampai enam poin persentase. Pada validasi silang, penyeimbangan tidak memberikan pengaruh sama sekali angkanya identik hingga digit terakhir pada keempat nilai k. Ketiadaan pengaruh yang begitu sempurna merupakan hal yang tidak lazim dan patut dipertanyakan.

Sebelum melangkah lebih jauh, satu pembanding perlu ditetapkan terlebih dahulu sebagaimana telah diperkenalkan pada Bab 3 melalui Persamaan 3.10, yaitu akurasi yang dicapai oleh sebuah model tanpa kecerdasan apa pun yang selalu menebak kelas mayoritas. Pada data asli, kelas tanpa komplikasi berjumlah 142 dari 387 sampel, sehingga *baseline* tersebut bernilai 36,7%. Pada data yang telah diseimbangkan, keempat kelas berjumlah sama sehingga *baseline*-nya turun menjadi 25%.

Perbandingan terhadap *baseline* ini menempatkan hasil pada bab ini dalam perspektif yang jauh lebih menguntungkan daripada yang dijumpai pada Bab 3. Akurasi 87% pada validasi silang berada sekitar lima puluh poin persentase di atas *baseline* kelas mayoritas. Selisih sebesar itu tidak mungkin diperoleh secara kebetulan dan menunjukkan bahwa model benar-benar mempelajari pola yang bermakna dari data rekam medis. Dengan kata lain, seluruh kritik yang diuraikan pada subbab-subbab berikutnya menyangkut ketelitian pengukuran dan pilihan metodologis, bukan menyangkut apakah pendekatan ini berhasil pendekatan ini jelas berhasil.

Perbandingan tersebut juga bermanfaat sebagai pembanding lintas bab. Pada Bab 3, akurasi 96,9% hanya unggul tiga poin dari *baseline* sebesar 93,8% sehingga sumbangan modelnya sangat kecil meskipun angkanya tinggi. Pada bab ini, akurasi 87% unggul lima puluh poin sehingga sumbangan modelnya sangat besar meskipun angkanya lebih rendah. Kedua kasus tersebut memperlihatkan bahwa angka akurasi tidak dapat dibandingkan antarpermasalahan tanpa mengetahui *baseline* masing-masing.

Confusion matrix model terbaik tanpa optimalisasi ditunjukkan pada Gambar 6.6.



Gambar 6.6 *Confusion Matrix XGBoost* Tanpa Optimalisasi (*Hold-out* 80:20, Data Seimbang)

Dari 29 data kelas tanpa komplikasi, sebanyak 26 dikenali dengan benar. Kelas hipertensi memperoleh 27 prediksi benar dari 28 data, kelas hipotensi 27 dari 29, dan kelas gastrointestinal seluruhnya benar sebanyak 28 data. Perhatikan bahwa jumlah data

uji pada setiap kelas nyaris sama, yaitu sekitar 28 sampel, hal ini menegaskan bahwa data uji yang digunakan berasal dari data yang telah diseimbangkan, bukan dari distribusi asli.

Hasil *XGBoost* dengan Optimalisasi ABC

Optimalisasi menggunakan ABC dijalankan pada seluruh skema pembagian data dengan lima skenario konfigurasi parameter ABC untuk masing-masing skema. Hasil pada skema *hold-out* disajikan pada Tabel 6.7, disertai waktu pelatihan yang dibutuhkan.

Tabel 6.7 Hasil *XGBoost*-ABC pada Skema *Hold-Out*

Uji	Akurasi 75:25 Tak Seimbang	Waktu (menit)	Akurasi 75:25 Seimbang	Waktu (menit)	Akurasi 80:20 Seimbang	Waktu (menit)
1	90%	7	95%	9	93%	10
2	89%	23	93%	28	93%	39
3	**89%**	**66**	**96%**	**68**	**94%**	**104**
4	89%	11	92%	17	92%	37
5	91%	18	93%	19	91%	71

Akurasi tertinggi sebesar 96% diperoleh pada skema 75:25 dengan data seimbang melalui skenario uji 3, dengan waktu pelatihan mencapai 68 menit. Perlu dicatat bahwa skenario uji 1 pada skema yang sama mencapai 95% hanya dalam 9 menit selisih satu poin persentase yang dibayar dengan waktu pelatihan lebih dari tujuh kali lipat. Hasil pada skema validasi silang disajikan pada Tabel 6.11.

Tabel 6.8 Hasil *XGBoost*-ABC pada Skema Validasi Silang

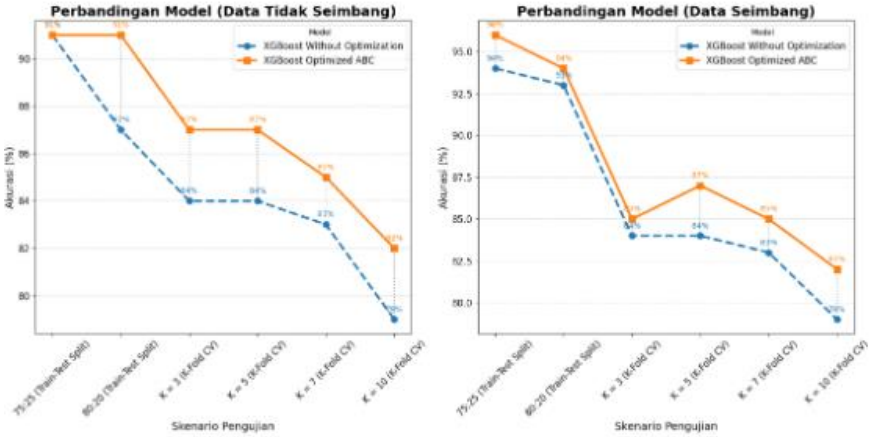
Nilai K	Akurasi Terbaik Tak Seimbang	Akurasi Terbaik Seimbang	Waktu Tercepat
K = 3	87%	85%	8 menit
K = 5	**87%**	**87%**	**8 menit**
K = 7	85%	85%	8 menit
K = 10	82%	82%	11 menit

Akurasi tertinggi pada validasi silang sebesar 87% diperoleh pada k sama dengan 5, baik pada data seimbang maupun tidak seimbang. Perbandingan menyeluruh antara *XGBoost* tanpa optimalisasi dan *XGBoost*-ABC disajikan pada Tabel 6.11.

Tabel 6.9 Perbandingan *XGBoost* Tanpa Optimalisasi dan *XGBoost*-ABC

Skema	**Tak Seimbang Tanpa ABC**	**Tak Seimbang Dengan ABC**	**Seimbang Tanpa ABC**	**Seimbang Dengan ABC**
Hold-out 75:25	91%	91%	94%	96%
Hold-out 80:20	87%	91%	93%	94%
K = 3	84%	87%	84%	85%
K = 5	84%	87%	84%	87%
K = 7	83%	85%	83%	85%
K = 10	79%	82%	79%	82%

Optimalisasi ABC memberikan peningkatan pada hampir seluruh skenario, dengan besaran antara satu sampai empat poin persentase. Peningkatan paling konsisten justru terlihat pada data tidak seimbang dan pada skema validasi silang dua kondisi yang paling menantang. Visualisasi perbandingannya ditunjukkan pada Gambar 6.7.



Gambar 6.7 Perbandingan Akurasi *XGBoost* Tanpa Optimalisasi dan *XGBoost*-ABC

Pada kedua panel, garis oranye yang mewakili model teroptimasi konsisten berada di atas atau sejajar dengan garis biru. Namun yang lebih menarik untuk diamati adalah bentuk kedua kurva, bukan jarak di antaranya. Keduanya menurun tajam dari kiri ke kanan, yaitu dari skema *hold-out* menuju validasi silang dengan jumlah lipatan yang bertambah. Penurunan sebesar sembilan sampai empat belas poin persentase antara ujung kiri dan ujung kanan jauh melampaui pengaruh optimalisasi itu sendiri.

Peningkatan yang Diberikan Optimalisasi

Sebelum menelusuri persoalan yang lebih besar, perlu dinilai terlebih dahulu seberapa berarti sumbangan optimalisasi ABC. Pada validasi silang dengan k sama dengan 5 skema yang paling layak dipercaya pada data berukuran ini akurasi meningkat dari 84% menjadi 87%, yaitu sebesar tiga poin persentase.

Dengan 387 data, tiga poin persentase setara dengan sekitar dua belas sampel yang berpindah dari salah menjadi benar. Angka ini tidak besar namun juga tidak dapat diabaikan, terutama karena peningkatan tersebut konsisten muncul pada seluruh nilai k dan pada kedua kondisi keseimbangan data. Konsistensi lintas skenario merupakan bukti yang jauh lebih kuat daripada satu angka tertinggi pada satu skenario tertentu.

Meskipun demikian, biaya untuk memperoleh tiga poin tersebut perlu dinyatakan secara jujur. Waktu pelatihan meningkat dari hitungan detik menjadi puluhan menit, dan pada skenario tertentu melampaui satu jam. Perbandingan antara akurasi yang dicapai dan waktu yang dibutuhkan pada skema *hold-out* 75:25 dengan data seimbang disajikan pada Tabel 6.10.

Tabel 6.10 Perbandingan Akurasi terhadap Waktu Pelatihan (*Hold-out* 75:25, Data Seimbang)

Skenario	**Akurasi**	**Waktu (menit)**	**Poin Akurasi per Menit**
Uji 4	92%	17	5,4
Uji 2	93%	28	3,3
Uji 5	93%	19	4,9
Uji 1	**95%**	**9**	**10,6**
Uji 3	96%	68	1,4

Kolom terakhir pada tabel tersebut menempatkan skenario uji 1 pada posisi yang sangat menonjol. Skenario tersebut mencapai akurasi 95% hanya dalam 9 menit, sedangkan skenario uji 3 memerlukan 68 menit lebih dari tujuh kali lipat untuk menambahkan satu poin persentase saja. Pada data uji berukuran sekitar 142 sampel, satu poin persentase setara dengan sekitar satu setengah sampel. Dengan kata lain, tambahan waktu selama hampir satu jam dibayar dengan perbaikan pada satu sampai dua pasien, dan perbaikan sebesar itu tidak dapat dibedakan dari keacakan.

Pengamatan ini mengarah pada satu prinsip yang berlaku umum pada penalaan *hyperparameter*. Hubungan antara anggaran pencarian dan kualitas solusi bersifat menurun secara tajam: sebagian besar perbaikan diperoleh pada tahap awal pencarian, sedangkan sisa waktu dihabiskan untuk mengejar perbaikan yang semakin kecil. Kurva kebugaran pada Gambar 6.9 memperlihatkan hal yang sama dalam skala satu percobaan, yaitu konvergensi tercapai pada iterasi ke-15 dari 30 iterasi yang dijalankan.

Pada konteks Metodologi, biaya waktu ini dapat diterima karena optimalisasi hanya dilakukan sekali. Pada konteks penerapan yang menuntut pelatihan ulang berkala seiring bertambahnya data pasien, pertimbangannya menjadi berbeda. Konfigurasi yang mencapai 95% dalam 9 menit jauh lebih sesuai untuk dijalankan setiap bulan daripada konfigurasi yang menuntut lebih dari satu jam. Penerapan kriteria penghentian dini pada ABC, sebagaimana disarankan pada Subbab 6.4.8, akan memangkas selisih ini secara berarti tanpa mengorbankan kualitas solusi.

Selisih Sembilan Poin: Sebuah Penelusuran

Bagian ini menelusuri pertanyaan yang telah beberapa kali disinggung sebelumnya. Model terbaik pada skema *hold-out* mencapai akurasi 96%, sedangkan model terbaik pada validasi silang hanya mencapai 87%. Selisih sembilan poin persentase ini terlalu besar untuk diabaikan sebagai keacakan biasa, dan menelusuri penyebabnya jauh lebih bermanfaat daripada sekadar melaporkan angka tertingginya.

Kemungkinan pertama: keacakan pembagian. Dengan 387 data, skema *hold-out* 75:25 hanya menyisakan sekitar 97 data uji, atau 142 apabila data telah diseimbangkan. Pada jumlah sekecil itu,

satu pembagian yang kebetulan menguntungkan dapat menaikkan akurasi beberapa poin. Namun penjelasan ini memiliki batas: keacakan dapat menjelaskan selisih dua sampai tiga poin, tidak sembilan poin. Terlebih lagi, seluruh lima skenario *hold-out* pada data seimbang menghasilkan akurasi antara 92% sampai 96%, sedangkan seluruh skenario validasi silang berkisar 82% sampai 87%. Kedua rentang itu bahkan tidak bersinggungan.

Kemungkinan kedua: perbedaan jumlah data pelatihan. Pada validasi silang dengan k sama dengan 5, setiap model dilatih menggunakan 80% data, yaitu proporsi yang sebanding dengan *hold-out* 80:20. Perbedaan jumlah data pelatihan karena itu tidak dapat menjelaskan selisihnya.

Kemungkinan ketiga: kebocoran data melalui SMOTE. Penjelasan ini paling sesuai dengan seluruh pola yang teramati, dan alasannya diuraikan pada kotak berikut.

Analisis: Mengapa Pola Angka Mengarah pada Kebocoran SMOTE

Terdapat tiga pengamatan yang secara bersama-sama mengarah pada satu penjelasan.

Pengamatan pertama. Pada Tabel 6.6, penyeimbangan data menaikkan akurasi *hold-out* sebesar 3 sampai 6 poin persentase, tetapi sama sekali tidak menaikkan akurasi validasi silang angkanya identik hingga digit terakhir pada keempat nilai k. Apabila penyeimbangan benar-benar membantu model belajar, manfaatnya semestinya terlihat pada kedua skema. Apabila manfaatnya hanya muncul pada satu skema, maka yang terjadi kemungkinan bukan pembelajaran yang lebih baik melainkan pengukuran yang tidak sahih.

Pengamatan kedua. Confusion matrix pada Gambar 6.6 memperlihatkan jumlah data uji yang nyaris sama pada keempat kelas, yaitu sekitar 28 sampel. Ini menandakan data uji berasal dari data yang telah diseimbangkan, sehingga memuat sampel sintetis. Sebaliknya, *confusion matrix* model terbaik pada validasi silang di Gambar 6.8 memperlihatkan dukungan sebesar 28, 15, 16, dan 18 yaitu proporsi yang menyerupai distribusi asli.

Dengan kata lain, kedua skema tidak diuji pada jenis data yang sama.

Pengamatan ketiga. Menguji model pada data sintetis pada dasarnya menguji kemampuan model menebak titik-titik interpolasi dari data yang sebagian telah dipelajarinya. Tugas ini jauh lebih mudah daripada mengenali pasien baru yang sesungguhnya.

Kesimpulan yang wajar. Angka 96% kemungkinan besar merupakan taksiran yang terlalu optimistis, sedangkan angka 87% dari validasi silang lebih mendekati performa sebenarnya. Hal ini tidak mengurangi nilai Metodologi; justru sebaliknya, keberadaan kedua skema secara berdampingan itulah yang memungkinkan persoalan ini terdeteksi. Apabila hanya skema *hold-out* yang dijalankan, angka 96% akan diterima tanpa pertanyaan.

Pemeriksaan yang perlu dilakukan. Jalankan ulang skema *hold-out* dengan urutan yang benar bagi data terlebih dahulu, terapkan SMOTE hanya pada bagian pelatihan, lalu uji pada data asli. Apabila akurasinya turun mendekati 87%, dugaan ini terkonfirmasi. Apabila tetap berada di kisaran 95%, berarti penjelasannya terletak di tempat lain dan penelusuran perlu dilanjutkan.

Terlepas dari hasil pemeriksaan tersebut, satu keputusan pada Metodologi ini patut memperoleh apresiasi khusus dan akan dibahas pada subbab berikutnya.

Pemilihan Model Terbaik

Model yang dipilih untuk diterapkan pada sistem bukanlah model dengan akurasi tertinggi. Model dengan akurasi 96% pada skema *hold-out* tidak dipilih; yang dipilih adalah model *XGBoost*-ABC pada validasi silang dengan k sama dengan 5 dan data seimbang, dengan akurasi 87%.

Apresiasi: Memilih Angka yang Lebih Rendah karena Lebih Layak Dipercaya

Keputusan ini pantas mendapat perhatian tersendiri karena berlawanan dengan kecenderungan yang lazim. Ketika sebuah Metodologi menghasilkan angka 96%, godaan untuk menampilkannya sebagai capaian utama sangatlah besar.

Alasan yang dikemukakan pada laporan adalah keseimbangan performa di seluruh kelas, bukan sekadar akurasi keseluruhan dan alasan tersebut sudah tepat. Penelusuran pada Subbab 6.4.5 memberikan alasan tambahan yang bahkan lebih kuat: taksiran dari validasi silang lebih layak dipercaya karena menggunakan seluruh data secara bergiliran sebagai data uji dan tidak bergantung pada satu pembagian yang kebetulan.

Prinsip yang berlaku umum: *pilih model berdasarkan skema evaluasi yang paling layak dipercaya, bukan berdasarkan angka yang paling tinggi.* Kedua kriteria tersebut sering kali menunjuk pada model yang berbeda, dan ketika hal itu terjadi, angka yang lebih tinggi hampir selalu merupakan yang kurang dapat dipercaya.

Parameter terbaik yang ditemukan ABC untuk model terpilih disajikan pada Tabel 6.11.

Tabel 6.11 Parameter Terbaik Hasil Optimalisasi ABC

Parameter	**Nilai**	**Catatan**
n_estimators	424	Jumlah pohon tergolong banyak
learning_rate (η)	0,9292	Sangat tinggi untuk *XGBoost*
reg_lambda (λ)	16,4798	Regularisasi L2 kuat
max_depth	9	Pohon tergolong dalam
min_child_weight	0	Membolehkan daun bersampel sangat sedikit
gamma (γ)	0,4833	Ambang manfaat pemisahan sedang
subsample	0,9782	Hampir seluruh sampel digunakan
colsample_bytree	0,4638	Sekitar separuh fitur per pohon

Perhatian: Kombinasi Parameter yang Perlu *Dice*rmati

Nilai *learning_rate* sebesar 0,929 merupakan angka yang sangat tidak lazim untuk *XGBoost*. Rentang yang lazim digunakan berkisar antara 0,01 sampai 0,3, dan nilai mendekati satu berarti setiap pohon baru diberi bobot penuh sehingga pada dasarnya menghapus efek peredaman yang menjadi inti gagasan *boosting*. Nilai tersebut menjadi lebih patut *dice*rmati apabila dibaca bersama tiga parameter lain. Jumlah pohon sebanyak 424 tergolong banyak, kedalaman maksimum sebesar 9 tergolong dalam, dan *min_child_weight* sebesar 0 membolehkan terbentuknya daun yang hanya memuat satu sampel. Keempatnya secara bersama-sama menghasilkan model dengan kapasitas sangat besar kombinasi yang secara teori sangat rentan terhadap *overfitting*.

Satu-satunya penyeimbang adalah *reg_lambda* sebesar 16,48 yang tergolong kuat. Kemungkinan besar ABC menemukan bahwa regularisasi L2 yang kuat cukup untuk mengimbangi kapasitas berlebih tersebut.

Meskipun demikian, kombinasi semacam ini merupakan pertanda yang perlu ditelusuri. Algoritma optimasi menemukan apa pun yang memaksimalkan fungsi kebugarannya; apabila kebugaran dihitung pada data yang memuat kebocoran, maka model berkapasitas besar yang mampu menghafal justru akan memperoleh nilai kebugaran terbaik. Dengan kata lain, parameter pada Tabel 6.11 dapat dibaca sebagai bukti tidak langsung yang mendukung dugaan pada Subbab 6.4.5.

Pemeriksaan yang disarankan: batasi rentang pencarian *learning_rate* pada 0,01 sampai 0,3 dan tetapkan *min_child_weight* minimal 1, lalu jalankan ulang optimalisasi. Apabila akurasinya tidak turun berarti, model yang dihasilkan menjadi jauh lebih dapat dipertanggungjawabkan.

Evaluasi per Kelas pada Model Terpilih

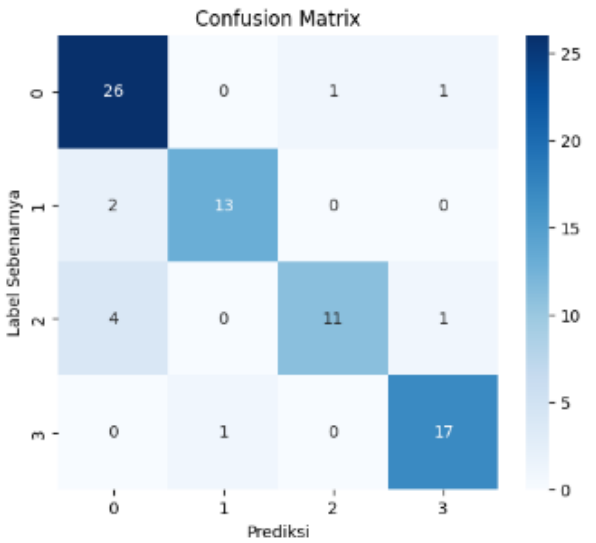
Metrik evaluasi rinci untuk model terpilih disajikan pada Tabel 6.12.

Tabel 6.12 Metrik Evaluasi Model Terpilih *XGBoost*-ABC (K = 5)

Kelas	***Precision***	***Recall***	**F1-Score**	***Support***
0 — Tanpa komplikasi	0,81	0,93	0,87	28
1 — Hipertensi	0,93	0,87	0,90	15
2 — Hipotensi	0,92	0,69	0,79	16
3 — Gastrointestinal	0,89	0,94	0,92	18
Rerata makro	**0,89**	**0,86**	**0,87**	**77**
Rerata terbobot	**0,88**	**0,87**	**0,87**	**77**

Akurasi keseluruhan sebesar 87% dengan rerata makro *F1-score* sebesar 0,87 menunjukkan performa yang cukup baik dan cukup merata. Namun sebaran per kelas mengungkap sesuatu yang tidak tampak pada angka rerata: nilai *recall* pada kelas hipotensi hanya sebesar 0,69, yaitu yang terendah di antara keempat kelas dan terpaut jauh dari ketiga kelas lainnya.

Confusion matrix model terpilih ditunjukkan pada Gambar 6.8, dan angka di dalamnya menjelaskan sepenuhnya asal-usul nilai tersebut.



Gambar 6.8 *Confusion Matrix* Model Terpilih *XGBoost*-ABC (K = 5)

Dari 16 kasus hipotensi yang sebenarnya, hanya 11 yang dikenali dengan benar. Sebanyak 4 kasus diprediksi sebagai tanpa komplikasi dan 1 kasus diprediksi sebagai gastrointestinal. Pola kekeliruan pada seluruh kelas dirangkum pada Tabel 6.13.

Tabel 6.13 Pola Kekeliruan Model Terpilih

Kelas Sebenarnya	**Benar**	**Salah ke Tanpa Komplikasi**	**Salah ke Kelas Lain**	***Recall***
Tanpa komplikasi	26	—	2	0,93
Hipertensi	13	2	0	0,87
Hipotensi	11	4	1	0,69
Gastrointestinal	17	0	1	0,94

Perhatian: Kelas Paling Berbahaya Justru yang Paling Sering Terlewat

Angka pada Tabel 6.13 menyampaikan pesan yang tegas. Sebanyak enam pasien yang sesungguhnya akan mengalami komplikasi diprediksi sebagai tanpa komplikasi dua kasus hipertensi dan empat kasus hipotensi.

Keempat kasus hipotensi tersebut merupakan jenis kekeliruan yang paling merugikan pada keseluruhan sistem. Sebagaimana dibahas pada Subbab 6.2.1, hipotensi intradialitik dapat berkembang dalam hitungan menit dan berujung pada kehilangan kesadaran ketika pasien sedang terhubung dengan mesin. Sebuah sistem yang menyatakan pasien tersebut aman justru menghilangkan kewaspadaan yang paling dibutuhkan.

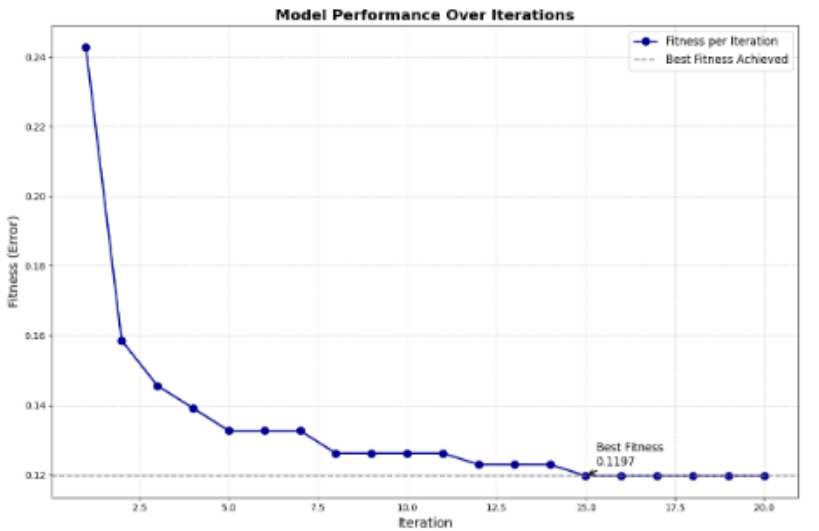
Perhatikan pula bahwa arah kekeliruannya sangat tidak simetris. Model mengirim 6 kasus berkomplikasi ke kelas tanpa komplikasi, sedangkan hanya 2 kasus tanpa komplikasi yang dikirim ke kelas berkomplikasi. Ketidaksimetrisan ini menunjukkan kecenderungan sistematis untuk meremehkan risiko kecenderungan yang tepat berlawanan dengan yang dibutuhkan pada sistem penapisan klinis.

Penyebabnya dapat ditelusuri ke dua hal. Pertama, kelas hipotensi memang memiliki data asli paling sedikit, yaitu 58 sampel. Kedua, sebagian besar sampel hipotensi setelah

penyeimbangan merupakan data sintetis, sehingga model mempelajari titik-titik interpolasi dan bukan variasi pasien yang sesungguhnya.
Perbaikan yang disarankan: terapkan bobot kelas yang lebih besar pada kelas hipotensi saat pelatihan, atau ubah fungsi kebugaran ABC dari akurasi keseluruhan menjadi *recall* kelas hipotensi maupun rerata makro *F1-score*. Perubahan pada fungsi kebugaran merupakan perbaikan yang paling langsung, sebab algoritma optimasi hanya akan mengejar apa pun yang diminta kepadanya.

Konvergensi Optimalisasi dan Kemampuan Pemisahan

Perkembangan nilai kebugaran selama proses optimalisasi ABC pada model terpilih ditunjukkan pada Gambar 6.9.



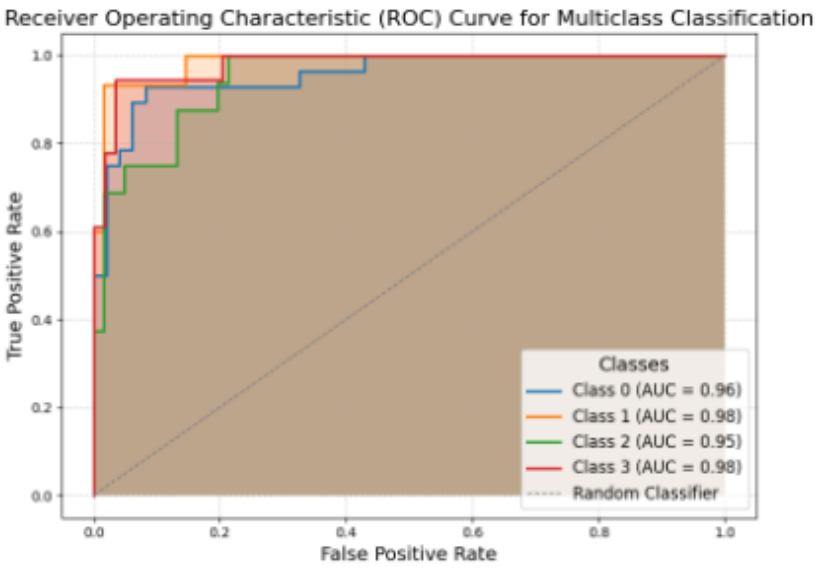
Gambar 6.9 Iterasi Nilai Fitness pada Model Terpilih

Nilai kebugaran menurun tajam pada iterasi awal dari sekitar 0,24 menjadi 0,16, kemudian melandai secara bertahap hingga mencapai nilai terbaik sebesar 0,1197 pada iterasi ke-15. Setelah titik tersebut tidak terjadi perbaikan berarti hingga iterasi berakhir.

Bentuk kurva ini merupakan pola yang khas bagi algoritma optimasi berbasis populasi dan dapat dibaca sebagai tiga fase. Fase penurunan tajam pada iterasi awal mencerminkan tahap eksplorasi, ketika populasi masih tersebar luas sehingga selisih pada Persamaan 6.10 bernilai besar dan langkah perubahannya pun besar. Fase pelandaian mencerminkan tahap eksploitasi, ketika populasi mulai memusat dan pencarian menjadi halus. Fase datar setelah iterasi ke-15 menandakan konvergensi telah tercapai.

Fase terakhir tersebut memiliki makna praktis. Karena tidak terjadi perbaikan setelah iterasi ke-15, separuh anggaran komputasi pada percobaan dengan 30 iterasi sesungguhnya terbuang. Penerapan kriteria penghentian dini yang memantau stagnasi nilai kebugaran akan memangkas waktu pelatihan secara berarti tanpa mengorbankan kualitas solusi. Perlu ditambahkan bahwa nilai kebugaran sebesar 0,1197 setara dengan akurasi sekitar 88%, yang konsisten dengan akurasi pengujian sebesar 87% kesesuaian ini menandakan tidak terdapat penyimpangan besar antara data tempat optimasi dilakukan dan data pengujian pada skema validasi silang.

Kemampuan pemisahan model dinilai melalui kurva ROC satu-lawan-sisanya sebagaimana ditunjukkan pada Gambar 6.10.



Gambar 6.10 Kurva ROC Model Terpilih untuk Klasifikasi Multikelas

Nilai AUC yang diperoleh sangat tinggi pada seluruh kelas, yaitu 0,96 untuk tanpa komplikasi, 0,98 untuk hipertensi, 0,95 untuk hipotensi, dan 0,98 untuk gastrointestinal. Angka-angka ini menandakan bahwa model sesungguhnya mampu memisahkan keempat kelas dengan sangat baik.

Catatan: AUC Tinggi namun *Recall* Rendah, Bagaimana Keduanya Bisa Bersamaan

Terdapat ketidakserasian yang tampak mencolok antara Gambar 6.10 dan Tabel 6.10. Kelas hipotensi memperoleh AUC sebesar 0,95 yang tergolong sangat baik, namun *recall*-nya hanya 0,69. Keduanya sama sekali tidak bertentangan, dan penjelasannya justru membuka peluang perbaikan yang murah. AUC dihitung dari nilai probabilitas dan menilai kemampuan model mengurutkan sampel pada seluruh nilai ambang yang mungkin. Nilai *recall*, sebaliknya, dihitung pada satu ambang tertentu saja, yaitu ambang bawaan yang memilih kelas dengan probabilitas tertinggi.

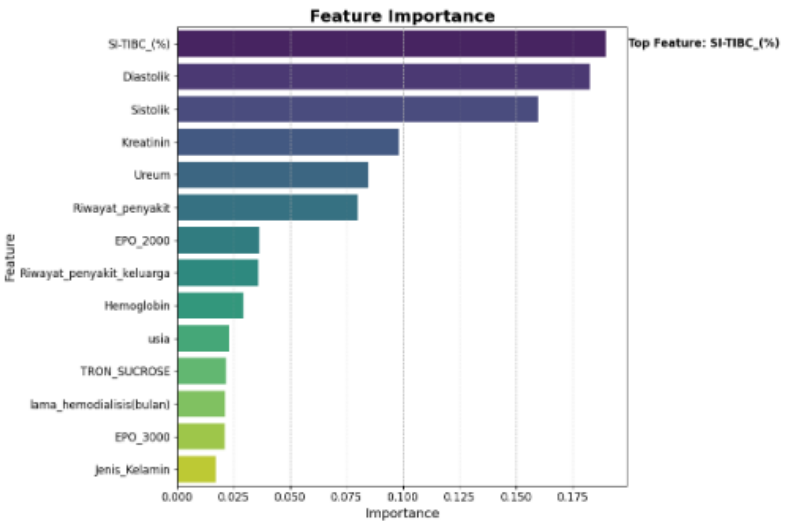
AUC sebesar 0,95 berarti model memberikan probabilitas kelas hipotensi yang lebih tinggi pada pasien hipotensi dibandingkan pasien lainnya pada 95% pasangan yang dibandingkan. Informasi itu ada di dalam keluaran model. Yang gagal adalah aturan keputusannya: pada keempat kasus yang terlewat, probabilitas kelas hipotensi kemungkinan berada di posisi kedua, sedikit di bawah probabilitas kelas tanpa komplikasi.

Implikasinya sangat menguntungkan. Perbaikan tidak menuntut model yang lebih baik, melainkan aturan keputusan yang lebih baik. Menurunkan ambang bagi kelas hipotensi, atau menerapkan matriks biaya yang memberi penalti lebih besar pada kekeliruan tersebut, berpeluang menaikkan *recall* secara berarti tanpa melatih ulang apa pun.

Inilah alasan AUC dan *recall* keduanya perlu dilaporkan. AUC menunjukkan apakah informasinya tersedia; *recall* menunjukkan apakah informasi itu dimanfaatkan.

Kepentingan Fitur

Kepentingan fitur pada model terpilih ditunjukkan pada Gambar 6.11.



Gambar 6.11 Kepentingan Fitur pada Model Terpilih *XGBoost*-ABC

Lima fitur teratas berturut-turut adalah SI-TIBC, tekanan diastolik, tekanan sistolik, kreatinin, dan ureum. Kelimanya merupakan parameter klinis yang memang lazim digunakan dalam evaluasi kondisi pasien hemodialisis, sehingga hasil ini memperkuat kewajaran model. Peringkat SI-TIBC sebagai fitur paling berpengaruh juga sejalan dengan matriks korelasi pada Gambar 6.4 yang menempatkannya sebagai variabel dengan korelasi tertinggi terhadap komplikasi.

Secara klinis, dominasi SI-TIBC dapat dijelaskan. Saturasi transferin merupakan penanda status zat besi, dan gangguan status zat besi berkaitan erat dengan anemia yang hampir selalu menyertai gagal ginjal kronik. Anemia menurunkan kemampuan darah mengangkut oksigen, yang pada gilirannya memengaruhi kestabilan hemodinamik selama sesi dialisis. Keberadaan tekanan diastolik dan sistolik pada peringkat berikutnya juga masuk akal mengingat dua dari tiga komplikasi yang diprediksi merupakan gangguan tekanan darah.

Perhatian: Peringkat Bawah Belum Tentu Berarti Tidak Penting

Sebelum kesimpulan diambil dari Gambar 6.11, satu hal perlu diperiksa: ukuran kepentingan apa yang digunakan.

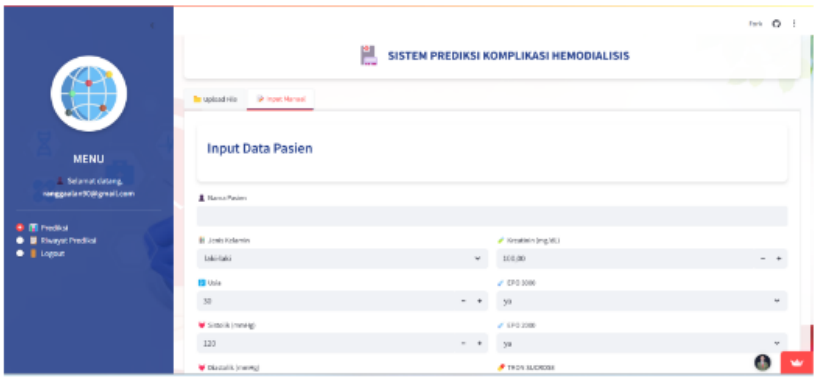
Sebagaimana dibahas pada Subbab 6.2.7, ukuran *weight* menghitung frekuensi penggunaan sebuah fitur sebagai dasar pemisahan, dan ukuran ini bias terhadap fitur dengan banyak nilai berbeda. Perhatikan peringkat pada Gambar 6.11: kelima fitur teratas seluruhnya merupakan fitur kontinu, sedangkan keempat fitur terbawah jenis kelamin, EPO 3000, EPO 2000, dan lama hemodialisis seluruhnya merupakan fitur biner atau berkardinalitas rendah.

Susunan semacam ini tidak dapat langsung dibaca sebagai bukti bahwa jenis kelamin tidak berpengaruh terhadap risiko komplikasi. Fitur biner hanya menyediakan satu titik pemisahan, sehingga secara struktural tidak mungkin muncul sesering fitur kontinu, betapapun kuat pengaruhnya.

Pemeriksaan yang disarankan: tampilkan ulang kepentingan fitur menggunakan ukuran *gain*, dan bandingkan peringkatnya. Apabila urutannya berubah, maka peringkat berdasarkan *weight* memang mencerminkan kardinalitas dan bukan pengaruh. Penggunaan nilai SHAP akan memberikan gambaran yang lebih tepat lagi sekaligus memungkinkan penjelasan pada tingkat pasien individual kemampuan yang justru sangat dibutuhkan pada sistem pendukung keputusan klinis.

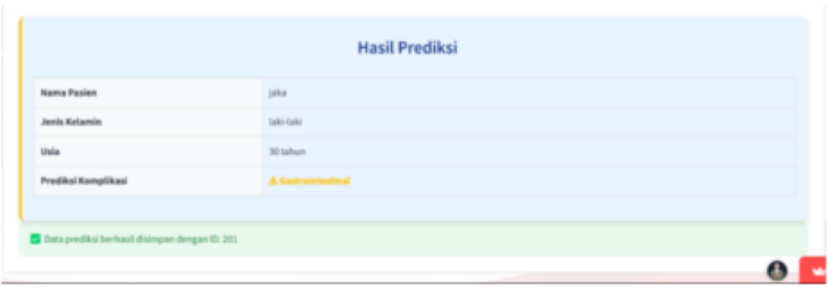
Satu hal lagi perlu ditambahkan mengenai penafsiran kepentingan fitur pada data ini. Sebagaimana terlihat pada Gambar 6.4, korelasi antara ureum dan kreatinin mencapai 0,71. Ketika dua fitur membawa informasi yang hampir sama, pohon keputusan akan memilih salah satunya pada setiap pemisahan, dan pilihan itu sebagian bersifat kebetulan. Akibatnya, kepentingan yang seharusnya terbagi di antara keduanya dapat tampak terpusat pada satu fitur. Kesimpulan bahwa kreatinin lebih penting daripada ureum karena berada satu peringkat di atasnya karena itu tidak dapat ditopang oleh data ini.

Model terpilih diterapkan pada sistem berbasis web yang dibangun menggunakan kerangka kerja *Streamlit*. Sistem menyediakan dua jalur masukan, yaitu pengunggahan berkas untuk prediksi sekaligus pada banyak pasien, dan pengisian manual untuk prediksi pada satu pasien. Tampilan halaman masukan manual ditunjukkan pada Gambar 6.12.



Gambar 6.12 Tampilan Halaman Input Data Pasien

Antarmuka menyajikan seluruh variabel prediktor dalam bentuk isian yang tertata rapi, dengan satuan pengukuran dicantumkan secara eksplisit pada setiap kolom. Pencantuman satuan merupakan detail kecil yang berdampak nyata pada keselamatan penggunaan, sebab kekeliruan satuan pada nilai laboratorium merupakan sumber kesalahan yang lazim terjadi di lingkungan klinis. Hasil prediksi ditampilkan sebagaimana Gambar 6.13.



Gambar 6.13 Tampilan Hasil Prediksi Komplikasi

Sistem menampilkan identitas pasien beserta prediksi jenis komplikasi, dan menyimpan hasilnya ke dalam basis data disertai nomor identifikasi. Ketersediaan riwayat prediksi merupakan fitur yang berguna karena memungkinkan penelusuran ulang keputusan yang pernah diambil.

Catatan: Satu Angka yang Belum Ditampilkan
Keluaran sistem pada Gambar 6.13 berupa satu label komplikasi tanpa disertai tingkat keyakinan. Model sesungguhnya menghasilkan probabilitas untuk keempat kelas, tetapi hanya label dengan probabilitas tertinggi yang ditampilkan.
Perbedaan antara keduanya sangat berarti bagi pengguna klinis. Sebuah prediksi hipertensi dengan probabilitas 0,92 dan sebuah prediksi hipertensi dengan probabilitas 0,35 ketika kelas kedua terpaut sangat tipis memiliki bobot yang sama sekali berbeda dalam pengambilan keputusan, namun keduanya ditampilkan secara identik.
Sebagaimana telah dibahas pada Bab 4, keluaran model tidak dengan sendirinya terkalibrasi sehingga nilai probabilitasnya perlu diperlakukan dengan hati-hati. Meskipun demikian, menampilkan urutan probabilitas keempat kelas tetap jauh lebih informatif daripada satu label tunggal, sebab pengguna dapat melihat apakah model benar-benar yakin atau sesungguhnya sedang ragu di antara dua kemungkinan.
Tambahan yang paling bernilai bagi konteks ini adalah menampilkan probabilitas kelas hipotensi secara terpisah dan menonjol, mengingat kelas inilah yang paling sering terlewat sekaligus paling berbahaya.

Selain kedua tampilan tersebut, sistem menyediakan pengelolaan akun pengguna, penyimpanan riwayat prediksi ke dalam basis data, serta fasilitas pengeksporan hasil ke berkas CSV. Kelengkapan semacam ini menunjukkan bahwa pengembangan tidak berhenti pada model, melainkan berlanjut sampai menjadi perangkat yang benar-benar dapat digunakan dan hal ini merupakan jarak yang sering kali tidak ditempuh oleh Metodologi pembelajaran mesin.

Perlu ditambahkan bahwa penerapan sistem semacam ini pada lingkungan klinis nyata menuntut sejumlah hal yang berada di luar lingkup pemodelan. Penyimpanan data pasien ke dalam basis data menuntut pengendalian akses, penyandian data, serta pencatatan jejak audit yang menunjukkan siapa mengakses data apa dan kapan. Selain itu, prediksi yang tersimpan perlu disertai penanda versi model yang menghasilkannya, sebab ketika model kelak diperbarui, hasil lama tidak lagi dapat dibandingkan langsung dengan hasil baru.

Satu hal terakhir menyangkut penempatan sistem di dalam alur kerja. Prediksi dilakukan menggunakan data yang tersedia sebelum sesi hemodialisis dimulai, sehingga keluarannya paling bermanfaat apabila disajikan pada saat perawat menyiapkan pengaturan mesin yaitu ketika laju ultrafiltrasi dan target penarikan cairan masih dapat disesuaikan. Sebuah prediksi yang akurat namun tiba setelah keputusan diambil tidak memberikan manfaat apa pun. Penetapan titik penyajian dalam alur kerja karena itu sama pentingnya dengan ketepatan prediksinya, dan hal ini hanya dapat ditentukan melalui pengamatan langsung bersama tenaga medis di unit hemodialisis.

Pembahasan dan Keterbatasan

Hasil pada bab ini perlu dibaca bersama sejumlah keterbatasan berikut, yang dikelompokkan menurut tahapannya.

Keterbatasan pada data:

a. Jumlah data hanya 387 baris dengan kelas terkecil sebanyak 58 sampel, sehingga setiap metrik yang dihitung pada data uji memiliki ketelitian yang rendah dan selisih beberapa poin antarmodel sulit dibedakan dari keacakan.
b. Data berasal dari satu rumah sakit sehingga model belum diuji terhadap variasi karakteristik pasien maupun praktik klinis di fasilitas lain.
c. Variabel EPO dan *Tron Sucrose* merupakan catatan tindakan terapi, bukan pengukuran kondisi pasien, sehingga hubungannya dengan komplikasi mencerminkan keputusan klinis dokter dan bukan hubungan fisiologis.

Keterbatasan pada prapemrosesan:

a. Urutan penerapan SMOTE relatif terhadap pembagian data belum dapat dipastikan, dan pola angka pada Tabel 6.6 mengarah pada kemungkinan kebocoran data yang menaikkan akurasi *hold-out* secara semu.
b. Imputasi nilai kosong dilakukan menggunakan statistik keseluruhan data, bukan hanya data pelatihan.
c. Pembagian data tidak menggunakan stratifikasi, sehingga proporsi kelas pada bagian pelatihan dan pengujian tidak dijamin sebanding.
d. Variabel kategorik nominal diproses menggunakan *label encoding* yang menetapkan urutan pada kategori yang sesungguhnya tidak berurutan.

Keterbatasan pada pemodelan dan evaluasi:

a. Kombinasi parameter hasil optimalisasi memuat *learning_rate* sebesar 0,929 dan *min_child_weight* sebesar 0 yang secara teori sangat rentan terhadap *overfitting*.
b. Fungsi kebugaran ABC menggunakan akurasi keseluruhan, sehingga proses optimasi tidak memberikan perhatian khusus pada kelas hipotensi yang paling penting secara klinis.
c. Nilai *recall* kelas hipotensi hanya 0,69, dan enam pasien berkomplikasi diprediksi sebagai tanpa komplikasi.
d. Aturan keputusan menggunakan probabilitas tertinggi tanpa penalaan ambang, padahal AUC yang tinggi menunjukkan bahwa informasi untuk keputusan yang lebih baik sesungguhnya tersedia.
e. Kepentingan fitur dilaporkan menggunakan ukuran yang bias terhadap fitur berkardinalitas tinggi.

Keterbatasan pada sistem:

a. Keluaran sistem berupa label tunggal tanpa tingkat keyakinan maupun urutan probabilitas kelas.
b. Sistem belum diuji pada alur kerja nyata bersama tenaga medis, sehingga kegunaan praktisnya belum terukur.

Arah pengembangan disusun berikut ini menurut perbandingan antara dampak dan biayanya. Kelompok pertama tidak menuntut data tambahan maupun pelatihan ulang: mengubah aturan keputusan agar mengutamakan *recall* kelas hipotensi, menampilkan urutan probabilitas keempat kelas pada antarmuka, dan melaporkan ulang kepentingan fitur menggunakan ukuran *gain* atau nilai SHAP.

Kelompok kedua menuntut pelatihan ulang namun tidak menuntut data tambahan: menjalankan ulang seluruh percobaan dengan SMOTE yang diterapkan hanya pada bagian pelatihan melalui *Pipeline*, mengganti fungsi kebugaran ABC menjadi rerata makro *F1-score* atau *recall* kelas hipotensi, membatasi rentang pencarian *learning_rate* pada nilai yang wajar, menerapkan stratifikasi pada pembagian data, dan menambahkan kriteria penghentian dini pada ABC untuk memangkas waktu pelatihan.

Kelompok ketiga menuntut upaya yang lebih besar: memperluas data melalui kerja sama dengan beberapa fasilitas layanan hemodialisis, menambahkan variabel yang secara klinis diketahui berkaitan dengan komplikasi intradialitik seperti berat badan kering dan laju ultrafiltrasi, mengubah rumusan permasalahan dari klasifikasi jenis komplikasi menjadi penaksiran risiko per sesi, serta melakukan uji coba pada alur kerja nyata bersama tenaga medis.

➢ Ringkasan

Bab ini telah menguraikan pembangunan model prediksi komplikasi hemodialisis menggunakan *XGBoost* yang *hyperparameter*-nya ditala melalui algoritma *Artificial Bee Colony*, beserta penerapannya pada sistem berbasis web. Beberapa hal penting dapat dirangkum sebagai berikut.

a. Pada data tabular, metode berbasis pohon seperti *XGBoost* lebih sesuai daripada jaringan saraf dalam karena hubungan antara variabel klinis dan komplikasi umumnya berbentuk ambang, dan pohon keputusan memodelkan ambang secara alamiah.
b. Optimalisasi ABC meningkatkan akurasi sebesar satu sampai empat poin persentase secara konsisten di seluruh skenario, dengan peningkatan paling stabil justru pada kondisi yang paling menantang berupa data tidak seimbang dan validasi silang.

c. Selisih sembilan poin antara akurasi *hold-out* sebesar 96% dan validasi silang sebesar 87% terlalu besar untuk dijelaskan oleh keacakan, dan pola angkanya mengarah pada kemungkinan kebocoran data akibat penerapan SMOTE sebelum pembagian data.
d. Keputusan memilih model dengan akurasi 87% dan bukan 96% merupakan keputusan yang tepat, sebab skema evaluasi yang lebih layak dipercaya lebih penting daripada angka yang lebih tinggi.
e. Kelas hipotensi yang paling berbahaya secara klinis justru memiliki *recall* terendah sebesar 0,69, dengan empat dari enam belas kasus diprediksi sebagai tanpa komplikasi.
f. Nilai AUC sebesar 0,95 pada kelas hipotensi menunjukkan bahwa informasi untuk mengenali kelas tersebut sesungguhnya tersedia di dalam keluaran model, sehingga perbaikan dapat dicapai melalui penalaan aturan keputusan tanpa melatih ulang model.
g. Kombinasi parameter hasil optimalisasi yang memuat *learning_rate* sebesar 0,929 dan *min_child_weight* sebesar 0 merupakan pertanda yang perlu ditelusuri, sebab algoritma optimasi akan mengejar apa pun yang diminta oleh fungsi kebugarannya.

Gagasan yang paling penting untuk dibawa dari bab ini adalah bahwa menjalankan lebih dari satu skema evaluasi bukan sekadar kelengkapan, melainkan alat diagnosis. Apabila Metodologi ini hanya menjalankan skema *hold-out*, angka 96% akan diterima tanpa pertanyaan dan tidak akan ada yang mengetahui bahwa performa sesungguhnya berada di kisaran 87%. Justru karena kedua skema dijalankan berdampingan, selisih di antara keduanya menjadi terlihat dan selisih itulah yang menyampaikan informasi paling berharga pada keseluruhan Metodologi.

Buku ini mencatat pola yang sama pada keenam babnya, meskipun muncul dalam wujud yang berbeda-beda. Bab 1 menemukan model yang benar karena alasan yang keliru. Bab 2 menemukan satu model yang tampak baik atau buruk bergantung pada cara pembacaan metriknya. Bab 3 menemukan angka yang benar secara perhitungan namun menyesatkan tanpa pembanding. Bab 4 menemukan perbandingan yang tampak adil namun runtuh

oleh hal-hal di luar tabel konfigurasi. Bab 5 menemukan kegagalan yang menyerupai kegagalan model namun berasal dari satu baris kode yang tidak pernah ditulis. Bab ini menemukan angka tertinggi yang justru merupakan angka yang paling tidak layak dipercaya. Keenamnya menunjuk pada keterampilan yang sama: kesediaan untuk bertanya, sebelum menerima sebuah angka, apa sesungguhnya yang sedang diukur oleh angka itu.

➢ Daftar Pustaka

Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research, 13, 281–305.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of *Artificial Intelligence* Research, 16, 321–357.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Dalam Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, hlm. 785–794.

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics, 29(5), 1189–1232.

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why Do Tree-Based Models Still Outperform *Deep Learning* on Typical Tabular Data? Dalam Advances in Neural Information Processing Systems (NeurIPS).

Isroin, L. (2016). Manajemen Cairan pada Pasien Hemodialisis untuk Meningkatkan Kualitas Hidup. Ponorogo: Unmuh Ponorogo Press.

Kaggle Karaboga, D., & Basturk, B. (2007). A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. Journal of Global Optimization, 39(3), 459–471.

Kementerian Kesehatan Republik Indonesia. (2018). Hasil Utama Riset Kesehatan Dasar 2018. Jakarta: Balitbangkes.

Kuhn, M., & Johnson, K. (2019). *Feature* Engineering and Selection: A Practical Approach for Predictive Models. Boca Raton: CRC Press.

Lundberg, S. M., & Lee, S. I. (2017). A Unified Approach to Interpreting Model Predictions. Dalam Advances in Neural Information Processing Systems (NeurIPS), hlm. 4765–4774.

Perhimpunan Nefrologi Indonesia. (2020). 13th Report of Indonesian Renal Registry. Jakarta: PERNEFRI.

Sandberg, F., dkk. (2021). Intradialytic Hypotension: Mechanisms, Risk Factors and Management. Clinical Kidney Journal, 14(1), 8–17.

Strobl, C., Boulesteix, A. L., Zeileis, A., & Hothorn, T. (2007). Bias in *Random Forest* Variable Importance Measures: Illustrations, Sources and a Solution. BMC Bioinformatics, 8, 25.

Varoquaux, G., & Colliot, O. (2023). Evaluating Machine Learning Models and Their Diagnostic Value. Dalam Machine Learning for Brain Disorders. New York: Springer.

World Health Organization. (2021). Global Health Estimates: Leading Causes of Death. Jenewa: WHO.

BAB 7
KLASIFIKASI EMPAT KELAS TUMOR OTAK PADA CITRA MRI: PERBANDINGAN CNN, *VISION TRANSFORMER*, DAN MODEL HIBRIDA CNN-ViT

Setelah mempelajari bab ini, pembaca diharapkan mampu:

a. Menjelaskan karakteristik empat kelas tumor otak pada citra MRI serta implikasi klinis dari kekeliruan klasifikasi di antaranya.
b. Menguraikan prinsip kerja *Vision Transformer*, mulai dari pemecahan citra menjadi patch hingga mekanisme multi-head *self-attention*.
c. Menjelaskan konsep bias induktif dan alasan *Vision Transformer* memerlukan data jauh lebih banyak daripada CNN.
d. Merancang model hibrida melalui penggabungan vektor fitur dari dua jalur pengekstraksi yang berbeda.
e. Membaca kurva pelatihan untuk membedakan model yang telah *overfitting* dari model yang belum selesai belajar.
f. Menilai apakah selisih akurasi antarmodel bermakna secara statistik pada himpunan uji berukuran kecil.
g. Mengenali kemungkinan persoalan mutu label pada dataset publik melalui pola kekeliruan yang dihasilkan model.

➢ Pendahuluan

Tumor merupakan pertumbuhan sel yang berlangsung secara abnormal dan dapat muncul pada hampir seluruh bagian tubuh manusia, termasuk otak. Tumor otak timbul akibat mutasi DNA yang menyebabkan sel yang seharusnya tumbuh dan mati pada kurun waktu tertentu justru tetap hidup dan berkembang biak tanpa kendali. Karena otak terkurung di dalam rongga tengkorak yang volumenya tetap, pertumbuhan massa sekecil apa pun akan menaikkan tekanan intrakranial dan menekan jaringan di sekitarnya, sehingga menimbulkan gejala yang berkembang cepat.

Angka yang menyertai penyakit ini menunjukkan beratnya beban yang ditimbulkan. Tumor otak memengaruhi sekurang-kurangnya 250.000 orang di seluruh dunia setiap tahun, termasuk anak-anak di bawah usia lima belas tahun. *American Society of Clinical Oncology* mencatat angka harapan hidup lima tahun bagi penderita tumor otak hanya sebesar 36%. Pada tahun 2022 melaporkan peningkatan yang sangat tajam sepanjang 1990 sampai 2019, dengan kenaikan angka kejadian mencapai 109,04% dan kenaikan angka kematian sebesar 76,36%; untuk wilayah Asia Tenggara, kenaikannya masing-masing sebesar 2,55% dan 2,23%.

Diagnosis yang cepat dan tepat karena itu menjadi penentu. Keterlambatan penanganan dapat berujung pada kerusakan jaringan otak yang menetap, hilangnya fungsi tubuh, penyebaran ke bagian lain, hingga kematian. Yang membedakan permasalahan ini dari sekadar mendeteksi keberadaan tumor adalah bahwa jenis tumornya turut menentukan tindakan. Meningioma, glioma, dan tumor pituitari memerlukan pendekatan terapi yang berbeda, sehingga sebuah sistem yang hanya menyatakan ada atau tidak adanya tumor belum memadai bagi kebutuhan klinis.

Bab ini kembali bekerja pada citra MRI otak sebagaimana Bab 2 dan Bab 5, namun dengan pertanyaan yang berbeda. Bab 2 menanyakan di mana letak lesi melalui segmentasi. Bab 5 menanyakan irisan mana yang mengandung lesi beserta volumenya. Bab ini menanyakan jenis tumor apa yang tampak pada citra, yaitu permasalahan klasifikasi empat kelas yang mencakup glioma, meningioma, tumor pituitari, dan kondisi tanpa tumor.

Yang menjadikan bab ini berbeda dari seluruh bab sebelumnya adalah arsitektur yang dibandingkan. Enam bab terdahulu seluruhnya menggunakan arsitektur berbasis konvolusi atau metode berbasis pohon. Bab ini memperkenalkan *Vision Transformer* (ViT), yaitu pendekatan yang meninggalkan konvolusi sepenuhnya dan menggantinya dengan mekanisme *self-attention* yang semula dikembangkan untuk pemrosesan bahasa alami. Perbedaannya bersifat mendasar: CNN membangun pemahaman citra secara bertahap dari ketetanggaan lokal, sedangkan ViT memandang seluruh bagian citra secara serentak dan membiarkan model sendiri yang menentukan bagian mana yang saling berkaitan.

Kedua pendekatan tersebut memiliki keunggulan yang saling melengkapi, dan hal itulah yang melatarbelakangi gagasan model hibrida. CNN unggul dalam menangkap pola lokal seperti tepi, tekstur, dan bentuk, sedangkan ViT unggul dalam menangkap hubungan berjarak jauh dan konteks menyeluruh. Model hibrida CNN-ViT menjalankan keduanya secara paralel pada citra yang sama, kemudian menggabungkan vektor fitur yang dihasilkan sebelum melakukan klasifikasi. Bab ini membandingkan ketiganya CNN murni, ViT murni, dan hibrida CNN-ViT pada dataset yang sama melalui tiga puluh enam skenario pengujian yang mencakup variasi rasio pembagian data, algoritma optimasi, dan nilai *learning rate*.

Sebagaimana bab-bab sebelumnya, hasil yang diperoleh akan dibaca secara kritis. Pertanyaan utama yang ditelusuri bukan sekadar model mana yang paling akurat, melainkan apakah perbandingannya sendiri sudah adil. Kurva pelatihan yang akan dibahas pada Subbab 7.4.5 mengungkap bahwa ketiga model berada pada kondisi belajar yang sangat berbeda ketika pelatihan dihentikan, dan kenyataan itu mengubah cara seluruh angka pada bab ini harus ditafsirkan.

➢ Pengantar Teori

Bagian ini menguraikan landasan teori yang diperlukan pada bab ini. Konsep dasar CNN telah dibahas pada Bab 1 dan diperdalam pada bab-bab berikutnya, sehingga pembahasannya di sini diringkas dan difokuskan pada rumusan matematisnya. Bagian terbesar dicurahkan pada *Vision Transformer* yang baru diperkenalkan pada bab ini, serta pada konsep bias induktif yang menjelaskan perbedaan perilaku kedua arsitektur.

Tumor Otak dan Empat Kelas yang Dianalisis

Empat kelas yang dianalisis pada bab ini beserta karakteristiknya dirangkum pada Tabel 7.1. Memahami perbedaan di antara keempatnya bukan sekadar pelengkap, sebab pola kekeliruan model yang dibahas pada Subbab 7.4.10 hanya dapat dinilai kewajarannya apabila pembaca mengetahui kelas mana yang memang mirip secara radiologis.

Tabel 7.1 Empat Kelas pada Dataset Tumor Otak

Kelas	Asal Jaringan	Ciri Radiologis	Sifat
Glioma	Sel glia di dalam jaringan otak	Batas tidak tegas, menyusup ke jaringan sekitar	Umumnya ganas
Meningioma	Selaput pembungkus otak	Batas tegas, menempel pada duramater, mendesak bukan menyusup	Umumnya jinak
Tumor pituitari	Kelenjar hipofisis	Terletak pada sella tursika di dasar otak	Umumnya jinak
Tanpa tumor	—	Struktur otak tampak utuh	—

Catatan Klinis: Mengapa Glioma dan Meningioma Sulit Dibedakan

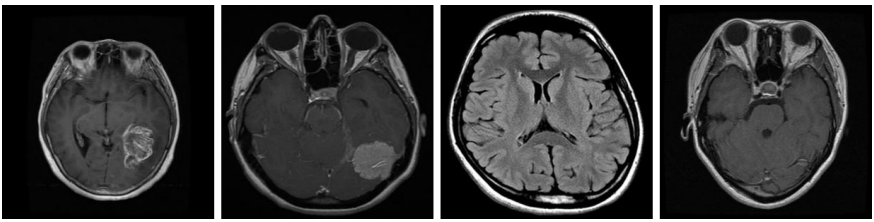
Perbedaan antara glioma dan meningioma bersifat mendasar secara klinis. Meningioma berasal dari selaput pembungkus otak sehingga letaknya *di luar* jaringan otak dikenal sebagai lesi ekstra-aksial dan umumnya bersifat jinak serta dapat diangkat secara utuh melalui pembedahan. Glioma berasal dari sel glia *di dalam* jaringan otak lesi intra-aksial bersifat menyusup sehingga batasnya sulit ditentukan, dan sebagian besar bersifat ganas.

Konsekuensi terapinya berlawanan arah. Meningioma sering kali dapat disembuhkan melalui pembedahan tunggal; glioma menuntut kombinasi pembedahan, radioterapi, dan kemoterapi, dengan prognosis yang jauh lebih buruk.

Meskipun demikian, keduanya dapat tampak serupa pada citra MRI potongan tunggal, terutama ketika meningioma berukuran besar mendesak jaringan otak sehingga tampak seolah berada di dalamnya, atau ketika glioma memiliki batas yang tampak relatif tegas. Radiolog membedakan keduanya melalui petunjuk tambahan seperti keberadaan *dural tail* pada meningioma serta pola penyangatan kontras petunjuk yang tidak selalu tampak pada satu potongan citra.

Kenyataan ini perlu diingat, sebab kekeliruan antara kedua kelas inilah yang akan mendominasi hasil pada ketiga model yang dibandingkan.

Contoh citra dari keempat kelas ditunjukkan pada Gambar 7.1.



Gambar 7.1 Contoh Citra MRI pada Keempat Kelas (dari Kiri: Meningioma, Glioma, Tanpa Tumor, Tumor Pituitari)

Convolutional Neural Network

CNN merupakan kelas jaringan saraf tiruan yang dirancang untuk data berstruktur grid seperti citra. Operasi intinya adalah konvolusi, yaitu penggeseran kernel berukuran kecil ke seluruh permukaan citra untuk menghasilkan *feature map*.

$$S(i,j) = f(\sum_m \sum_n I(i + m, j + n) \cdot K(m,n) + b) \quad (7.1)$$

Dengan I menyatakan matriks masukan, K menyatakan kernel, b menyatakan bias, dan f menyatakan fungsi aktivasi. Dua parameter turut menentukan perilaku operasi ini. *Padding* menambahkan piksel bernilai nol pada tepi masukan sehingga informasi di tepi tidak hilang, sedangkan *stride* menentukan besar pergeseran kernel pada setiap langkah. Ukuran keluaran yang dihasilkan dapat dihitung sebagai berikut.

$$Output = \frac{N - F + 2P}{S} + 1 \quad (7.2)$$

Dengan N menyatakan ukuran spasial masukan, F menyatakan ukuran kernel, P menyatakan besar *padding*, dan S menyatakan *stride*. Keluaran konvolusi kemudian dilewatkan pada fungsi aktivasi ReLU yang memetakan nilai negatif menjadi nol.

$$f(x) = max(0, x) \quad (7.3)$$

Tahap berikutnya adalah *pooling* yang mengecilkan dimensi spasial sambil mempertahankan informasi paling menonjol. Jenis yang paling banyak digunakan pada citra medis adalah *max pooling*.

$$P(x, y) = max_{(i,j) \in R(x,y)} F(i, j) \quad (7.4)$$

Dengan R(x, y) menyatakan jendela *pooling* pada posisi tersebut. Ukuran keluarannya dihitung dengan cara yang serupa dengan konvolusi.

$$Output = \frac{N - F}{S} + 1 (7.5)$$

Setelah rangkaian konvolusi dan *pooling*, peta fitur diratakan menjadi vektor satu dimensi melalui operasi *flatten*, kemudian diteruskan ke *fully connected layer* yang melakukan klasifikasi. Lapisan keluaran menggunakan fungsi softmax untuk mengubah keluaran mentah menjadi distribusi probabilitas.

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad (7.6)$$

Fungsi kerugian yang bersesuaian adalah *categorical cross-entropy*, sebagaimana telah dibahas pada Bab 4 melalui Persamaan 4.2 dan dituliskan kembali di sini untuk kelengkapan.

$$L = - \sum_{i=1}^{K} y_i \, log(\hat{y}_i) \quad (7.7)$$

Vision Transformer

Vision Transformer diperkenalkan Dosovitskiy dan rekan pada tahun 2020 dengan gagasan yang pada saat itu terdengar berani: memperlakukan citra seperti kalimat. Pada pemrosesan bahasa, sebuah kalimat dipecah menjadi rangkaian kata yang masing-masing diubah menjadi vektor, lalu hubungan antarkata dipelajari melalui mekanisme *attention*. ViT menerapkan gagasan yang sama pada citra dengan memecahnya menjadi rangkaian potongan berukuran tetap yang disebut *patch*, kemudian memperlakukan setiap *patch* sebagaimana sebuah kata diperlakukan.

Langkah pertama adalah pemecahan citra. Sebuah citra dua dimensi diubah menjadi rangkaian *patch* yang telah diratakan.

$$x \in \mathbb{R}^{H \times W \times C} \rightarrow x_p \in \mathbb{R}^{N \times (P^2 \cdot C)} \quad (7.8)$$

Dengan H dan W menyatakan tinggi dan lebar citra, C menyatakan jumlah kanal, dan P menyatakan ukuran sisi setiap *patch*. Jumlah *patch* yang terbentuk ditentukan oleh perbandingan berikut.

$$N = \frac{H \cdot W}{P^2} \quad (7.9)$$

Setiap *patch* yang telah diratakan kemudian diproyeksikan secara linear menjadi vektor berdimensi tetap. Karena rangkaian *patch* pada dasarnya tidak memuat informasi mengenai posisinya di dalam citra, ditambahkan *position embedding* yang dipelajari bersama model. Selain itu disisipkan satu token khusus di awal rangkaian, dikenal sebagai *class token*, yang berperan sebagai tempat berkumpulnya informasi menyeluruh untuk keperluan klasifikasi.

$$z_0 = [\, x_{class} \,;\, x_p{}^1 \, E \,;\, \ldots \,;\, x_p{}^N \, E \,] + E_{pos} \quad (7.10)$$

dengan E menyatakan matriks proyeksi linear dan E indeks pos menyatakan *position embedding*. Rangkaian vektor tersebut kemudian diproses melalui sejumlah blok *transformer encoder*. Setiap blok diawali dengan *layer normalization* yang menormalkan setiap vektor terhadap rerata dan simpangan bakunya sendiri.

$$x' = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \quad (7.11)$$

Perlu diperhatikan bahwa normalisasi ini bekerja pada arah yang berbeda dari *batch normalization* yang lazim pada CNN. *Batch normalization* menormalkan setiap fitur terhadap seluruh sampel dalam satu *batch*, sedangkan *layer normalization* menormalkan setiap sampel terhadap seluruh fiturnya sendiri. Perbedaan ini membuat *layer normalization* tidak bergantung pada ukuran *batch* dan tetap berperilaku sama saat pelatihan maupun inferensi.

Bagian inti dari setiap blok adalah mekanisme *self-attention*. Setiap vektor masukan diproyeksikan menjadi tiga peran, yaitu *query*, *key*, dan *value*. Kesesuaian antara *query* sebuah *patch* dan *key* seluruh *patch* lain menentukan seberapa besar perhatian yang diberikan kepada masing-masing.

$$Attention(Q, K, V) = softmax\left(\frac{Q\,K^T}{\sqrt{d_k}}\right) V \quad (7.12)$$

Dengan d indeks k menyatakan dimensi vektor *key*. Pembagian dengan akar dimensi tersebut bukan hiasan matematis: tanpa pembagian itu, hasil perkalian titik antara dua vektor berdimensi besar akan menghasilkan nilai yang sangat besar, sehingga fungsi softmax mendekati bentuk ekstrem dan gradiennya nyaris hilang. Penyekalaan menjaga nilai tetap berada pada rentang yang membuat proses pelatihan tetap berjalan.

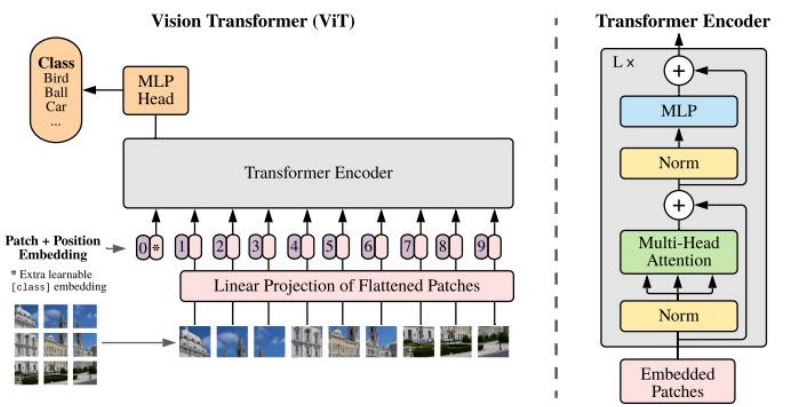
Mekanisme tersebut dijalankan beberapa kali secara paralel melalui *multi-head attention*, sehingga model dapat memperhatikan beberapa jenis hubungan sekaligus.

$$MultiHead(Q,K,V) = Concat(head_1, \dots, head_h)\, W^O \quad (7.13)$$

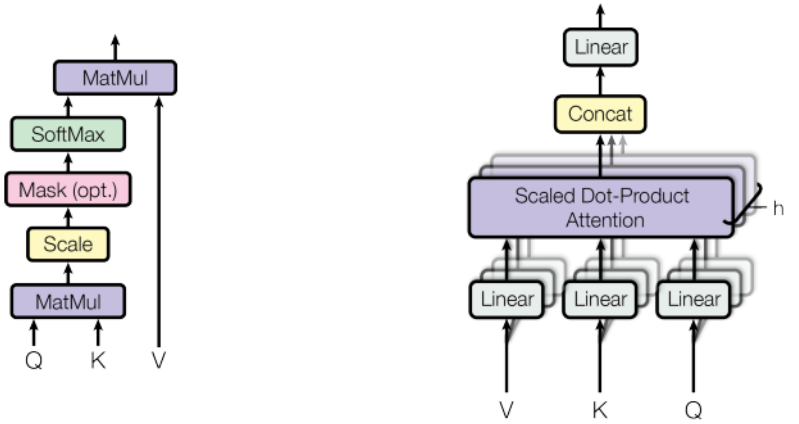
Setiap blok *encoder* diakhiri dengan *multilayer perceptron*, dan kedua submodulnya dibungkus koneksi residual sebagaimana pada *ResNet* yang telah dibahas pada Bab 1.

$$z_l' = MHA(LN(z_{l-1})) + z_{l-1}, \quad z_l = MLP(LN(z_l')) + z_l' \quad (7.14)$$

Arsitektur lengkapnya ditunjukkan pada Gambar 7.2, sedangkan rincian mekanisme *attention* ditunjukkan pada Gambar 7.3.



Gambar 7.2 Arsitektur *Vision Transformer*



Gambar 7.3 *Scaled Dot-Product Attention* dan *Multi-Head Attention*

Bias Induktif dan Kebutuhan Data

Bagian ini merupakan yang paling menentukan bagi pemahaman hasil pada bab ini, meskipun tidak memuat satu pun persamaan. Pertanyaan yang dijawabnya sederhana: mengapa ViT, yang secara teori lebih fleksibel daripada CNN, justru sering berperforma lebih rendah pada dataset berukuran kecil.

Jawabannya terletak pada konsep *bias induktif*, yaitu asumsi yang telah tertanam di dalam rancangan sebuah arsitektur sebelum ia melihat data apa pun. CNN memuat tiga asumsi semacam itu. Asumsi *lokalitas* menyatakan bahwa piksel yang berdekatan lebih berkaitan daripada piksel yang berjauhan, dan asumsi ini diwujudkan melalui kernel yang hanya melihat ketetanggaan kecil. Asumsi *pembagian bobot* menyatakan bahwa pola yang sama dapat muncul di posisi mana pun, sehingga kernel yang sama digunakan di seluruh bidang citra. Asumsi *hierarki* menyatakan bahwa pola kompleks tersusun dari pola sederhana, dan hal ini diwujudkan melalui penumpukan lapisan.

Ketiga asumsi tersebut kebetulan benar untuk citra alami maupun citra medis. Karena benar, CNN tidak perlu mempelajarinya dari data asumsi itu sudah tersedia secara cuma-cuma sejak awal. *Vision Transformer* tidak memuat asumsi tersebut sama sekali. Mekanisme *self-attention* memperlakukan seluruh pasangan *patch* secara setara, tanpa memandang jaraknya. Kebebasan ini merupakan

keunggulan sekaligus beban: model dapat menemukan hubungan yang tidak terpikirkan sebelumnya, tetapi ia harus mempelajari bahkan hal-hal paling dasar sekalipun, termasuk bahwa piksel bertetangga umumnya saling berkaitan.

Peringatan: ViT Menuntut Data Jauh Lebih Banyak daripada CNN

Dosovitskiy dan rekan menyatakan hal ini secara eksplisit pada makalah yang memperkenalkan ViT. Ketika dilatih pada dataset berukuran menengah seperti *ImageNet* dengan sekitar satu juta citra, ViT berperforma *lebih rendah* daripada *ResNet* berukuran sebanding. Keunggulan ViT baru muncul ketika dilatih pada dataset berukuran empat belas juta citra, dan menjadi sangat jelas pada tiga ratus juta citra.

Bab ini menggunakan 2.000 citra. Angka tersebut lima ratus kali lebih kecil daripada *ImageNet* dan seratus lima puluh ribu kali lebih kecil daripada dataset tempat ViT membuktikan keunggulannya.

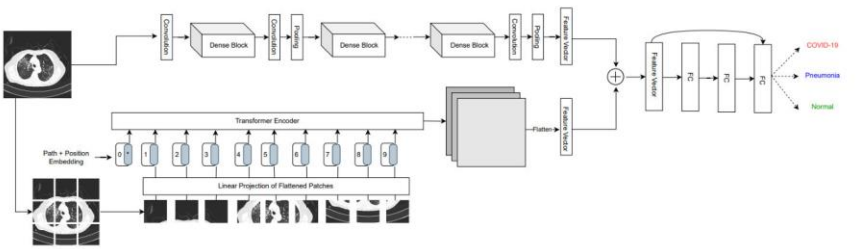
Konsekuensinya perlu dinyatakan sejak awal agar hasil pada Subbab 7.4 tidak disalahtafsirkan. Apabila ViT berperforma lebih rendah daripada CNN pada bab ini, hal itu *bukan* merupakan bukti bahwa ViT merupakan arsitektur yang lebih lemah. Yang terbukti adalah bahwa ViT yang dilatih dari awal pada 2.000 citra berperforma lebih rendah daripada CNN pada kondisi yang sama pernyataan yang jauh lebih terbatas dan justru sudah diperkirakan oleh teori.

Jalan keluar yang lazim ditempuh adalah *transfer learning*: menggunakan ViT yang telah dilatih pada *ImageNet*-21k atau JFT-300M, lalu menyesuaikannya pada data yang tersedia. Dengan cara itu, seluruh pengetahuan dasar mengenai struktur citra sudah dimiliki model sebelum ia melihat satu pun citra MRI. Bab ini melatih ViT sepenuhnya dari awal tanpa bobot awal apa pun, dan pilihan tersebut merupakan penjelasan paling langsung bagi hasil yang diperoleh.

Model Hibrida CNN-ViT

Gagasan model hibrida berangkat dari pengamatan bahwa keunggulan CNN dan ViT bersifat saling melengkapi, bukan saling menggantikan. CNN unggul dalam menangkap fitur lokal berupa tepi, tekstur, dan bentuk, sedangkan ViT unggul dalam memodelkan hubungan berjarak jauh dan memahami konteks citra secara menyeluruh. Model hibrida berupaya memperoleh keduanya sekaligus.

Terdapat beberapa cara menggabungkan keduanya. Penggabungan *berurutan* menempatkan CNN sebagai pengekstraksi fitur awal, kemudian keluarannya diteruskan ke blok *transformer*. Penggabungan *paralel* menjalankan keduanya secara terpisah pada citra yang sama, lalu menyatukan vektor fitur yang dihasilkan. Bab ini menggunakan pendekatan kedua, sebagaimana dilakukan Nguyen dan rekan pada klasifikasi COVID-19 berbasis citra rontgen yang ditunjukkan pada Gambar 7.4.



Gambar 7.4 Model Hibrida CNN-ViT sebagai Acuan Perancangan

Penggabungan dilakukan melalui operasi *concatenate* pada vektor fitur kedua jalur.

$$f_{hybrid} = [\, f_{CNN} \,;\, f_{ViT} \,] \in \mathbb{R}^{(d_{CNN} + d_{ViT})} \quad (7.15)$$

Vektor gabungan tersebut kemudian diteruskan ke *fully connected layer* dengan aktivasi softmax. Perlu diperhatikan bahwa penggabungan melalui *concatenate* mempertahankan kedua sumber informasi secara utuh dan menyerahkan kepada lapisan klasifikasi untuk menimbang keduanya. Alternatif berupa penjumlahan akan mencampur keduanya sehingga tidak lagi dapat dipisahkan, dan menuntut kedua vektor memiliki dimensi yang sama.

Algoritma Optimasi Adam dan RMSprop

Dua algoritma optimasi dibandingkan pada bab ini. RMSprop yang dikembangkan Geoffrey Hinton menyesuaikan *learning rate* setiap parameter berdasarkan rerata bergerak kuadrat gradiennya.

$$E_t = \gamma E_{t-1} + (1 - \gamma) g_t{}^2, \quad \theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{E_t + \epsilon}} g_t \quad (7.16)$$

Gagasannya sederhana namun efektif: parameter yang gradiennya besar dan bergejolak memperoleh langkah yang lebih kecil, sedangkan parameter yang gradiennya kecil memperoleh langkah yang lebih besar. Adam melangkah lebih jauh dengan menambahkan estimasi momen pertama yang berperan sebagai momentum.

$$\theta_{t+1} = \theta_t - \alpha \frac{m_t{}'}{\sqrt{v_t{}'} + \epsilon} \quad (7.17)$$

Dengan m dan v beraksen prima menyatakan estimasi momen pertama dan kedua yang telah dikoreksi terhadap bias awal, sebagaimana telah diuraikan pada Bab 5 melalui Persamaan 5.13 sampai 5.15. Adam pada dasarnya merupakan gabungan antara RMSprop dan momentum, dan pada sebagian besar kasus berperforma sebanding atau lebih baik. Meskipun demikian, tidak terdapat jaminan teoretis bahwa Adam selalu unggul hasil pada bab ini justru memperlihatkan bahwa RMSprop menang pada satu dari tiga arsitektur yang diuji.

Metrik Evaluasi

Metrik evaluasi pada permasalahan multikelas telah dibahas secara rinci pada Bab 4 melalui Persamaan 4.3 dan 4.4. Untuk kelengkapan, definisi dasarnya dituliskan kembali di sini. Akurasi menyatakan proporsi seluruh prediksi yang benar.

$$Accuracy = \frac{\sum_{i=1}^{K} TP_i}{N} \quad (7.18)$$

Precision dan *recall* dihitung untuk setiap kelas dengan memperlakukan kelas tersebut sebagai positif dan seluruh kelas lain sebagai negatif.

$$Precision_i = \frac{TP_i}{TP_i + FP_i}, \quad Recall_i = \frac{TP_i}{TP_i + FN_i} \quad (7.19)$$

$$F1_i = 2 \cdot \frac{Precision_i \cdot Recall_i}{Precision_i + Recall_i} \quad (7.20)$$

Karena dataset pada bab ini seimbang sempurna dengan 500 citra pada setiap kelas, rerata makro dan rerata terbobot menghasilkan nilai yang hampir sama. Perlu pula dicatat bahwa *baseline* kelas mayoritas pada dataset seimbang berisi empat kelas hanya sebesar 25%, sehingga seluruh angka akurasi pada bab ini yang berkisar 85% sampai 94% berada jauh di atasnya dan menandakan pembelajaran yang bermakna.

Regularisasi pada Dataset Berukuran Kecil

Bagian terakhir pada landasan teori ini membahas sekumpulan teknik yang seluruhnya bertujuan sama, yaitu mencegah model menghafal data latih alih-alih mempelajari polanya. Pembahasan ini ditempatkan di sini karena ketiadaan sebagian besar teknik tersebut merupakan salah satu penjelasan utama bagi hasil yang diperoleh pada bab ini.

Overfitting terjadi ketika kapasitas model melampaui informasi yang tersedia pada data latih. Model yang berkapasitas besar mampu menemukan aturan yang menjelaskan setiap sampel latih secara sempurna, termasuk aturan yang sesungguhnya hanya berlaku bagi derau pada sampel tersebut. Gejalanya sangat khas dan mudah dikenali pada kurva pelatihan: akurasi latih terus menaik mendekati satu sementara akurasi validasi mendatar, dan *validation loss* mulai menaik meskipun *training loss* terus menurun.

Teknik pertama untuk menahannya adalah *dropout*, yaitu menonaktifkan sebagian neuron secara acak pada setiap iterasi pelatihan.

$$r \sim Bernoulli(p), \quad y' = \frac{r \odot y}{p} \qquad (7.21)$$

Dengan p menyatakan peluang sebuah neuron dipertahankan dan tanda lingkaran bersilang menyatakan perkalian elemen demi elemen. Pembagian dengan p menjaga agar nilai harapan keluaran tetap sama antara saat pelatihan dan saat inferensi. Gagasan di balik teknik ini adalah memaksa jaringan agar tidak bergantung pada satu neuron tertentu, sebab neuron mana pun dapat menghilang sewaktu-waktu; akibatnya representasi yang terbentuk menjadi lebih tersebar dan lebih tahan.

Teknik kedua adalah *batch normalization*, yaitu menormalkan keluaran setiap lapisan terhadap statistik *batch* yang sedang diproses. Selain mempercepat konvergensi, teknik ini memberikan efek regularisasi karena statistik yang digunakan berbeda-beda pada setiap *batch*, sehingga menimbulkan derau kecil yang menyulitkan penghafalan.

Teknik ketiga adalah augmentasi data, yang telah dibahas berulang kali pada bab-bab sebelumnya. Pada citra MRI otak, transformasi yang sah secara anatomis mencakup rotasi kecil di bawah lima belas derajat, pembalikan horizontal yang menyerupai lesi pada hemisfer berlawanan, pergeseran kecil, serta penyesuaian kecerahan yang meniru variasi antarperangkat. Pembalikan vertikal perlu dihindari sebagaimana telah dibahas pada Bab 5, sebab orientasi atas-bawah pada citra otak bermakna secara anatomis.

Teknik keempat bukan berupa perubahan pada model, melainkan pada prosedur pelatihannya. *EarlyStopping* menghentikan pelatihan ketika *validation loss* tidak lagi membaik selama sejumlah *epoch* tertentu, sedangkan *ModelCheckpoint* menyimpan bobot pada titik validasi terbaik sehingga model yang digunakan bukan model pada *epoch* terakhir melainkan model terbaik yang pernah dicapai.

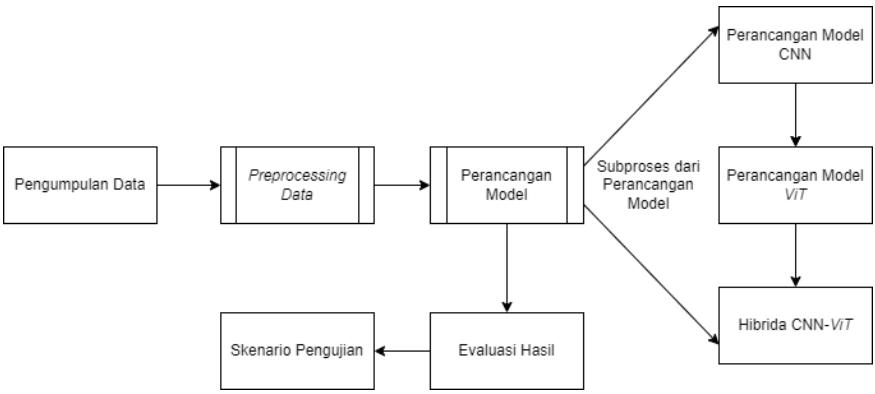
Catatan: Empat Teknik, Tidak Satu pun Digunakan
Rancangan pada bab ini tidak menggunakan *dropout* pada model CNN, tidak menggunakan *batch normalization*, tidak menerapkan augmentasi data, dan tidak menggunakan *EarlyStopping* maupun *ModelCheckpoint*.
Satu-satunya mekanisme regularisasi yang hadir adalah *dropout* sebesar 0,1 di dalam blok *transformer* pada model ViT dan justru ViT merupakan satu-satunya model yang tidak mengalami *overfitting*, meskipun karena alasan yang berbeda, yaitu karena ia belum sempat selesai belajar.
Ketiadaan keempat teknik tersebut menjelaskan sepenuhnya bentuk kurva pelatihan yang akan dibahas pada Subbab 7.4.2 sampai 7.4.4. Yang perlu digarisbawahi adalah bahwa keempatnya tergolong murah: *dropout* dan *batch normalization* masing-masing menambah satu baris pada definisi model,

augmentasi menambah beberapa baris pada penyiapan data, sedangkan *EarlyStopping* dan *ModelCheckpoint* menambah dua baris pada pemanggilan fungsi pelatihan.
Pada dataset berisi 1.600 citra latih, penerapan keempatnya berpeluang memberikan peningkatan yang lebih besar daripada penggantian arsitektur mana pun.

➢ Metodologi

Alur Metodologi

Keseluruhan Metodologi disusun sebagai lima tahap berurutan sebagaimana ditunjukkan pada Gambar 7.5.



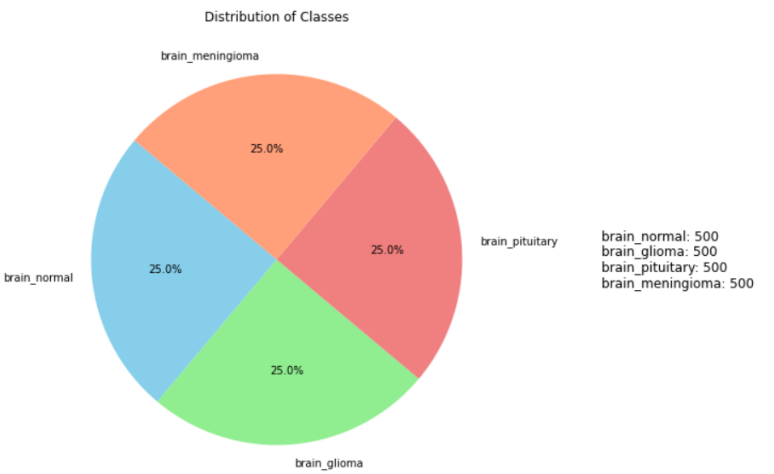
Gambar 7.5 Alur Metodologi

Proses diawali dengan pengumpulan data dari sumber publik, dilanjutkan dengan prapemrosesan yang memuat beberapa subproses. Tahap berikutnya berupa perancangan skenario pengujian yang mencakup variasi rasio pembagian data, algoritma optimasi, dan nilai *learning rate*. Setelah itu dilakukan perancangan ketiga model, yaitu CNN, ViT, dan hibrida CNN-ViT, yang seluruhnya dilatih dan dibandingkan pada data uji. Tahap terakhir berupa evaluasi hasil dari seluruh skenario yang dijalankan.

Dataset

Data yang digunakan merupakan data sekunder berupa dataset *Brain Tumor Classification (MRI)* yang tersedia pada platform Kaggle. Dataset asli memuat 3.264 citra berformat JPG dengan resolusi 512 × 512 piksel dan tiga kanal warna, terbagi menjadi 926 citra glioma, 937 citra meningioma, 500 citra tanpa tumor, dan 901 citra tumor pituitari.

Pada Metodologi ini digunakan 500 citra dari setiap kelas sehingga total data menjadi 2.000 citra. Pemilihan jumlah tersebut menghasilkan dataset yang seimbang sempurna sebagaimana ditunjukkan pada Gambar 7.6.



Gambar 7.6 Distribusi Data pada Keempat Kelas

Keputusan menyeimbangkan data melalui pemotongan kelas mayoritas patut *dice*rmati dari dua sisi. Sisi yang menguntungkan, dataset seimbang membuat seluruh metrik dapat dibaca secara langsung tanpa penyesuaian, dan *baseline* kelas mayoritas menjadi tetap sebesar 25% bagi keempat kelas. Sisi yang merugikan, sebanyak 1.264 citra dari kelas glioma, meningioma, dan pituitari dibuang begitu saja yaitu 39% dari keseluruhan data yang tersedia.

Pada dataset berukuran kecil, pembuangan sebesar itu merupakan biaya yang tinggi. Alternatif yang lebih hemat adalah mempertahankan seluruh data dan menangani ketimpangan melalui pembobotan kelas pada fungsi kerugian, sebagaimana telah dibahas pada Bab 5 melalui Persamaan 5.10. Terlebih lagi, ketimpangan pada dataset asli sesungguhnya tergolong ringan: perbandingan antara kelas terbesar dan terkecil hanya sekitar 1,9 berbanding 1, jauh lebih ringan daripada yang dijumpai pada Bab 3.

Perhatian: Mutu Label pada Dataset Publik Perlu Diverifikasi

Dataset yang digunakan pada bab ini tergolong populer dan banyak dipakai pada berbagai Metodologi. Meskipun demikian, dataset turunan yang lebih baru yang disusun dari sumber yang sama memuat catatan bahwa citra pada kelas glioma dari dataset ini tidak seluruhnya terkategori dengan benar, sehingga pada dataset turunan tersebut kelas glioma diganti dengan citra dari sumber lain.

Catatan ini tidak dapat diverifikasi langsung tanpa pemeriksaan oleh ahli radiologi, dan bab ini tidak menyatakannya sebagai fakta yang pasti. Yang dinyatakan adalah bahwa hal ini merupakan kemungkinan yang perlu diperiksa, terlebih karena sebagaimana akan terlihat pada Subbab 7.4.10 kekeliruan yang paling dominan pada ketiga model justru terjadi antara kelas glioma dan meningioma.

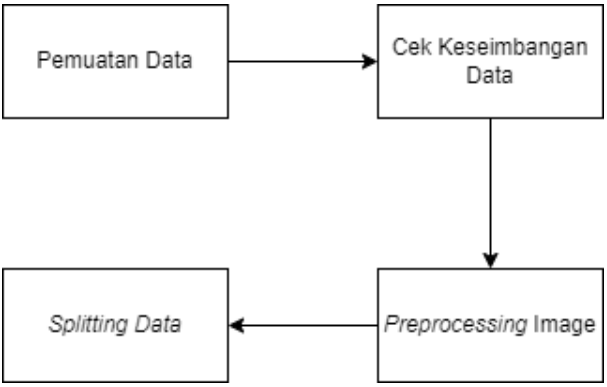
Kaitan antara keduanya penting untuk disadari. Apabila sebagian label pada kelas glioma memang keliru, maka sebagian kekeliruan yang tercatat sebagai kesalahan model sesungguhnya merupakan kesalahan label dan tidak ada arsitektur apa pun yang dapat memperbaikinya. Dalam kondisi seperti itu, upaya menaikkan akurasi dari 93% menjadi 95% melalui pergantian arsitektur menjadi sia-sia, sebab batas atasnya ditentukan oleh mutu label dan bukan oleh kemampuan model.

Pemeriksaan yang disarankan: tampilkan sejumlah citra dari kelas glioma yang keliru diprediksi sebagai meningioma, lalu mintakan penilaian ulang kepada ahli radiologi. Pemeriksaan terhadap dua puluh sampai tiga puluh citra sudah cukup untuk

mengetahui apakah persoalannya terletak pada model atau pada label.

Prapemrosesan Citra

Tahap prapemrosesan disusun sebagai rangkaian empat langkah sebagaimana ditunjukkan pada Gambar 7.7.

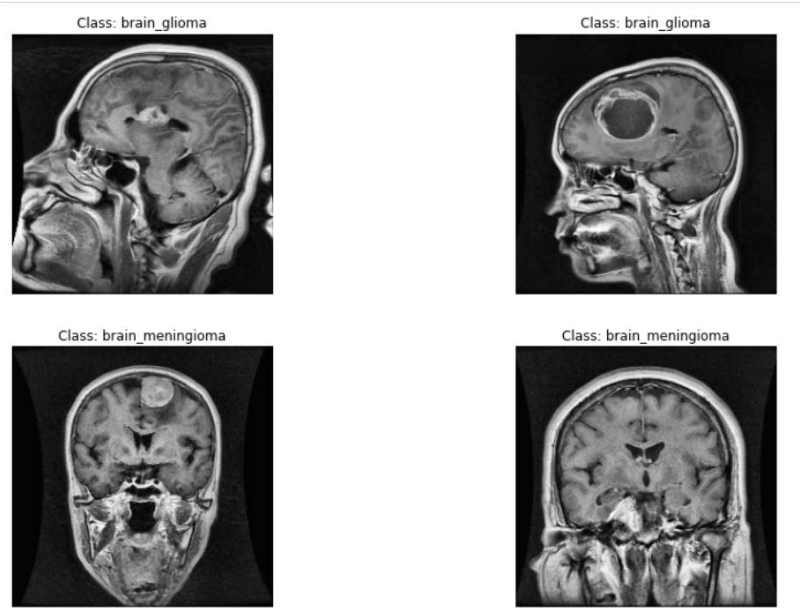


Gambar 7.7 Alur Prapemrosesan Data

Langkah-langkah yang dijalankan diuraikan sebagai berikut.

a. *Pengubahan ukuran.* Seluruh citra diubah dari 512 × 512 piksel menjadi 224 × 224 piksel agar sesuai dengan ukuran masukan ketiga arsitektur sekaligus menekan beban komputasi.
b. *Konversi ke keabuan.* Citra diubah menjadi satu kanal, mengingat citra MRI pada dasarnya memang tidak memuat informasi warna.
c. *Peningkatan kontras.* Diterapkan *Contrast Limited Adaptive Histogram Equalization* (CLAHE) sebagaimana telah diuraikan pada Bab 5 melalui Persamaan 5.7 dan 5.8, untuk memperjelas batas antara massa tumor dan jaringan otak di sekitarnya.
d. *Pengembalian ke tiga kanal.* Citra keabuan direplikasi menjadi tiga kanal identik karena bentuk masukan yang diharapkan model menuntut tiga kanal.

Hasil prapemrosesan ditunjukkan pada Gambar 7.8.



Gambar 7.8 Hasil Prapemrosesan Citra

Peningkatan kontras tampak nyata, terutama pada batas antara massa tumor dan jaringan otak. Pada baris atas terlihat dua citra *glioma* dengan potongan sagital, sedangkan baris bawah menampilkan dua citra meningioma dengan potongan koronal.

Gambar tersebut sekaligus mengungkap satu hal yang perlu dicatat mengenai dataset ini, yaitu keberagaman bidang potong. Citra pada dataset memuat potongan aksial, sagital, dan koronal secara bercampur tanpa penandaan. Bagi model, hal ini berarti ia harus mempelajari penampakan setiap jenis tumor pada tiga bidang potong yang berbeda sekaligus, dan hal itu secara efektif melipattigakan keragaman yang harus dikuasai dari jumlah data yang tetap. Keberagaman ini menambah tantangan pada dataset yang sudah berukuran kecil.

Perlu ditambahkan catatan mengenai urutan langkah prapemrosesan. Pengubahan ukuran dari 512 × 512 menjadi 224 × 224 dilakukan *sebelum* penerapan CLAHE. Urutan ini menghemat komputasi karena CLAHE bekerja pada citra yang lebih kecil, namun konsekuensinya sebagian detail halus sudah hilang sebelum

kontras diperkuat. Urutan sebaliknya memperkuat kontras pada resolusi penuh terlebih dahulu, baru mengecilkan ukuran akan mempertahankan lebih banyak informasi, dengan biaya komputasi yang lebih besar.

Pembagian Data

Tiga rasio pembagian data diuji sebagaimana dirangkum pada Tabel 7.2.

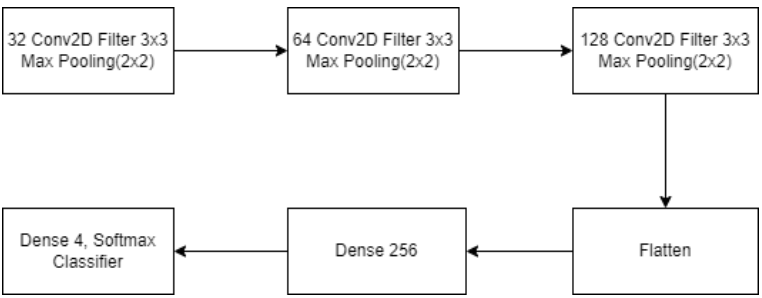
Tabel 7.2 Tiga Rasio Pembagian Data yang Diuji

Rasio	Data Latih	Data Validasi	Data Uji	Citra Uji per Kelas
80 : 10 : 10	1.600	200	200	± 50
70 : 15 : 15	1.400	300	300	± 75
60 : 20 : 20	1.200	400	400	± 100

Kolom terakhir pada tabel tersebut memuat informasi yang akan menjadi penting pada Subbab 7.4.12. Pada rasio 80:10:10, setiap kelas hanya diwakili sekitar 50 citra pada data uji, sehingga satu poin persentase akurasi setara dengan dua citra. Pada rasio 60:20:20, jumlahnya menjadi dua kali lipat sehingga taksirannya lebih stabil. Terdapat pertukaran yang melekat di sini: rasio yang memberikan data latih terbanyak justru memberikan pengukuran yang paling tidak teliti.

Arsitektur Model CNN

Model CNN yang dirancang bersifat sederhana dan dibangun sepenuhnya dari awal tanpa bobot *pretrained*. Arsitekturnya terdiri atas tiga blok konvolusi dan dua lapisan terhubung penuh sebagaimana ditunjukkan pada Gambar 7.9.



Gambar 7.9 Arsitektur Model CNN

Rincian setiap lapisan disajikan pada Tabel 7.3.

Tabel 7.3 Rincian Lapisan Model CNN

Lapisan	**Konfigurasi**	**Keluaran**
Conv2D 1	32 filter, kernel 3×3, ReLU	32 *feature* map
MaxPooling 1	Kernel 2×2	Dimensi menjadi separuh
Conv2D 2	64 filter, kernel 3×3, ReLU	64 *feature* map
MaxPooling 2	Kernel 2×2	Dimensi menjadi separuh
Conv2D 3	128 filter, kernel 3×3, ReLU	128 *feature* map
MaxPooling 3	Kernel 2×2	Dimensi menjadi separuh
Flatten	—	Vektor satu dimensi
Dense 1	256 neuron	Vektor fitur
Dense 2	4 neuron, softmax	Probabilitas empat kelas

Rancangan ini patut memperoleh catatan tersendiri. Tidak terdapat *dropout* maupun *batch normalization* di dalamnya, dan tidak digunakan bobot *pretrained*. Dengan 1.600 citra latih dan lapisan *dense* berisi 256 neuron yang terhubung ke seluruh keluaran *flatten*, jumlah parameter pada lapisan tersebut saja sudah sangat besar. Ketiadaan regularisasi pada rancangan sesederhana ini merupakan kondisi yang secara teori sangat rentan terhadap *overfitting*, dan hal itu akan terlihat langsung pada kurva pelatihan di Subbab 7.4.2.

Arsitektur Model ViT

Model ViT dibangun dengan parameter sebagaimana disajikan pada Tabel 7.4.

Tabel 7.4 Parameter Model *Vision Transformer*

Parameter	**Nilai**	**Keterangan**
image_size	224	Ukuran citra masukan
patch_size	16	Ukuran sisi setiap patch
num_classes	4	Jumlah kelas keluaran
hidden_dim	256	Dimensi representasi setiap patch
mlp_dim	512	Dimensi lapisan MLP pada blok encoder

atten_heads	16	Jumlah head pada *multi-head attention*
encoder_depth	12	Jumlah blok transformer encoder
dropout_rate	0,1	Peluang penonaktifan neuron
activation	softmax	Fungsi aktivasi keluaran

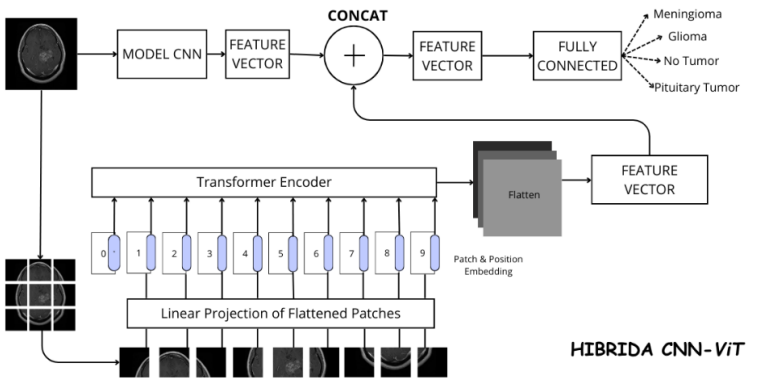
Dengan ukuran citra 224 dan ukuran *patch* 16, jumlah *patch* yang terbentuk berdasarkan Persamaan 7.9 adalah 224 kuadrat dibagi 16 kuadrat, yaitu 196 *patch*. Ditambah satu *class token*, panjang rangkaian yang diproses menjadi 197.

Catatan: Ketidakselarasan Nilai *Patch Size*
Terdapat ketidakselarasan yang perlu diklarifikasi pada laporan yang mendasari bab ini. Tabel parameter mencantumkan *patch_size* sebesar 16, sedangkan uraian penjelasnya menyatakan bahwa ukuran *patch* yang digunakan adalah 32.
Selisih ini bukan perkara kecil. Dengan *patch* berukuran 16, terbentuk 196 *patch*; dengan *patch* berukuran 32, hanya terbentuk 49 *patch*. Karena biaya komputasi *self-attention* tumbuh sebanding dengan kuadrat panjang rangkaian, kedua pilihan tersebut berbeda sekitar enam belas kali lipat dalam beban komputasi.
Perbedaannya juga menyangkut kualitas. *Patch* berukuran 16 menghasilkan resolusi spasial yang jauh lebih halus dan lebih mampu menangkap lesi berukuran kecil, sedangkan *patch* berukuran 32 pada citra 224 piksel berarti setiap *patch* mencakup wilayah yang cukup besar sehingga tumor kecil dapat tenggelam di dalam satu *patch* bersama jaringan sehat di sekitarnya.
Bab ini mengacu pada nilai 16 sebagaimana tercantum pada tabel parameter, namun ketidakselarasan ini perlu diperiksa dan diselaraskan pada penulisan berikutnya, sebab pembaca tidak memiliki cara untuk mengetahui nilai mana yang sesungguhnya dijalankan.

Kedalaman *encoder* sebanyak 12 blok dengan 16 *head* merupakan konfigurasi yang tergolong besar dan setara dengan varian ViT-Base pada makalah aslinya. Konfigurasi tersebut memuat jutaan parameter, dan melatihnya dari awal menggunakan 1.600 citra menempatkan model pada kondisi yang sangat tidak menguntungkan sebagaimana telah diuraikan pada Subbab 7.2.4.

Arsitektur Model Hibrida CNN-ViT

Model hibrida dibangun dengan menjalankan kedua arsitektur secara paralel pada citra yang sama, kemudian menggabungkan vektor fitur yang dihasilkan sebelum lapisan klasifikasi. Rancangannya ditunjukkan pada Gambar 7.10.

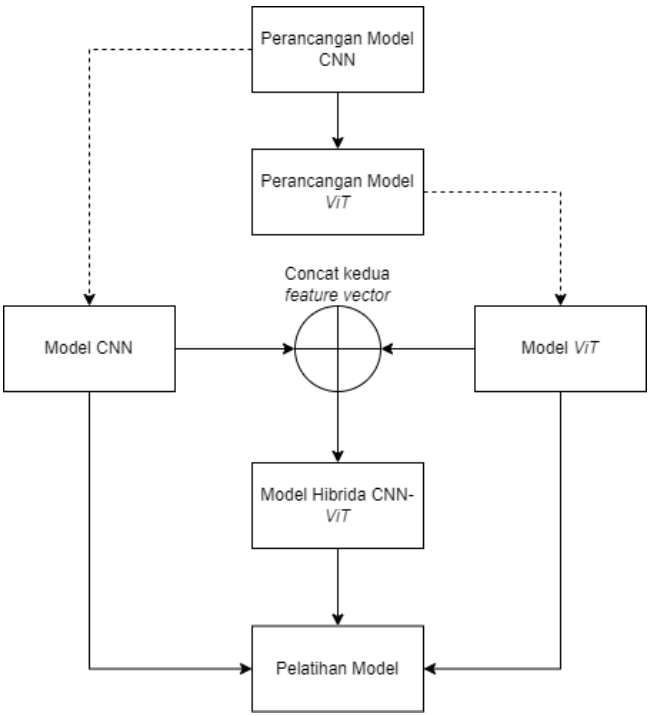


*Ilustrasi Gambar Diatas Belum Melewati *Preprocessing Image*

Gambar 7.10 Arsitektur Model Hibrida CNN-ViT

Citra masukan berukuran 224 × 224 × 3 diproses melalui dua jalur. Jalur atas menjalankan model CNN yang menghasilkan vektor fitur, sedangkan jalur bawah memecah citra menjadi *patch*, memproyeksikannya secara linear, menambahkan *position embedding*, dan meneruskannya melalui *transformer encoder* untuk menghasilkan vektor fitur kedua. Kedua vektor tersebut digabungkan melalui operasi *concatenate* sesuai Persamaan 7.15, kemudian diteruskan ke *fully connected layer* beraktivasi softmax yang menghasilkan probabilitas keempat kelas.

Perlu diperjelas satu hal yang berpotensi menimbulkan salah tafsir. Diagram pada Gambar 7.10 tampak seolah model menerima dua citra masukan yang berbeda, padahal yang terjadi adalah satu citra yang sama disalurkan ke kedua jalur. Alur penggabungan model ditunjukkan pada Gambar 7.11.



Gambar 7.11 Alur Perancangan dan Penggabungan Model

Konfigurasi Pelatihan

Konfigurasi pelatihan yang digunakan disajikan pada Tabel 7.5.

Tabel 7.5 Konfigurasi Pelatihan Ketiga Model

Parameter	**Nilai**	**Catatan**
Fungsi kerugian	*Categorical Crossentropy*	Sesuai klasifikasi empat kelas
Metrik	*Accuracy*	Sah karena data seimbang
Epoch	10	Tetap untuk seluruh skenario
Batch size	32	Tetap untuk seluruh skenario

Optimizer	Adam dan RMSprop	Menjadi variabel pengujian
Learning rate	0,001 dan 0,0001	Menjadi variabel pengujian
Bobot awal	Acak	Tanpa *transfer learning*
Early stopping	Tidak digunakan	Pelatihan berjalan penuh 10 *epoch*

Dua baris terakhir pada tabel tersebut merupakan yang paling menentukan bagi seluruh hasil pada bab ini, dan keduanya akan dibahas kembali secara mendalam pada Subbab 7.4.5. Ketiadaan *early stopping* berarti model yang dievaluasi adalah model pada *epoch* kesepuluh, bukan model dengan performa validasi terbaik. Ketiadaan bobot awal berarti seluruh model, termasuk ViT yang berkapasitas besar, harus mempelajari segalanya dari nol menggunakan 1.600 citra.

Satu catatan penulisan perlu ditambahkan. Nilai *learning rate* pada laporan asli dituliskan sebagai "0,001%" dan "0,0001%". Apabila tanda persen dibaca secara harfiah, nilainya menjadi 0,00001 dan 0,000001 yaitu nilai yang terlalu kecil untuk menghasilkan konvergensi dalam sepuluh *epoch*. Yang dimaksudkan hampir pasti adalah 0,001 dan 0,0001 tanpa tanda persen, dan bab ini menggunakan penafsiran tersebut.

Rancangan Skenario Pengujian

Rancangan pengujian disusun sebagai percobaan faktorial atas empat faktor sebagaimana dirangkum pada Tabel 7.6.

Tabel 7.6 Komposisi Skenario Pengujian

Faktor	**Tingkat**	**Jumlah**
Arsitektur	CNN, ViT, hibrida CNN-ViT	3
Rasio pembagian data	80:10:10, 70:15:15, 60:20:20	3
Algoritma optimasi	Adam, RMSprop	2
Learning rate	0,001 dan 0,0001	2
Total skenario	**3 × 3 × 2 × 2**	**36**
Pengulangan per skenario	**Lima kali, dilaporkan sebagai rerata**	**180 pelatihan**

Pengulangan sebanyak lima kali pada setiap skenario merupakan keputusan metodologis yang patut memperoleh apresiasi khusus, dan merupakan hal yang belum dilakukan pada bab-bab sebelumnya di buku ini. Pelatihan jaringan saraf bersifat stokastik: inisialisasi bobot acak, urutan pengacakan data, dan *dropout* seluruhnya menghasilkan variasi antarjalannya. Melaporkan hasil dari satu kali pelatihan berarti melaporkan satu titik dari suatu sebaran, dan tidak ada cara untuk mengetahui apakah titik itu mewakili atau kebetulan menguntungkan. Perata-rataan atas lima kali pelatihan menghasilkan taksiran yang jauh lebih stabil.

Meskipun demikian, satu hal masih kurang. Yang dilaporkan hanyalah nilai reratanya, sedangkan simpangan bakunya tidak disertakan. Padahal justru simpangan baku itulah yang memungkinkan pembaca menilai apakah selisih antara dua rerata bermakna atau masih berada dalam rentang keragaman antarjalan. Hal ini akan dibahas kembali pada Subbab 7.4.12.

Lingkungan Komputasi

Seluruh skenario dijalankan menggunakan sebuah laptop dengan prosesor Intel Core i5-8265U berkecepatan 1,6 GHz, grafis terintegrasi Intel UHD Graphics 620, dan memori sebesar 12 GB. Pengembangan dilakukan menggunakan Jupyter Notebook dengan bahasa Python.

Spesifikasi tersebut perlu dinyatakan secara terbuka karena menjelaskan banyak keputusan perancangan pada bab ini. Grafis terintegrasi tidak dapat digunakan untuk mempercepat pelatihan jaringan saraf sebagaimana kartu grafis khusus, sehingga seluruh komputasi dijalankan pada prosesor. Selisih kecepatan antara keduanya pada pelatihan jaringan saraf umumnya mencapai puluhan kali lipat.

Catatan: Keterbatasan Perangkat Menjelaskan Anggaran Sepuluh *Epoch*
Penetapan sepuluh *epoch* bagi seluruh skenario, yang telah dipersoalkan pada Subbab 7.3.8, menjadi jauh lebih dapat dipahami dalam konteks ini.

Melatih model ViT dengan dua belas blok *encoder* dan enam belas *head* pada prosesor tanpa akselerasi grafis merupakan pekerjaan yang sangat berat. Dengan 36 skenario yang masing-masing diulang lima kali, jumlah pelatihan yang harus dijalankan mencapai 180 kali. Menaikkan anggaran menjadi lima puluh *epoch* akan melipatlimakan waktu yang dibutuhkan, dan pada perangkat semacam ini hal itu berpotensi memakan waktu berminggu-minggu.

Dengan demikian, keputusan menetapkan sepuluh *epoch* bukanlah kelalaian melainkan pertukaran yang disadari antara kelengkapan rancangan faktorial dan kedalaman pelatihan setiap model. Keputusan untuk mengutamakan kelengkapan 36 skenario dengan lima pengulangan merupakan pilihan yang dapat dibenarkan.

Yang perlu disampaikan bersama hasilnya adalah bahwa pilihan tersebut membatasi jenis kesimpulan yang dapat ditarik. Pada anggaran ini, kesimpulan yang sah menyangkut perbandingan pada kondisi sepuluh *epoch*, bukan perbandingan antararsitektur secara umum.

Jalan keluar yang tersedia: memanfaatkan layanan komputasi awan yang menyediakan akselerasi grafis secara cuma-cuma untuk penggunaan terbatas, atau mengurangi jumlah skenario agar setiap model memperoleh anggaran pelatihan yang memadai. Alternatif kedua ini sering kali lebih baik: dua belas skenario yang dilatih hingga konvergen menjawab pertanyaan Metodologi jauh lebih meyakinkan daripada tiga puluh enam skenario yang seluruhnya dihentikan sebelum waktunya.

➢ Hasil dan Ujicoba

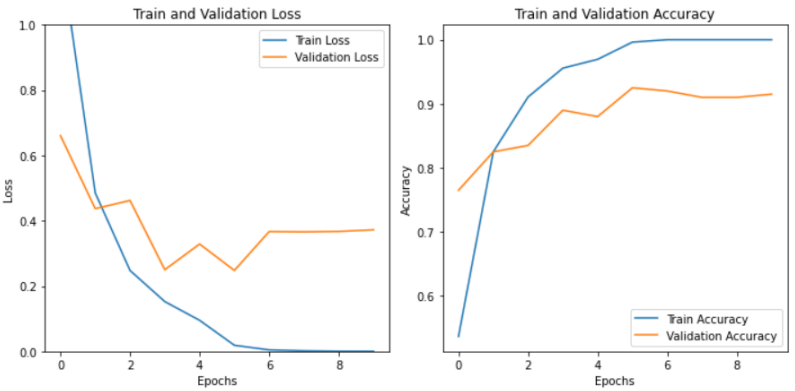
Kondisi Awal Data

Pemeriksaan keseimbangan data mengonfirmasi bahwa keempat kelas memiliki jumlah yang sama, yaitu 500 citra per kelas dengan total 2.000 citra. Prapemrosesan berhasil meningkatkan kontras citra sebagaimana telah ditunjukkan pada Gambar 7.8, sehingga batas antara massa tumor dan jaringan otak menjadi lebih tegas.

Setelah prapemrosesan, seluruh citra dikonversi menjadi larik *NumPy* yang menyimpan nilai piksel, kanal warna, dan label kelasnya dalam bentuk larik multidimensi, siap diproses oleh ketiga model.

Kurva Pelatihan Model CNN

Perkembangan pelatihan model CNN pada salah satu skenario ditunjukkan pada Gambar 7.12.



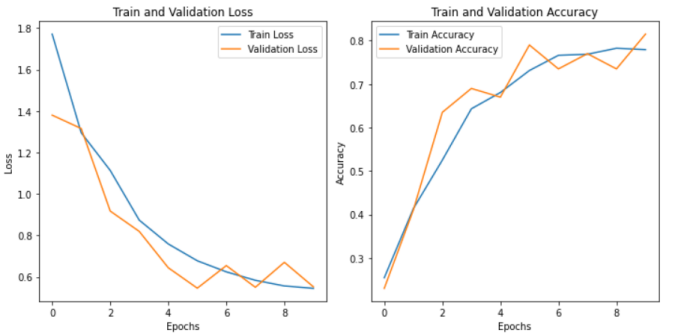
Gambar 7.12 Kurva Pelatihan Model CNN

Kurva tersebut memperlihatkan pola yang sangat khas dan perlu dibaca dengan cermat. Pada panel kanan, akurasi pelatihan menaik dengan cepat dan mencapai 1,00 pada *epoch* keenam, kemudian bertahan pada nilai tersebut hingga akhir. Akurasi validasi menaik hingga sekitar 0,92 pada *epoch* kelima lalu mendatar. Pada panel kiri, *training loss* turun mendekati nol, sedangkan *validation loss* mencapai titik terendahnya sekitar 0,25 pada *epoch* kelima kemudian justru menaik hingga sekitar 0,37 pada akhir pelatihan.

Akurasi pelatihan yang mencapai 1,00 dengan *loss* mendekati nol berarti model telah menghafal seluruh 1.600 citra latih secara sempurna. Sejak titik itu, tidak ada lagi yang dapat dipelajari model dari data latih; setiap iterasi selanjutnya hanya mempertegas hafalan tersebut. Menaiknya *validation loss* setelah *epoch* kelima merupakan tanda *overfitting* yang tidak dapat disangkal.

Kurva Pelatihan Model ViT

Perkembangan pelatihan model ViT ditunjukkan pada Gambar 7.13.



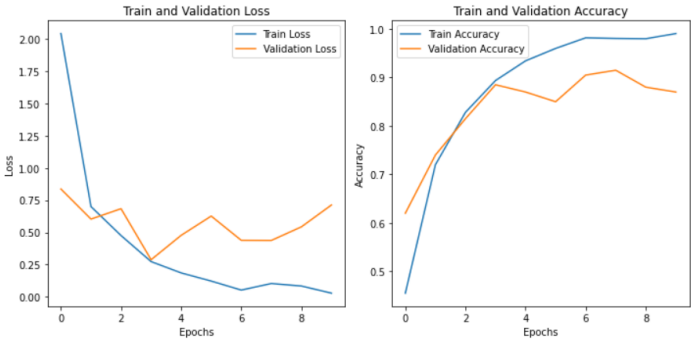
Gambar 7.13 Kurva Pelatihan Model ViT

Polanya sepenuhnya berbeda dari CNN. Akurasi pelatihan dan validasi bergerak nyaris berdampingan sepanjang sepuluh *epoch*, keduanya masih menaik ketika pelatihan berakhir pada kisaran 0,78 sampai 0,82. Nilai *loss* keduanya masih menurun dan belum menunjukkan tanda mendatar sama sekali.

Bacaan atas kurva ini tegas: model ViT belum selesai belajar. Tidak terdapat satu pun tanda *overfitting* jarak antara kurva pelatihan dan validasi justru sangat rapat, yang menandakan model belum sempat menghafal apa pun. Yang terjadi adalah kebalikannya, yaitu *underfitting*: model dihentikan ketika ia masih berada di tengah proses belajar.

Kurva Pelatihan Model Hibrida

Perkembangan pelatihan model hibrida ditunjukkan pada Gambar 7.14.



Gambar 7.14 Kurva Pelatihan Model Hibrida CNN-ViT

Polanya menyerupai CNN namun sedikit lebih terkendali. Akurasi pelatihan mencapai sekitar 0,99 pada *epoch* keenam, sedangkan akurasi validasi memuncak pada sekitar 0,91 pada *epoch* ketujuh lalu turun sedikit. Nilai *validation loss* mencapai titik terendahnya sekitar 0,27 pada *epoch* ketiga, kemudian menaik hingga sekitar 0,72 pada akhir pelatihan kenaikan hampir tiga kali lipat.

Model hibrida karena itu juga mengalami *overfitting*, bahkan lebih dini daripada CNN. Hal ini masuk akal mengingat kapasitasnya lebih besar: ia memuat seluruh parameter CNN ditambah seluruh parameter ViT.

Tiga Model, Tiga Kondisi Belajar yang Berbeda

Bagian ini menyatukan pengamatan atas ketiga kurva pelatihan, dan merupakan bagian terpenting pada keseluruhan bab ini. Rangkuman kondisi ketiga model pada *epoch* kesepuluh disajikan pada Tabel 7.7.

Tabel 7.7 Kondisi Ketiga Model pada Akhir Pelatihan

Model	**Akurasi Latih**	**Akurasi Validasi**	**Arah *Validation Loss***	**Kondisi**
CNN	± 1,00	± 0,92	Menaik sejak *epoch* 5	*Overfitting*
ViT	± 0,78	± 0,81	Masih menurun	Underfitting
Hibrida CNN-ViT	± 0,99	± 0,87	Menaik sejak *epoch* 3	*Overfitting*

Perhatian: Perbandingan Antararsitektur pada Anggaran *Epoch* yang Sama

Menetapkan jumlah *epoch* yang sama bagi ketiga model tampak seperti keputusan yang adil. Tabel 7.7 menunjukkan bahwa kenyataannya justru sebaliknya.

Pada *epoch* kesepuluh, CNN dan hibrida sudah *melewati* titik terbaiknya dan sedang memburuk, sedangkan ViT *belum mencapai* titik terbaiknya dan masih membaik. Ketiganya diukur

pada saat yang sama menurut hitungan *epoch*, tetapi pada tahap yang sama sekali berbeda menurut kemajuan belajarnya.
Analoginya seperti menilai tiga pelari pada detik yang sama, padahal yang pertama sudah melewati garis akhir dan mulai berjalan santai, yang kedua baru menempuh separuh lintasan, dan yang ketiga sudah selesai lalu berbalik arah. Angka yang tercatat pada detik itu memang benar, tetapi tidak menjawab pertanyaan siapa yang tercepat.
Kesimpulan yang sah dari eksperimen ini adalah bahwa pada anggaran sepuluh *epoch* tanpa bobot awal, hibrida CNN-ViT mencapai akurasi tertinggi. *Kesimpulan yang tidak sah* adalah bahwa ViT merupakan arsitektur yang lebih lemah untuk permasalahan ini sebab ViT tidak pernah diberi kesempatan menyelesaikan proses belajarnya.
Perbaikan yang disarankan: latih setiap model hingga *validation loss*-nya benar-benar mendatar, gunakan *EarlyStopping* untuk menentukan titik berhentinya secara otomatis, dan gunakan *ModelCheckpoint* untuk menyimpan bobot pada titik validasi terbaik. Ketiganya tidak menuntut perubahan arsitektur apa pun, hanya beberapa baris tambahan pada kode pelatihan.

Terdapat satu konsekuensi lanjutan yang perlu dinyatakan. Karena *ModelCheckpoint* tidak digunakan, model yang dievaluasi pada seluruh skenario adalah model pada *epoch* kesepuluh. Bagi CNN dan hibrida, *epoch* tersebut merupakan titik dengan *validation loss* terburuk sepanjang pelatihan. Dengan kata lain, angka akurasi yang dilaporkan pada Subbab 7.4.6 sesungguhnya berasal dari kondisi model yang paling buruk secara validasi dan penambahan *ModelCheckpoint* saja berpeluang menaikkan seluruh angka tersebut tanpa mengubah apa pun pada rancangan.

Hasil Pengujian Seluruh Skenario

Hasil pengujian model CNN pada dua belas skenario disajikan pada Tabel 7.8. Setiap nilai merupakan rerata atas lima kali pelatihan.

Tabel 7.8 Hasil Pengujian Skenario Model CNN

Pembagian Data	**RMSprop 0,001**	**RMSprop 0,0001**	**Adam 0,001**	**Adam 0,0001**
80 : 10 : 10	**90,80%**	**88,00%**	**89,90%**	**88,80%**
70 : 15 : 15	82,73%	84,20%	87,40%	87,07%
60 : 20 : 20	82,90%	83,45%	86,05%	85,85%

Akurasi tertinggi sebesar 90,80% diperoleh pada rasio 80:10:10 dengan optimasi RMSprop dan *learning rate* 0,001. Hasil pengujian model ViT disajikan pada Tabel 7.9.

Tabel 7.9 Hasil Pengujian Skenario Model ViT

Pembagian Data	**RMSprop 0,001**	**RMSprop 0,0001**	**Adam 0,001**	**Adam 0,0001**
80 : 10 : 10	**77,80%**	**82,90%**	**75,40%**	**84,80%**
70 : 15 : 15	71,93%	79,60%	72,53%	78,93%
60 : 20 : 20	69,94%	80,15%	67,70%	80,95%

Akurasi tertinggi sebesar 84,80% diperoleh pada rasio 80:10:10 dengan optimasi Adam dan *learning rate* 0,0001. Hasil pengujian model hibrida disajikan pada Tabel 7.10.

Tabel 7.10 Hasil Pengujian Skenario Model Hibrida CNN-ViT

Pembagian Data	**RMSprop 0,001**	**RMSprop 0,0001**	**Adam 0,001**	**Adam 0,0001**
80 : 10 : 10	**88,70%**	**87,70%**	**93,00%**	**90,70%**
70 : 15 : 15	86,20%	86,20%	88,27%	87,13%
60 : 20 : 20	85,50%	83,85%	84,50%	85,55%

Akurasi tertinggi sebesar 93,00% diperoleh pada rasio 80:10:10 dengan optimasi Adam dan *learning rate* 0,001. Rangkuman akurasi terbaik ketiga model disajikan pada Tabel 7.11.

Tabel 7.11 Rangkuman Akurasi Terbaik Ketiga Model

Model	**Akurasi Terbaik**	**Konfigurasi**	**Selisih terhadap ViT**
Hibrida CNN-ViT	**93,00%**	**Adam, lr 0,001, 80:10:10**	**+8,20**
CNN	90,80%	RMSprop, lr 0,001, 80:10:10	+6,00
ViT	84,80%	Adam, lr 0,0001, 80:10:10	—

Perbandingan langsung ketiga arsitektur pada setiap pasangan skenario yang bersesuaian disajikan pada Tabel 7.12. Tabel ini menempatkan angka dari ketiga tabel sebelumnya secara berdampingan sehingga pola keunggulannya dapat diamati sekaligus.

Tabel 7.12 Perbandingan Ketiga Arsitektur pada Setiap Skenario

Skenario	**CNN**	**ViT**	**Hibrida**	**Unggul**
80:10:10, RMSprop 0,001	90,80%	77,80%	88,70%	CNN
80:10:10, RMSprop 0,0001	88,00%	82,90%	87,70%	CNN
80:10:10, Adam 0,001	89,90%	75,40%	93,00%	Hibrida
80:10:10, Adam 0,0001	88,80%	84,80%	90,70%	Hibrida
70:15:15, RMSprop 0,001	82,73%	71,93%	86,20%	Hibrida
70:15:15, RMSprop 0,0001	84,20%	79,60%	86,20%	Hibrida
70:15:15, Adam 0,001	87,40%	72,53%	88,27%	Hibrida
70:15:15, Adam 0,0001	87,07%	78,93%	87,13%	Hibrida
60:20:20, RMSprop 0,001	82,90%	69,94%	85,50%	Hibrida
60:20:20, RMSprop 0,0001	83,45%	80,15%	83,85%	Hibrida
60:20:20, Adam 0,001	86,05%	67,70%	84,50%	CNN
60:20:20, Adam 0,0001	85,85%	80,95%	85,55%	CNN

Tabel tersebut menyampaikan tiga hal yang tidak terlihat pada tabel-tabel sebelumnya. Yang pertama, model ViT berada pada peringkat terbawah pada dua belas dari dua belas skenario tanpa terkecuali. Konsistensi mutlak semacam ini merupakan bukti yang sangat kuat jauh lebih kuat daripada selisih rerata mana pun bahwa pada kondisi eksperimen ini ViT memang tertinggal. Perlu diingat kembali bahwa kesimpulan tersebut menyangkut kondisi eksperimennya, bukan arsitekturnya, sebagaimana telah diuraikan pada Subbab 7.2.4.

Yang kedua, model hibrida mengungguli CNN pada delapan dari dua belas skenario. Angka ini lebih baik daripada kebetulan namun jauh dari mutlak. Menariknya, keempat skenario tempat CNN unggul tersebar pada kedua ujung dua pada rasio 80:10:10 dengan RMSprop dan dua pada rasio 60:20:20 dengan Adam sehingga tidak membentuk pola yang jelas.

Yang ketiga, selisih antara hibrida dan CNN pada sebagian besar skenario tergolong sangat kecil. Pada skenario 70:15:15 dengan Adam dan *learning rate* 0,0001, selisihnya hanya 0,06 poin persentase setara kurang dari satu citra pada himpunan uji berisi 300 citra. Selisih sekecil itu tidak menyampaikan informasi apa pun.

Pengaruh Rasio Pembagian Data

Satu pola muncul secara sangat konsisten pada ketiga tabel hasil: rasio 80:10:10 menghasilkan akurasi tertinggi pada seluruh model dan seluruh kombinasi optimasi tanpa terkecuali. Rangkuman pengaruhnya disajikan pada Tabel 7.13.

Tabel 7.13 Rerata Akurasi menurut Rasio Pembagian Data

Rasio	Rerata CNN	Rerata ViT	Rerata Hibrida	Data Latih
80 : 10 : 10	89,38%	80,23%	90,03%	1.600
70 : 15 : 15	85,35%	75,75%	86,95%	1.400
60 : 20 : 20	84,56%	74,69%	84,85%	1.200

Penurunan akurasi seiring mengecilnya proporsi data latih merupakan hal yang diperkirakan dan konsisten dengan seluruh teori. Yang menarik adalah besarnya penurunan tersebut. Beralih dari 1.600 menjadi 1.200 citra latih pengurangan sebesar 25%

menurunkan akurasi sekitar lima poin persentase pada ketiga model. Kepekaan sebesar itu terhadap jumlah data latih merupakan tanda yang jelas bahwa dataset ini terlalu kecil bagi kapasitas model yang digunakan; pada dataset yang memadai, pengurangan seperempat data tidak akan berdampak sebesar itu.

Perlu ditambahkan satu pertimbangan yang jarang dinyatakan. Rasio 80:10:10 memang menghasilkan akurasi tertinggi, tetapi juga memiliki data uji paling sedikit sehingga taksirannya paling tidak teliti. Sebaliknya, rasio 60:20:20 menghasilkan akurasi terendah namun taksirannya paling stabil karena diukur pada 400 citra. Ketika sebuah hasil "terbaik" kebetulan juga merupakan hasil yang paling tidak teliti pengukurannya, terdapat alasan untuk berhati-hati.

Pengaruh Algoritma Optimasi dan *Learning Rate*

Berbeda dengan rasio pembagian data yang polanya konsisten, pengaruh algoritma optimasi justru berbeda-beda menurut arsitekturnya sebagaimana dirangkum pada Tabel 7.14.

Tabel 7.14 Kombinasi Optimasi Terbaik pada Setiap Arsitektur

Model	**Optimasi Terbaik**	***Learning Rate***	**Akurasi**
CNN	RMSprop	0,001	90,80%
ViT	Adam	0,0001	84,80%
Hibrida CNN-ViT	Adam	0,001	93,00%

Ketiadaan pola tunggal ini sesungguhnya merupakan temuan tersendiri: tidak terdapat kombinasi optimasi yang unggul secara universal, sehingga penalaan perlu diulang setiap kali arsitektur diganti. Kesimpulan yang sama telah muncul pada Bab 1 mengenai *learning rate* dan pada Bab 4 mengenai kesetaraan konfigurasi.

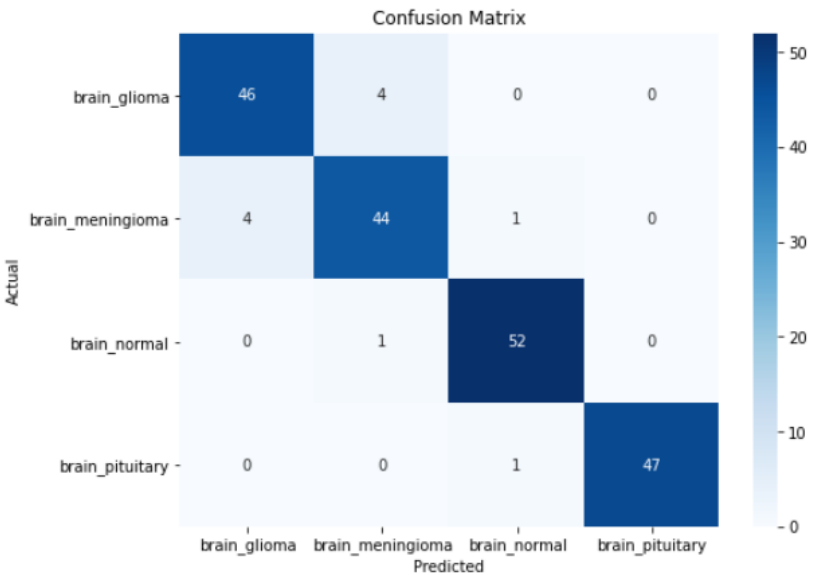
Satu pola pada model ViT layak *dice*rmati lebih dekat. Pada Tabel 7.9, *learning rate* 0,0001 secara konsisten mengungguli 0,001 pada seluruh kombinasi rasio dan optimasi selisihnya bahkan mencapai lebih dari sepuluh poin persentase pada beberapa sel. Pola ini berlawanan dengan CNN dan hibrida yang justru lebih menyukai nilai yang lebih besar.

Penjelasannya berkaitan dengan sifat arsitektur *transformer*. Blok *self-attention* memuat operasi softmax atas hasil perkalian titik yang sangat peka terhadap besarnya nilai masukan; langkah

pembaruan yang terlalu besar dapat mendorong nilai *logit* ke wilayah ekstrem tempat *gradien softmax* nyaris hilang. Inilah sebabnya pelatihan *transformer* pada praktiknya lazim menggunakan *learning rate* kecil disertai pemanasan bertahap pada awal pelatihan teknik yang dikenal sebagai *warmup* dan tidak digunakan pada bab ini.

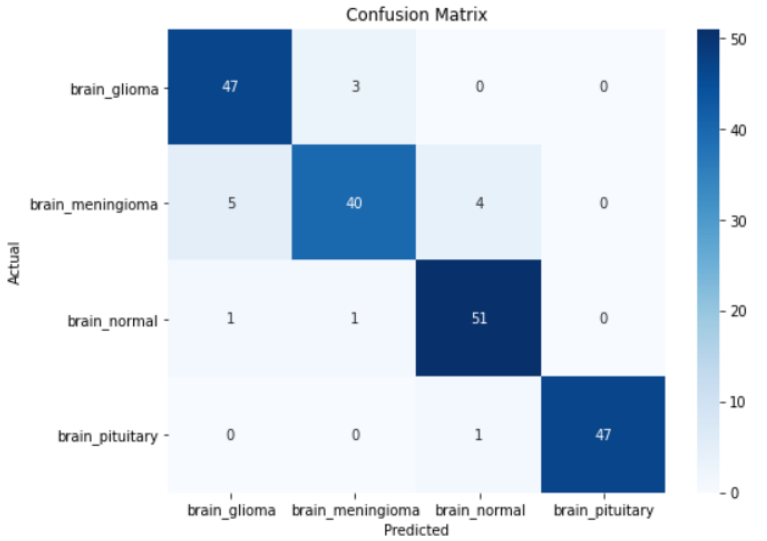
Confusion Matrix Ketiga Model

Evaluasi dilanjutkan dengan pemeriksaan *confusion matrix* pada skenario terbaik masing-masing model. Hasil model hibrida ditunjukkan pada Gambar 7.15.



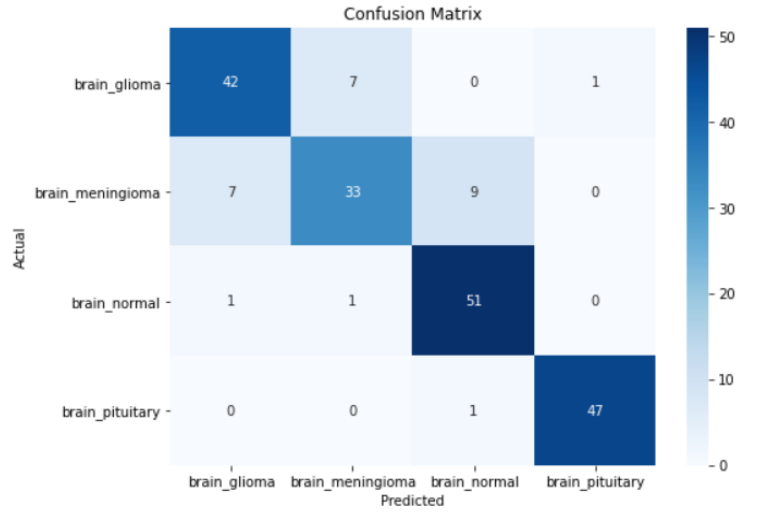
Gambar 7.15 *Confusion Matrix* Model Hibrida CNN-ViT

Model hibrida mengklasifikasikan dengan benar 189 dari 200 citra uji, setara akurasi 94,5%. Hasil model CNN ditunjukkan pada Gambar 7.16.



Gambar 7.16 *Confusion Matrix* Model CNN

Model CNN mengklasifikasikan dengan benar 185 dari 200 citra uji, setara akurasi 92,5%. Hasil model ViT ditunjukkan pada Gambar 7.17.



Gambar 7.17 *Confusion Matrix* Model ViT

Model ViT mengklasifikasikan dengan benar 173 dari 200 citra uji, setara akurasi 86,5%.

Analisis Pola Kekeliruan

Angka akurasi keseluruhan menyembunyikan informasi yang jauh lebih berguna, yaitu pola kekeliruannya. Rincian kekeliruan ketiga model disajikan pada Tabel 7.15.

Tabel 7.15 Rincian Pola Kekeliruan Ketiga Model

Jenis Kekeliruan	**Hibrida**	**CNN**	**ViT**
Glioma diprediksi meningioma	4	3	7
Meningioma diprediksi glioma	4	5	7
Meningioma diprediksi tanpa tumor	1	4	9
Glioma diprediksi tanpa tumor	0	0	0
Tanpa tumor diprediksi ada tumor	1	2	2
Pituitari diprediksi tanpa tumor	1	1	1
Kekeliruan lainnya	0	0	1
Total kekeliruan	**11**	**15**	**27**

Dua pola muncul dengan sangat jelas dari tabel tersebut. Pola pertama, kekeliruan antara glioma dan meningioma mendominasi pada ketiga model: delapan dari sebelas kekeliruan pada hibrida, delapan dari lima belas pada CNN, dan empat belas dari dua puluh tujuh pada ViT. Kenyataan bahwa tiga arsitektur yang sangat berbeda melakukan kekeliruan pada pasangan kelas yang sama merupakan petunjuk kuat bahwa penyebabnya tidak terletak pada arsitektur, melainkan pada kemiripan yang memang ada di dalam data.

Pola kedua bersifat lebih menentukan secara klinis, dan diuraikan pada kotak berikut.

Kajian Utama: Keunggulan Hibrida Terpusat pada Kekeliruan Paling Berbahaya

Perhatikan baris ketiga pada Tabel 7.15, yaitu kasus meningioma yang diprediksi sebagai tanpa tumor. ViT melakukan kekeliruan ini sebanyak 9 kali, CNN sebanyak 4 kali, dan hibrida hanya 1 kali.

Inilah jenis kekeliruan yang paling merugikan pada keseluruhan sistem. Menyatakan otak yang mengandung tumor sebagai normal berarti pasien dipulangkan tanpa penanganan apa pun. Berbeda dengan kekeliruan antara glioma dan meningioma yang

setidaknya masih mengarahkan pasien pada pemeriksaan lanjutan, kekeliruan ini menghentikan seluruh alur diagnostik.
Yang menarik adalah bahwa keunggulan model hibrida ternyata *terpusat* pada jenis kekeliruan ini. Selisih total kekeliruan antara CNN dan hibrida hanya empat kasus, dan tiga di antaranya berasal dari baris ini. Dengan kata lain, sumbangan utama model hibrida bukan sekadar menaikkan akurasi keseluruhan sebesar dua poin, melainkan menekan kekeliruan yang paling berbahaya dari empat kasus menjadi satu kasus.
Temuan ini memiliki penjelasan yang masuk akal. Mengenali keberadaan massa tumor menuntut perbandingan antara satu bagian otak dan bagian lainnya apakah terdapat asimetri, apakah struktur terdesak dari posisi seharusnya. Perbandingan berjarak jauh semacam itu merupakan hal yang tepat dikerjakan mekanisme *self-attention*, sedangkan CNN dengan kernel kecilnya harus membangunnya secara bertahap melalui banyak lapisan.
Catatan kehati-hatian: seluruh angka pada tabel ini berasal dari satu kali pelatihan pada 200 citra uji. Selisih tiga kasus tidak dapat dinyatakan bermakna secara statistik tanpa pengulangan. Yang dapat dinyatakan adalah bahwa pola ini konsisten dengan penjelasan teoretis dan layak diverifikasi melalui pengulangan yang melaporkan *recall* setiap kelas, bukan hanya akurasi keseluruhan.

Terdapat pula satu pola yang justru muncul melalui ketiadaannya, dan hal itu patut dicatat. Pada baris keempat Tabel 7.15 terlihat bahwa tidak satu pun dari ketiga model pernah memprediksi glioma sebagai tanpa tumor nilainya nol pada ketiganya. Padahal meningioma keliru dinyatakan tanpa tumor sebanyak satu sampai sembilan kali bergantung modelnya.

Ketidaksimetrisan ini memiliki penjelasan radiologis yang masuk akal. Glioma bersifat menyusup ke dalam jaringan otak dan umumnya disertai edema di sekitarnya, sehingga menimbulkan perubahan yang meluas dan tampak jelas pada citra. Meningioma sebaliknya berbatas tegas dan berada di luar jaringan otak; ketika ukurannya kecil dan letaknya di tepi, ia dapat menyerupai penebalan

selaput otak yang normal dan nyaris tidak mengubah penampakan jaringan otak di sekitarnya.

Konsekuensinya bagi penerapan bersifat langsung. Apabila sistem semacam ini digunakan sebagai penapisan awal, kelas meningioma memerlukan kewaspadaan khusus, dan pelaporan *recall* per kelas menjadi jauh lebih penting daripada pelaporan akurasi keseluruhan. Hal ini sejalan dengan prinsip yang telah berulang kali muncul sepanjang buku ini, yaitu bahwa angka rerata menyembunyikan justru informasi yang paling dibutuhkan klinisi.

Classification Report Skenario Terbaik

Metrik rinci per kelas untuk model hibrida disajikan pada Tabel 7.16.

Tabel 7.16 *Classification Report* Model Hibrida CNN-ViT

Kelas	***Precision***	***Recall***	**F1-Score**	***Support***
Glioma	0,92	0,92	0,91	50
Meningioma	0,90	0,90	0,90	49
Tanpa tumor	0,96	0,98	0,97	53
Pituitari	1,00	0,98	0,99	48
Accuracy	—	—	**0,94**	**200**
Rerata makro	**0,95**	**0,94**	**0,94**	**200**

Metrik model CNN disajikan pada Tabel 7.17.

Tabel 7.17 *Classification Report* Model CNN

Kelas	***Precision***	***Recall***	**F1-Score**	***Support***
Glioma	0,89	0,94	0,91	50
Meningioma	0,91	0,82	0,86	49
Tanpa tumor	0,91	0,96	0,94	53
Pituitari	1,00	0,98	0,99	48
Accuracy	—	—	**0,93**	**200**
Rerata makro	**0,93**	**0,92**	**0,92**	**200**

Metrik model ViT disajikan pada Tabel 7.18.

Tabel 7.18 *Classification Report* Model ViT

Kelas	***Precision***	***Recall***	**F1-Score**	***Support***
Glioma	0,84	0,84	0,84	50
Meningioma	0,80	0,67	0,73	49
Tanpa tumor	0,84	0,96	0,89	53
Pituitari	0,98	0,98	0,98	48
Accuracy	—	—	**0,86**	**200**
Rerata makro	**0,87**	**0,86**	**0,86**	**200**

Pembacaan ketiga tabel secara berdampingan mengungkap tiga hal. Pertama, kelas pituitari dikenali nyaris sempurna oleh ketiga model dengan *F1-score* antara 0,98 dan 0,99. Hal ini masuk akal secara anatomis: tumor pituitari selalu berada pada lokasi yang sama, yaitu sella tursika di dasar otak, sehingga posisinya sendiri sudah merupakan penanda yang sangat kuat dan mudah dipelajari.

Kedua, kelas meningioma merupakan yang paling sulit bagi ketiga model, dengan *F1-score* terendah pada masing-masing. Penurunan yang paling tajam terjadi pada ViT dengan *recall* hanya 0,67 yang berarti sepertiga kasus meningioma tidak dikenali. Ketiga, perbedaan antarmodel terpusat justru pada kelas meningioma ini: *recall*-nya sebesar 0,90 pada hibrida, 0,82 pada CNN, dan 0,67 pada ViT. Pada ketiga kelas lainnya, ketiga model berperforma jauh lebih berdekatan.

Apakah Selisihnya Bermakna?

Bagian ini memeriksa kesahihan kesimpulan utama Metodologi, yaitu bahwa model hibrida mengungguli CNN. Pemeriksaan dimulai dari ukuran himpunan uji.

Pada rasio 80:10:10, himpunan uji memuat 200 citra. Model hibrida benar pada 189 citra dan CNN benar pada 185 citra, sehingga selisihnya hanya empat citra. Satu poin persentase pada himpunan sebesar ini setara dengan dua citra. Selisih empat citra karena itu tidak dapat dinyatakan bermakna berdasarkan satu kali pengujian.

Perbandingan yang lebih layak dipercaya adalah rerata atas lima kali pelatihan sebagaimana disajikan pada Tabel 7.11, yaitu 93,00% berbanding 90,80% dengan selisih 2,20 poin persentase atau

setara sekitar empat setengah citra. Pengulangan lima kali jelas lebih kuat daripada satu kali, namun satu hal masih kurang untuk menuntaskan pertanyaan ini.

Yang Kurang: Simpangan Baku Antarjalan Pelatihan

Untuk menilai apakah selisih 2,20 poin persentase bermakna, dibutuhkan satu angka yang tidak dilaporkan, yaitu simpangan baku akurasi di antara kelima jalan pelatihan pada setiap skenario.

Alasannya dapat dipahami melalui dua kemungkinan. *Apabila* kelima jalan pada skenario hibrida menghasilkan akurasi antara 92,5% dan 93,5% dan kelima jalan pada skenario CNN menghasilkan antara 90,5% dan 91,0%, maka kedua sebaran tidak bersinggungan sama sekali dan selisihnya jelas bermakna. *Apabila* kelima jalan pada hibrida menghasilkan antara 89% dan 97% dan pada CNN antara 87% dan 95%, maka kedua sebaran bertumpang tindih hampir seluruhnya dan selisih reratanya tidak menyatakan apa pun.

Kedua kemungkinan tersebut menghasilkan rerata yang sama persis, namun kesimpulan yang berlawanan. Tanpa simpangan baku, pembaca tidak memiliki cara untuk membedakan keduanya.

Perbaikan yang sangat murah: data kelima jalan tersebut sudah tersedia pada lampiran laporan. Yang diperlukan hanyalah menghitung simpangan bakunya dan melaporkannya berdampingan dengan rerata dalam bentuk "93,00% ± x". Uji-t berpasangan atas kelima jalan bahkan akan memberikan jawaban yang tegas mengenai kebermaknaan selisihnya.

Perlu ditegaskan bahwa catatan ini tidak menyatakan kesimpulan Metodologi keliru. Sangat mungkin selisih tersebut memang bermakna pola kekeliruan pada Tabel 7.15 dan penjelasan teoretis mengenai penggabungan fitur keduanya mendukung arah kesimpulan itu. Yang dinyatakan adalah bahwa bukti yang tersedia belum cukup untuk menuntaskannya, padahal bukti yang dibutuhkan sudah ada di tangan.

Apresiasi: Pengulangan Lima Kali Merupakan Praktik yang Patut Ditiru

Di antara ketujuh bab pada buku ini, bab inilah satu-satunya yang mengulang setiap skenario sebanyak lima kali dan melaporkan reratanya. Praktik ini patut memperoleh perhatian tersendiri karena sering diabaikan.

Pelatihan jaringan saraf bersifat stokastik pada beberapa titik sekaligus: inisialisasi bobot dibangkitkan secara acak, urutan sampel diacak pada setiap *epoch*, dan *dropout* menonaktifkan neuron secara acak. Akibatnya, dua kali pelatihan pada kode dan data yang sama persis dapat menghasilkan akurasi yang berbeda beberapa poin persentase.

Melaporkan hasil dari satu kali pelatihan berarti melaporkan satu titik dari suatu sebaran tanpa mengetahui lebar sebaran tersebut. Pada Bab 4 misalnya, seluruh angka berasal dari satu kali pelatihan, sehingga selisih antarmodel di sana sesungguhnya menanggung ketidakpastian yang tidak terukur.

Bab ini menghindari persoalan tersebut dengan mengulang 36 skenario masing-masing lima kali yaitu 180 kali pelatihan pada perangkat tanpa akselerasi grafis. Upaya sebesar itu patut diapresiasi, dan satu-satunya yang kurang adalah melaporkan simpangan bakunya, yang sesungguhnya hanya berjarak satu perhitungan dari data yang sudah dimiliki.

Terdapat pula pemeriksaan sederhana yang dapat memperkuat kesimpulan tanpa perhitungan statistik apa pun, yaitu memeriksa konsistensi lintas skenario. Pada Tabel 7.8 dan Tabel 7.10, model hibrida mengungguli CNN pada delapan dari dua belas pasangan skenario yang bersesuaian. Angka tersebut lebih baik daripada kebetulan namun belum meyakinkan; keunggulan yang benar-benar kokoh semestinya tampak pada hampir seluruh pasangan, sebagaimana keunggulan rasio 80:10:10 yang tampak pada dua belas dari dua belas pasangan.

Pembahasan dan Keterbatasan

Hasil pada bab ini perlu dibaca bersama sejumlah keterbatasan berikut, yang dikelompokkan menurut tahapannya.

Keterbatasan pada data:

a. Jumlah data hanya 2.000 citra, yaitu ukuran yang sangat kecil terutama bagi *Vision Transformer* yang menurut makalah aslinya baru mengungguli CNN pada dataset berukuran belasan juta citra.
b. Sebanyak 1.264 citra dibuang demi menyeimbangkan kelas, padahal ketimpangan aslinya tergolong ringan dan dapat ditangani melalui pembobotan kelas tanpa membuang data.
c. Dataset memuat bidang potong aksial, sagital, dan koronal secara bercampur tanpa penandaan, sehingga keragaman yang harus dikuasai model bertambah secara berarti.
d. Mutu label pada kelas glioma perlu diverifikasi, mengingat catatan pada dataset turunan serta kenyataan bahwa kekeliruan glioma dan meningioma mendominasi hasil ketiga model.

Keterbatasan pada pelatihan:

a. Jumlah *epoch* ditetapkan sama sebanyak sepuluh bagi ketiga arsitektur, padahal ketiganya berada pada kondisi belajar yang sangat berbeda pada titik tersebut.
b. Tidak digunakan *EarlyStopping* maupun *ModelCheckpoint*, sehingga model yang dievaluasi merupakan model pada *epoch* terakhir yang bagi CNN dan hibrida justru merupakan titik dengan *validation loss* terburuk.
c. Ketiga model dilatih dari awal tanpa bobot *pretrained*, yang secara khusus sangat merugikan *Vision Transformer*.
d. Model CNN tidak memuat *dropout* maupun *batch normalization*, sehingga tidak terdapat mekanisme apa pun untuk menahan *overfitting*.
e. Tidak digunakan augmentasi data, padahal augmentasi merupakan penangkal *overfitting* yang paling murah pada dataset berukuran kecil.
f. Pelatihan *transformer* tidak disertai *learning rate warmup* yang lazim digunakan untuk menstabilkan iterasi awal.

Keterbatasan pada evaluasi:

a. Himpunan uji pada konfigurasi terbaik hanya memuat 200 citra, sehingga satu poin persentase setara dengan dua citra dan selisih antarmodel sulit dibedakan dari keacakan.
b. Simpangan baku antarjalan pelatihan tidak dilaporkan, padahal datanya sudah tersedia pada lampiran.
c. Terdapat ketidakselarasan antara nilai *patch_size* pada tabel parameter dan pada uraian penjelasnya.
d. Penulisan nilai *learning rate* menggunakan tanda persen yang apabila dibaca harfiah menghasilkan nilai seribu kali lebih kecil daripada yang dimaksudkan.

Arah pengembangan disusun berikut ini menurut perbandingan antara dampak dan biayanya. Kelompok pertama tidak menuntut data tambahan dan hanya menambahkan beberapa baris kode: menerapkan *EarlyStopping* dan *ModelCheckpoint*, melatih setiap model hingga *validation loss*-nya mendatar alih-alih menetapkan sepuluh *epoch* bagi semuanya, serta melaporkan simpangan baku berdampingan dengan rerata.

Kelompok kedua menuntut pelatihan ulang namun tetap tanpa data tambahan: menggunakan ViT yang telah dilatih pada *ImageNet*-21k alih-alih melatihnya dari awal, menambahkan augmentasi data berupa rotasi kecil dan pembalikan horizontal yang sah secara anatomis, menyisipkan *dropout* dan *batch normalization* pada model CNN, menerapkan *learning rate warmup* pada jalur *transformer*, serta memanfaatkan kembali 1.264 citra yang sebelumnya dibuang disertai pembobotan kelas.

Kelompok ketiga menuntut upaya yang lebih besar: memverifikasi mutu label pada kelas glioma bersama ahli radiologi, memisahkan atau menandai bidang potong agar model tidak perlu mempelajari ketiganya sekaligus, menerapkan validasi silang alih-alih pembagian tunggal mengingat kecilnya ukuran data, serta menambahkan metode interpretasi seperti *Grad-CAM* maupun visualisasi peta perhatian untuk memeriksa apakah kedua jalur pada model hibrida benar-benar memperhatikan area yang relevan.

➢ Ringkasan

Bab ini telah menguraikan perbandingan tiga arsitektur pada permasalahan klasifikasi empat kelas tumor otak berbasis citra MRI melalui tiga puluh enam skenario pengujian. Beberapa hal penting dapat dirangkum sebagai berikut.

a. Model hibrida CNN-ViT mencapai akurasi rerata tertinggi sebesar 93,00%, disusul CNN sebesar 90,80% dan ViT sebesar 84,80%, seluruhnya pada rasio pembagian data 80:10:10.
b. Ketiga model berada pada kondisi belajar yang sangat berbeda pada *epoch* kesepuluh: CNN dan hibrida sudah mengalami *overfitting*, sedangkan ViT masih *underfitting* dan kurvanya masih menaik. Kesimpulan yang sah karena itu terbatas pada anggaran sepuluh *epoch* tanpa bobot awal, bukan mengenai keunggulan arsitektur secara umum.
c. Rendahnya performa ViT sepenuhnya sejalan dengan teori bias induktif: ViT tidak memuat asumsi lokalitas dan pembagian bobot yang dimiliki CNN, sehingga harus mempelajari segalanya dari data dan 2.000 citra jauh dari memadai untuk itu.
d. Kekeliruan antara glioma dan meningioma mendominasi pada ketiga arsitektur, yang menandakan penyebabnya terletak pada kemiripan di dalam data dan bukan pada arsitektur, sehingga mengganti arsitektur tidak akan menyelesaikannya.
e. Keunggulan model hibrida terpusat pada kekeliruan yang paling berbahaya secara klinis, yaitu meningioma yang diprediksi sebagai tanpa tumor, yang turun dari sembilan kasus pada ViT dan empat kasus pada CNN menjadi satu kasus pada hibrida.
f. Rasio 80:10:10 unggul pada dua belas dari dua belas pasangan skenario, sedangkan hibrida unggul atas CNN hanya pada delapan dari dua belas pasangan perbedaan tingkat konsistensi ini menunjukkan kekuatan bukti yang berbeda bagi kedua kesimpulan.
g. Selisih 2,20 poin persentase antara hibrida dan CNN belum dapat dinyatakan bermakna tanpa simpangan baku antarjalan pelatihan, padahal data untuk menghitungnya sudah tersedia pada lampiran laporan.

Gagasan yang paling penting untuk dibawa dari bab ini adalah bahwa perbandingan antararsitektur menuntut kesetaraan yang jauh lebih dalam daripada menyamakan angka pada tabel konfigurasi. Menetapkan sepuluh *epoch* bagi ketiga model tampak sebagai perlakuan yang adil, padahal justru menempatkan mereka pada tahap belajar yang berbeda. Kesetaraan yang sesungguhnya bukan terletak pada penyamaan anggaran, melainkan pada pemberian kesempatan bagi setiap model untuk mencapai kondisi terbaiknya masing-masing baru setelah itu perbandingan menjadi bermakna.

Ketujuh bab pada buku ini menyampaikan pesan yang sama melalui wujud yang berbeda-beda. Bab 1 menemukan model yang benar karena alasan yang keliru. Bab 2 menemukan satu model yang tampak baik atau buruk bergantung pada cara pembacaan metriknya. Bab 3 menemukan angka yang benar secara perhitungan namun menyesatkan tanpa pembanding. Bab 4 menemukan perbandingan yang tampak adil namun runtuh oleh hal-hal di luar tabel konfigurasi. Bab 5 menemukan kegagalan yang menyerupai kegagalan model namun berasal dari satu baris kode yang tidak pernah ditulis. Bab 6 menemukan angka tertinggi yang justru paling tidak layak dipercaya. Bab ini menemukan perbandingan yang menempatkan tiga model pada tahap belajar yang berbeda. Ketujuhnya menunjuk pada keterampilan yang sama, yaitu kesediaan untuk memeriksa apa yang sesungguhnya sedang diukur sebelum menerima angka yang dihasilkannya.

➢ Daftar Pustaka

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). *Layer* Normalization. arXiv preprint arXiv:1607.06450.

Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K., & Ghayvat, H. (2021). CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope. Electronics, 10(20), 2470.

Cheng, J., Huang, W., Cao, S., Yang, R., Yang, W., Yun, Z., Wang, Z., & Feng, Q. (2015). Enhanced Performance of Brain Tumor Classification via Tumor *Region* Augmentation and Partition. PLOS ONE, 10(10).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., dkk. (2021). An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale. Dalam International Conference on Learning Representations (ICLR).

Fan, Y., dkk. (2022). Burden and Trend of Central Nervous System Cancer from 1990 to 2019. Frontiers in Public Health, 10.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Dalam Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), hlm. 770–778.

Kingma, D. P., & Ba, J. (2017). Adam: A Method for Stochastic Optimization. Dalam International Conference on Learning Representations (ICLR).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, 86(11), 2278–2324.

Liu, W., dkk. (2022). PHTrans: Parallelly Aggregating Global and Local Representations for Medical Image Segmentation. Dalam Medical Image Computing and Computer-Assisted Intervention (MICCAI).

Louis, D. N., dkk. (2021). The 2021 WHO Classification of Tumors of the Central Nervous System: A Summary. Neuro-Oncology, 23(8), 1231–1251.

Nguyen, T. T., dkk. (2023). A Hybrid *Deep Learning* Framework Combining CNN and Vision Transformer for COVID-19 Detection. Journal of Medical Imaging.

Pizer, S. M., dkk. (1987). Adaptive Histogram Equalization and Its Variations. Computer Vision, Graphics, and Image Processing, 39(3), 355–368.

Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., & Beyer, L. (2022). How to Train Your ViT? Data, Augmentation, and Regularization in Vision Transformers. Transactions on Machine Learning Research.

Tieleman, T., & Hinton, G. (2012). Lecture 6.5 — RMSProp: Divide the Gradient by a Running Average of Its Recent Magnitude. COURSERA: Neural Networks for Machine Learning.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. Dalam Advances in Neural Information Processing Systems (NeurIPS), hlm. 5998–6008.

Winnarto, M. N., dkk. (2022). Klasifikasi Tumor Otak Menggunakan Arsitektur MobileNet V2. Jurnal Teknologi Informasi dan Komunikasi.

BAB 8
KLASIFIKASI JENIS KELAMIN PADA CITRA PANORAMIK GIGI MENGGUNAKAN *DEEP LEARNING* BERBASIS *DENSENET169*

Setelah mempelajari bab ini, pembaca diharapkan mampu:

a. Menjelaskan kedudukan citra panoramik gigi sebagai sumber bukti biologis dalam forensik odontologi.
b. Menurunkan formulasi matematis operasi konvolusi, *dense connectivity*, dan *Grad-CAM*.
c. Merancang *pipeline* klasifikasi citra medis mulai dari eksplorasi data hingga interpretasi model.
d. Mengimplementasikan arsitektur *DenseNet169* dengan pendekatan *transfer learning* menggunakan *PyTorch*.
e. Mengevaluasi model secara robust menggunakan *Stratified K-Fold Cross Validation* dan metrik yang tahan terhadap ketidakseimbangan kelas.
f. Menilai secara kritis keterbatasan, aspek etis, dan kelayakan penerapan sistem pada konteks medis-legal.

➢ Pendahuluan

Latar Belakang

Identifikasi individu merupakan salah satu tahapan paling fundamental dalam praktik forensik. Di antara berbagai atribut biologis yang dapat diperkirakan dari sisa jasad manusia, jenis kelamin menempati posisi istimewa karena sifatnya yang biner dan karena keberhasilan penentuannya secara langsung memangkas ruang pencarian identitas menjadi kira-kira setengahnya. Pada kasus bencana alam, kecelakaan massal, kebakaran, maupun kejadian lain yang menyebabkan kondisi tubuh sulit dikenali secara visual, informasi jenis kelamin sering menjadi titik pijak pertama sebelum atribut lain seperti usia, tinggi badan, dan ciri individual ditelusuri lebih lanjut.

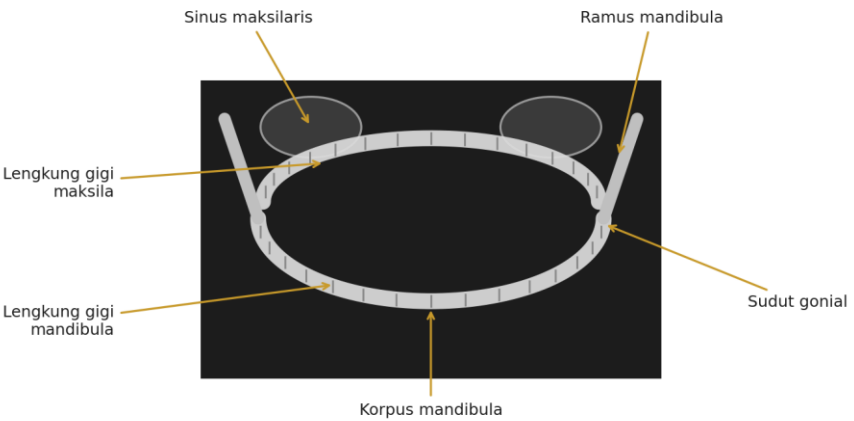
Indonesia, sebagai negara yang terletak pada pertemuan tiga lempeng tektonik dan busur vulkanik aktif, memiliki frekuensi kejadian bencana yang tinggi. Konsekuensinya, kebutuhan akan prosedur identifikasi korban yang cepat, terstandar, dan dapat direplikasi menjadi sangat nyata. Dalam kondisi seperti itu, metode identifikasi konvensional yang bertumpu pada pengenalan visual, sidik jari, atau ciri fisik permukaan sering kali tidak lagi dapat diterapkan karena jaringan lunak telah mengalami kerusakan berat.

Struktur gigi menawarkan jalan keluar. Dibandingkan bagian tubuh lain, gigi memiliki ketahanan yang jauh lebih tinggi terhadap suhu tinggi, benturan, maupun proses dekomposisi biologis. Email gigi merupakan jaringan terkeras pada tubuh manusia, dan posisinya yang terlindung di dalam rongga mulut serta dikelilingi jaringan lunak memberikan perlindungan termal tambahan. Karena itulah rekam radiografi gigi kerap menjadi satu-satunya data biometrik yang masih tersedia dan layak dianalisis ketika modalitas identifikasi lain telah gugur.

Citra panoramik gigi, yang juga dikenal sebagai *orthopantomogram* (OPG), merupakan modalitas radiografi dua dimensi yang menampilkan keseluruhan lengkung gigi, rahang atas (*maksila*), rahang bawah (*mandibula*), sinus maksilaris, serta sendi temporomandibular dalam satu bidang citra. Modalitas ini digunakan secara rutin dalam praktik kedokteran gigi, sehingga ketersediaannya pada fasilitas kesehatan relatif luas. Ketersediaan yang luas ini menjadikannya kandidat kuat sebagai sumber data untuk sistem identifikasi otomatis: data yang sudah ada dapat dimanfaatkan kembali tanpa memerlukan prosedur pencitraan tambahan.

Meskipun demikian, analisis citra panoramik secara manual masih sangat bergantung pada pengalaman dan keahlian tenaga ahli. Prosedur konvensional menuntut pengukuran metrik terhadap parameter mandibula, seperti tinggi ramus, lebar kondilus, dan besar sudut gonial, yang dilakukan satu per satu untuk setiap citra. Selain memakan waktu, interpretasi visual semacam ini rentan terhadap variasi penilaian antar pemeriksa (*inter-observer variability*) dan bahkan variasi pada pemeriksa yang sama pada waktu berbeda (*intra-observer variability*). Ketika jumlah citra yang harus

dianalisis mencapai ratusan atau ribuan, seperti pada skenario bencana massal, pendekatan manual menjadi tidak lagi praktis.



Gambar 8.1 Skema Struktur Anatomis pada Citra Panoramik Gigi Beserta *Region* yang Secara Klinis dikaitkan dengan *Dimorfisme* Seksual

Perkembangan kecerdasan buatan, khususnya *deep learning* berbasis *Convolutional Neural Network* (CNN), menawarkan jalan keluar yang menjanjikan. CNN mampu mempelajari representasi fitur secara otomatis dan bertingkat langsung dari data mentah, tanpa memerlukan rekayasa fitur manual yang bergantung pada pengetahuan domain. Kemampuan ini sangat relevan untuk kasus *dimorfisme* seksual *dentomaksilofasial*, yang polanya bersifat kompleks, non-linier, dan tersebar pada beberapa *region* anatomis sekaligus, sehingga sulit dirangkum oleh sejumlah kecil pengukuran linier.

Bab ini mengambil arsitektur *DenseNet169* sebagai contoh studi kasus. Arsitektur ini memiliki mekanisme *dense connectivity* yang menghubungkan setiap lapisan dengan seluruh lapisan sebelumnya dalam satu blok, sehingga aliran informasi dan gradien menjadi lebih baik, permasalahan *vanishing gradient* berkurang, dan penggunaan ulang fitur (*feature reuse*) menjadi efisien. Kombinasi kedalaman representasi dan efisiensi parameter membuat *DenseNet169* cocok untuk dataset citra medis yang jumlahnya relatif terbatas.

Akan tetapi, akurasi tinggi saja tidak cukup. Model *deep learning* kerap disebut sebagai kotak hitam (*black box*) karena sulit diketahui bagian citra mana yang menjadi dasar keputusannya. Pada ranah medis-legal, keberatan ini bukan sekadar persoalan akademis: hasil yang tidak dapat dijelaskan tidak dapat dipertanggungjawabkan di hadapan pengadilan maupun dalam tinjauan sejawat. Karena itu, bab ini memasangkan model klasifikasi dengan teknik *Gradient-weighted Class Activation Mapping* (*Grad-CAM*) sebagai representasi pendekatan *Explainable Artificial Intelligence* (XAI), untuk memverifikasi bahwa model benar-benar memanfaatkan struktur anatomis yang relevan dan bukan artefak atau derau pada citra.

Rumusan Permasalahan

Berdasarkan uraian di atas, permasalahan yang menjadi fokus bab ini dapat dirumuskan sebagai berikut:

a. Bagaimana merancang dan melatih model klasifikasi jenis kelamin berbasis citra panoramik gigi menggunakan arsitektur *DenseNet169* dengan pendekatan *transfer learning* pada dataset yang jumlahnya terbatas dan distribusi kelasnya tidak seimbang?
b. Skema validasi seperti apa yang mampu memberikan estimasi performa yang representatif dan tidak bias terhadap satu pembagian data tertentu?
c. Metrik evaluasi apa yang tepat digunakan ketika distribusi kelas timpang, dan bagaimana metrik tersebut ditafsirkan secara benar?
d. Sejauh mana keputusan model dapat dijelaskan melalui pemetaan area aktivasi, dan apakah area tersebut konsisten dengan pengetahuan anatomis yang berlaku dalam forensik odontologi?
e. Apa saja keterbatasan sistem yang dihasilkan dan implikasinya terhadap kelayakan penerapan pada lingkungan operasional?

Tujuan dan Ruang Lingkup

Bab ini disusun dengan tujuan membangun sistem klasifikasi jenis kelamin yang akurat, stabil, dan dapat diinterpretasikan, sekaligus mendemonstrasikan alur kerja Metodologi citra medis secara menyeluruh, mulai dari eksplorasi data hingga refleksi kritis atas

keterbatasan hasil. Sistem yang dibangun diposisikan sebagai alat bantu (*decision support*), bukan pengganti penilaian ahli forensik maupun klinisi.

Ruang lingkup pembahasan dibatasi pada klasifikasi biner (perempuan dan laki-laki) menggunakan citra panoramik dua dimensi dari satu sumber institusi. Estimasi usia, deteksi patologi, segmentasi gigi, serta pencocokan identitas individual berada di luar cakupan bab ini, meskipun sebagian di antaranya dibahas singkat sebagai arah pengembangan pada Subbab Keterbatasan dan Arah Pengembangan.

Peta Bab

Subbab Landasan Teori memaparkan landasan teori yang mencakup forensik odontologi, *dimorfisme* seksual *dentomaksilofasial*, serta dasar matematis *deep learning*, CNN, DenseNet, *transfer learning*, *Grad-CAM*, dan metrik evaluasi. Subbab Metodologi menguraikan metodologi Metodologi secara rinci. Subbab Implementasi menyajikan implementasi kode beserta penjelasannya. Subbab Hasil dan Pembahasan memaparkan hasil eksperimen dan pembahasannya. Subbab Interpretasi Model melalui *Grad-CAM* membahas interpretasi model melalui *Grad-CAM*. Subbab Keterbatasan dan Arah Pengembangan mendiskusikan keterbatasan dan arah pengembangan, Subbab Aspek Etika, Privasi, dan Kelayakan Penerapan membahas aspek etika dan legal, dan Subbab Ringkasan menutup dengan ringkasan serta soal latihan.

➢ Landasan Teori

Forensik Odontologi

Forensik odontologi adalah cabang ilmu kedokteran gigi yang berfokus pada pemanfaatan pengetahuan dental untuk kepentingan hukum dan identifikasi individu. Ruang lingkupnya mencakup pemeriksaan, evaluasi, dan penyajian bukti dental yang berkaitan dengan proses hukum. Bidang ini memegang peran sentral dalam identifikasi korban bencana massal, kecelakaan, kebakaran, maupun kasus kriminal, terutama ketika metode identifikasi lain seperti sidik jari atau pengenalan wajah tidak lagi dapat digunakan.

Keunggulan bukti dental terletak pada tiga sifat sekaligus. Pertama, ketahanan fisik: gigi mampu bertahan pada paparan suhu yang sangat tinggi, bahkan dilaporkan hingga sekitar 1.100 °C, serta relatif tahan terhadap proses pembusukan. Kedua, keunikan individual: kombinasi restorasi, pola karies, susunan gigi, dan perawatan yang pernah dilakukan membentuk pola yang secara praktis unik untuk setiap orang. Ketiga, ketersediaan rekam pembanding: praktik kedokteran gigi modern menyimpan rekam radiografi pasien, sehingga pembanding *antemortem* kerap tersedia. Identifikasi korban bencana massal di berbagai negara menempatkan forensik odontologi sebagai salah satu metode yang paling andal sekaligus paling ekonomis.

Selain identifikasi individual, karakteristik dental juga dimanfaatkan untuk memperkirakan atribut biologis populasi, yaitu usia, jenis kelamin, dan asal populasi. Estimasi atribut inilah yang menjadi objek otomatisasi dalam bab ini.

Citra Panoramik Gigi sebagai Data Digital

Radiografi panoramik dihasilkan melalui pergerakan sinkron antara sumber sinar-X dan detektor mengelilingi kepala pasien, sehingga menghasilkan proyeksi lengkung rahang yang "dibentangkan" menjadi citra dua dimensi. Nilai intensitas piksel merepresentasikan derajat atenuasi sinar-X: nilai yang mendekati nol menandakan struktur dengan densitas radiografis tinggi seperti logam restorasi atau tulang kortikal padat, sedangkan nilai yang mendekati 255 menandakan struktur yang lebih transparan terhadap sinar-X seperti rongga udara pada sinus.

Dari sudut pandang pengolahan citra digital, terdapat empat karakteristik yang perlu diperhatikan sebelum data dimasukkan ke dalam model. Pertama, citra pada dasarnya berskala keabuan (*grayscale*) meskipun disimpan dalam berkas tiga kanal. Kedua, resolusi dan dimensi citra bervariasi bergantung pada perangkat akuisisi. Ketiga, tingkat kecerahan dan kontras berbeda antar sampel akibat perbedaan pengaturan eksposur serta kondisi pasien. Keempat, sebagian kecil citra mengandung artefak berupa ornamen logam, restorasi, atau kaburan akibat pergerakan pasien selama akuisisi. Keempat hal ini secara langsung memotivasi pilihan langkah praproses yang dibahas pada Subbab 8.3.4.

Dimorfisme Seksual pada Struktur *Dentomaksilofasial*

Dimorfisme seksual merujuk pada perbedaan karakteristik biologis antara laki-laki dan perempuan yang timbul akibat interaksi faktor genetik, hormonal, dan perkembangan anatomis. Pada sistem *dentomaksilofasial*, perbedaan tersebut dapat diamati pada ukuran mahkota gigi, bentuk dan dimensi mandibula, tinggi ramus, besar sudut gonial, lebar kondilus, serta ketebalan tulang kortikal.

Secara umum, laki-laki cenderung memiliki mandibula yang lebih besar, tulang kortikal yang lebih tebal, ramus yang lebih tinggi, serta dimensi gigi yang lebih besar. Sebaliknya, perempuan cenderung memiliki kontur mandibula yang lebih halus dengan sudut gonial yang relatif lebih besar. Penting untuk dicatat bahwa perbedaan-perbedaan ini bersifat statistik, bukan deterministik: keduanya membentuk dua distribusi yang saling tumpang tindih, bukan dua kelompok yang terpisah sempurna. Derajat tumpang tindih tersebut juga dipengaruhi oleh populasi, etnis, usia, status hormonal, dan faktor lingkungan.

Konsekuensi metodologisnya ada dua. Pertama, terdapat batas atas teoretis bagi akurasi model mana pun: sebagian kasus memang secara biologis ambigu dan tidak dapat dipisahkan hanya dari morfologi tulang. Kedua, pendekatan *deep learning* menjadi menarik justru karena ia tidak perlu memilih terlebih dahulu parameter anatomis mana yang akan diukur; model mempelajari kombinasi pola yang paling diskriminatif langsung dari data.

Deep Learning dan Pembelajaran Representasi Bertingkat

Deep learning merupakan cabang *machine learning* yang termasuk dalam kategori *representation learning*, yaitu metode yang memungkinkan sistem mempelajari representasi data secara bertingkat. LeCun, Bengio, dan Hinton (2015) mendefinisikannya sebagai metode pembelajaran representasi dengan beberapa tingkat abstraksi, yang diperoleh dengan menyusun modul-modul nonlinier sederhana; masing-masing modul mentransformasikan representasi pada satu tingkat menjadi representasi pada tingkat yang lebih tinggi dan lebih abstrak.

Secara matematis, transformasi pada lapisan ke-l dari sebuah jaringan saraf dapat dinyatakan sebagai:

$$h(l) = f(W(l)\, h(l-1) + b(l)) \quad (8.1)$$

dengan $h^{(l)}$ menyatakan vektor aktivasi pada lapisan ke-*l*, $W^{(l)}$ matriks bobot, $b^{(l)}$ vektor bias, dan $f(\cdot)$ fungsi aktivasi nonlinier. Fungsi aktivasi yang paling umum digunakan adalah *Rectified Linear Unit* (ReLU):

$$f(z) = max(0, z) \quad (8.2)$$

Kehadiran fungsi nonlinier inilah yang memungkinkan penumpukan lapisan menghasilkan kapasitas representasi yang meningkat; tanpa nonlinieritas, komposisi transformasi linier tetaplah sebuah transformasi linier tunggal. Keseluruhan jaringan dengan L lapisan dapat dipandang sebagai komposisi fungsi:

$$\hat{y} = fL \circ fL - 1 \circ \ldots \circ f1\,(x\,;\,\theta) \quad (8.3)$$

dengan $\theta = \{W^{(1)}, b^{(1)}, \ldots, W^{(L)}, b^{(L)}\}$ menyatakan himpunan seluruh parameter yang dipelajari. Dalam konteks citra panoramik gigi, representasi bertingkat ini memungkinkan model menangkap pola morfologi halus yang sulit diukur secara manual: lapisan awal mempelajari tepi dan tekstur, lapisan menengah mempelajari kombinasi bentuk, dan lapisan dalam mempelajari konfigurasi anatomis tingkat tinggi.

Convolutional Neural Network

CNN merupakan arsitektur *deep learning* yang paling banyak digunakan untuk data citra. Secara umum CNN terdiri atas dua tahap: ekstraksi fitur dan klasifikasi. Pada tahap ekstraksi fitur, jaringan menerapkan operasi konvolusi terhadap masukan, biasanya diikuti fungsi aktivasi dan operasi *pooling*, secara berulang sehingga terbentuk representasi yang makin abstrak pada lapisan yang makin dalam. Pada tahap klasifikasi, peta fitur dari lapisan konvolusi terakhir diratakan menjadi vektor dan diproses melalui lapisan *fully connected* untuk menghasilkan keputusan akhir.

Operasi konvolusi dua dimensi antara citra masukan I dan kernel K dirumuskan sebagai:

$$S(i,j) = \Sigma m\, \Sigma n\, I(i + m, j + n) \cdot K(m,n) \quad (8.4)$$

Dalam praktiknya, sebuah lapisan konvolusi menerapkan sejumlah kernel sekaligus, menambahkan bias, dan meneruskan hasilnya melalui fungsi aktivasi:

$$Ac(i,j) = f(\Sigma d\, \Sigma m\, \Sigma n\, Id(i + m, j + n) \cdot Kc,d(m,n) + bc) \quad (8.5)$$

Dengan indeks c menandai kanal keluaran dan d kanal masukan. Dimensi spasial keluaran sebuah lapisan konvolusi ditentukan oleh ukuran masukan W, ukuran kernel K, *padding* P, dan *stride* S:

$$O = \lfloor (W - K + 2P) / S \rfloor + 1 \quad (8.6)$$

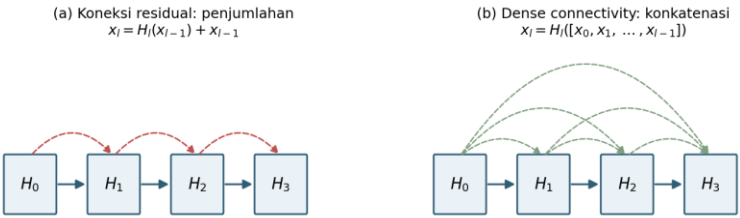
Operasi *pooling* digunakan untuk mereduksi dimensi spasial sekaligus memberikan toleransi terhadap pergeseran kecil. Bentuk yang umum adalah *max pooling* pada jendela R:

$$P(i,j) = max(m,n) \in R\, A(i + m, j + n) \quad (8.7)$$

Keunggulan utama CNN dibandingkan jaringan *fully connected* terletak pada dua sifat. Pertama, **konektivitas lokal**: setiap neuron hanya terhubung dengan wilayah kecil pada masukan, sesuai dengan sifat citra yang korelasi antarpikselnya bersifat lokal. Kedua, **pembagian bobot** (*weight sharing*): kernel yang sama digeser ke seluruh posisi citra, sehingga jumlah parameter jauh lebih sedikit sekaligus memberikan sifat *translational equivariance*. Jumlah parameter satu lapisan konvolusi adalah:

$$Nparam = (Kh \times Kw \times Din + 1) \times Dout \quad (8.8)$$

Yang tidak bergantung pada ukuran spasial citra. Sifat inilah yang membuat CNN praktis diterapkan pada citra medis berdimensi besar seperti radiografi panoramik.



Gambar 8.2 Perbandingan Skema Koneksi Antar Lapisan: (A) Koneksi Residual yang Menjumlahkan Fitur, dan (B) *Dense Connectivity* yang Mengkonkatenasi Seluruh Fitur Sebelumnya

Densely Connected Convolutional Network (DenseNet), diperkenalkan oleh Huang, Liu, van der Maaten, dan Weinberger (2017), menghubungkan setiap lapisan dengan seluruh lapisan lain secara *feed-forward* di dalam satu blok. Untuk memahami kebaruannya, bandingkan dengan formulasi *ResNet* yang menjumlahkan keluaran lapisan dengan masukannya:

$$xl = Hl(xl-1) + xl-1 \quad (8.9)$$

DenseNet menggantikan penjumlahan dengan konkatenasi seluruh peta fitur dari lapisan-lapisan sebelumnya:

$$xl = Hl([x0, x1, \ldots, xl-1]) \quad (8.10)$$

dengan $[\cdot]$ menyatakan operasi konkatenasi sepanjang dimensi kanal. Perbedaan yang tampak kecil ini berdampak besar: informasi dari lapisan awal tidak tercampur dan hilang melalui penjumlahan, melainkan tetap tersedia secara eksplisit bagi setiap lapisan berikutnya. Akibatnya, sebuah blok dengan L lapisan tidak hanya memiliki L koneksi, melainkan:

$$Nkoneksi = L(L+1)/2 \quad (8.11)$$

Jika setiap lapisan menghasilkan k peta fitur baru (disebut *growth rate*), maka jumlah kanal masukan pada lapisan ke-l adalah:

$$kl = k0 + k(l-1) \quad (8.12)$$

Fungsi komposit H_l pada DenseNet terdiri atas tiga operasi berurutan, yaitu normalisasi batch, aktivasi ReLU, dan konvolusi:

$$Hl(\cdot) = Conv3 \times 3(ReLU(BN(\cdot))) \quad (8.13)$$

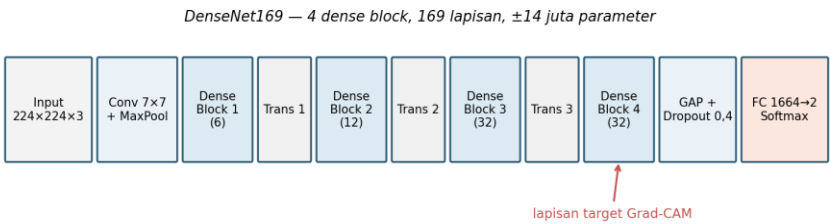
Karena konkatenasi mensyaratkan dimensi spasial yang sama, DenseNet membagi jaringan menjadi beberapa *dense block* yang dipisahkan oleh *transition layer*. Lapisan transisi bertugas mereduksi jumlah kanal dan dimensi spasial:

$$T(\cdot) = AvgPool2 \times 2(Conv1 \times 1(ReLU(BN(\cdot)))) \quad (8.14)$$

Rancangan ini memberikan empat keuntungan yang saling menguatkan: mitigasi *vanishing gradient* karena setiap lapisan memiliki akses langsung ke gradien dari fungsi *loss*; penguatan propagasi fitur; dorongan terhadap penggunaan ulang fitur; serta

pengurangan jumlah parameter secara signifikan dibandingkan arsitektur CNN konvensional dengan kedalaman setara.

DenseNet169 adalah salah satu varian keluarga DenseNet, bersama DenseNet121, DenseNet201, dan DenseNet264, yang dibedakan oleh jumlah lapisan pada tiap *dense block*. *DenseNet169* memiliki konfigurasi blok (6, 12, 32, 32) dengan kedalaman total 169 lapisan dan jumlah parameter sekitar 14 juta. Varian ini dipilih dalam bab ini karena menawarkan keseimbangan antara kapasitas representasi dan efisiensi komputasi, sehingga sesuai untuk dataset citra medis yang jumlahnya terbatas dan untuk lingkungan komputasi dengan sumber daya GPU yang dibatasi kuota.



Gambar 8.3 Arsitektur *DenseNet169* yang Digunakan, dengan Denseblock4 Sebagai Lapisan Target untuk Pemetaan *Grad-CAM*

Transfer Learning

Transfer learning adalah pendekatan yang memungkinkan pengetahuan yang telah dipelajari dari satu domain atau tugas dimanfaatkan untuk meningkatkan proses pembelajaran pada domain atau tugas lain yang berkaitan. Pan dan Yang (2009) menjelaskan bahwa pendekatan ini relevan ketika tugas pada domain sasaran hanya memiliki data pelatihan terbatas, sementara domain lain yang berkaitan memiliki data berlimpah meskipun mungkin berasal dari ruang fitur atau distribusi yang berbeda.

Secara formal, sebuah domain didefinisikan sebagai pasangan ruang fitur dan distribusi marginal:

$$\mathcal{D} = \{\mathcal{X}, P(X)\}, \quad X = \{x1, \ldots, xn\} \in \mathcal{X} \qquad (8.15)$$

Sedangkan sebuah tugas didefinisikan sebagai pasangan ruang label dan fungsi prediktif:

$$\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}, \quad f(x) = P(y \mid x) \qquad (8.16)$$

Transfer learning bertujuan meningkatkan pembelajaran fungsi f*T(·) pada domain sasaran* $\mathcal{D}$T dengan memanfaatkan pengetahuan dari $\mathcal{D}\,S$ *dan* $\mathcal{T}$S, ketika $\mathcal{D}\,S \neq \mathcal{D}$T atau $\mathcal{T}\,S \neq \mathcal{T}$T. Dalam konteks *deep learning*, hal ini diwujudkan melalui inisialisasi parameter:

$$\theta T(0) \leftarrow \theta S* \ , \ kemudian \ \ \theta T* = arg\,min\theta\, \mathcal{L}T(\theta) \quad (8.17)$$

Artinya parameter model pada domain sasaran tidak diinisialisasi secara acak, melainkan menggunakan parameter optimal hasil pra-pelatihan pada domain sumber, lalu disesuaikan melalui *fine-tuning.*

Pada bab ini, domain sumber adalah *ImageNet*, yaitu dataset citra alami berskala besar, sedangkan domain sasaran adalah radiografi panoramik gigi. Meskipun kedua domain jelas berbeda secara semantik, fitur tingkat rendah hingga menengah yang dipelajari dari *ImageNet*, berupa detektor tepi, tekstur, gradien, dan pola geometri sederhana, tetap relevan untuk citra radiografi. Pendekatan *fine-tuning* penuh diterapkan, yaitu seluruh lapisan diizinkan diperbarui, namun dengan laju pembelajaran yang kecil agar pengetahuan awal tidak terhapus secara drastis (*catastrophic forgetting*).

Fungsi *Loss* dan Label *Smoothing*

Untuk klasifikasi multikelas, keluaran jaringan berupa vektor skor mentah (*logit*) yang diubah menjadi distribusi probabilitas melalui fungsi softmax:

$$pi \ = \ exp(zi) \ / \ \Sigma j \ exp(zj) \quad (8.18)$$

Fungsi *loss* yang digunakan adalah *cross entropy* antara distribusi target q dan distribusi prediksi p:

$$\mathcal{L}CE \ = \ -\Sigma i \ qi \ log \ pi \qquad (8.19)$$

Pada bentuk baku, q merupakan vektor *one-hot* sehingga target bernilai tepat 1 untuk kelas benar dan 0 untuk kelas lain. Konfigurasi ini mendorong model menghasilkan logit yang sangat besar untuk kelas benar, yang pada gilirannya menyebabkan model menjadi terlalu yakin (*overconfident*) dan kurang mampu menggeneralisasi. *Label smoothing* mengatasinya dengan melunakkan target:

$$q'i \ = \ (1 \ - \ \varepsilon) \ qi \ + \ \varepsilon \ / \ C \quad (8.20)$$

Dengan ε sebagai parameter pelunakan dan C jumlah kelas. Untuk kasus biner (C = 2) dengan ε = 0,1, target berubah dari {0; 1} menjadi {0,05; 0,95}. Fungsi *loss* akhirnya menjadi:

$$\mathcal{L}LS = -\Sigma i\,[\,(1-\varepsilon)\,qi + \varepsilon / C\,]\,log\,pi \quad (8.21)$$

Dalam konteks medis, pelunakan ini memiliki justifikasi konseptual yang kuat: label "laki-laki" atau "perempuan" memang pasti secara administratif, tetapi manifestasi morfologisnya pada citra tidaklah pasti. Label *smoothing* merefleksikan ketidakpastian inheren tersebut ke dalam fungsi objektif.

Optimasi: AdamW dan Penjadwalan Laju Pembelajaran

Optimasi parameter dilakukan menggunakan AdamW, varian dari Adam yang memisahkan regularisasi *weight decay* dari pembaruan gradien adaptif. Adam memelihara estimasi momen pertama dan kedua dari gradien:

$$mt = \beta 1\,mt-1 + (1-\beta 1)\,gt \quad (8.22)$$
$$vt = \beta 2\,vt-1 + (1-\beta 2)\,gt2 \quad (8.23)$$

Karena kedua estimasi tersebut bias terhadap nol pada iterasi awal, dilakukan koreksi bias:

$$\hat{m}t = mt / (1-\beta 1t)\ ,\quad \hat{v}t = vt / (1-\beta 2t) \quad (8.24)$$

Pembaruan parameter pada AdamW kemudian dilakukan dengan menambahkan suku *weight decay* secara terpisah, tidak melalui gradien:

$$\theta t = \theta t-1 - \eta\,[\,\hat{m}t / (\sqrt{\hat{v}t} + \delta) + \lambda\,\theta t-1\,] \quad (8.25)$$

Dengan η laju pembelajaran, λ koefisien *weight decay*, dan δ konstanta kecil untuk stabilitas numerik. Pemisahan ini penting karena pada Adam baku, regularisasi L2 yang dimasukkan ke dalam gradien akan ikut diskalakan oleh √v̂_t sehingga efeknya menjadi tidak seragam antar parameter.

Laju pembelajaran tidak dipertahankan konstan, melainkan dijadwalkan mengikuti kebijakan *OneCycle* berbasis kosinus dengan fase pemanasan (*warmup*). Selama fase pemanasan sepanjang T_w epoch, laju pembelajaran dinaikkan secara linier; setelahnya diturunkan mengikuti kurva kosinus:

$$\eta t = \eta min + ½ (\eta max - \eta min) [1 + cos(\pi (t - Tw) / (T - Tw))] \quad (8.26)$$

Fase pemanasan mencegah pembaruan besar yang merusak bobot pra-latih pada *epoch* awal, sedangkan peluruhan kosinus memungkinkan model menyempurnakan solusi pada tahap akhir pelatihan dengan langkah yang makin kecil.

Grad-CAM sebagai Pendekatan *Explainable* AI

Grad-CAM, diperkenalkan oleh Selvaraju dkk. (2017), menghasilkan peta lokalisasi kasar yang menyoroti area citra yang paling berkontribusi terhadap prediksi suatu kelas. Kelebihannya dibandingkan teknik sebelumnya adalah fleksibilitas: teknik ini dapat diterapkan pada berbagai arsitektur CNN tanpa memerlukan perubahan arsitektur maupun pelatihan ulang.

Misalkan A^k menyatakan peta aktivasi kanal ke-*k* pada lapisan konvolusi target, dan y^c menyatakan skor (logit) untuk kelas c sebelum softmax. Bobot kepentingan setiap kanal dihitung sebagai rata-rata spasial dari gradien:

$$\alpha kc = (1 / Z) \Sigma i \Sigma j \ \partial yc / \partial Akij \quad (8.27)$$

Dengan Z menyatakan jumlah elemen spasial pada peta aktivasi. Peta *Grad-CAM* diperoleh melalui kombinasi linier berbobot yang diikuti aktivasi ReLU:

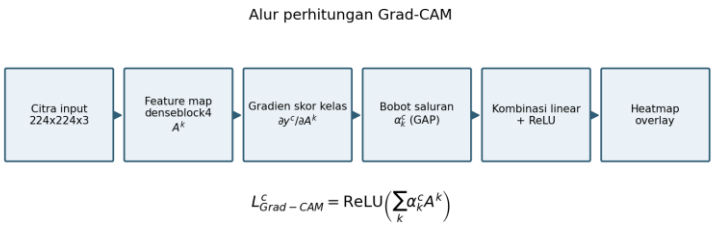
$$LcGrad - CAM = ReLU(\Sigma k \ \alpha kc \ Ak) \quad (8.28)$$

Penerapan ReLU bersifat esensial: tanpa ReLU, peta akan memuat kontribusi negatif, yaitu *region* yang justru menekan skor kelas c, sehingga visualisasi menjadi sulit ditafsirkan. Peta hasil perhitungan kemudian dinormalisasi ke rentang [0, 1] dan diperbesar ke dimensi citra asli melalui interpolasi bilinier sebelum ditumpangkan sebagai *heatmap*:

$$H = Upsample((L - min L) / (max L - min L)) \quad (8.29)$$

Pada bab ini, lapisan target yang dipilih adalah ***denseblock4***, yaitu blok konvolusi terakhir *DenseNet169*. Pemilihan ini didasarkan pada prinsip umum bahwa lapisan konvolusi terakhir memberikan keseimbangan terbaik antara ketajaman semantik dan resolusi spasial: lapisan yang terlalu awal menghasilkan peta yang

tajam namun tidak bermakna secara semantik, sedangkan lapisan *fully connected* telah kehilangan informasi spasial sepenuhnya.



Gambar 8.4 Alur Perhitungan *Grad-CAM* dari Citra Masukan Hingga Heatmap yang Ditumpangkan pada Citra Asli

Metrik Evaluasi Klasifikasi

Pada klasifikasi biner, hasil prediksi dirangkum dalam *confusion matrix* yang memuat empat besaran: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Dari keempat besaran ini diturunkan sejumlah metrik. Akurasi menyatakan proporsi prediksi benar terhadap seluruh sampel:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (8.30)$$

Presisi menyatakan proporsi prediksi positif yang benar-benar positif, sedangkan *recall* (sensitivitas) menyatakan proporsi sampel positif yang berhasil dikenali:

$$Precision = TP / (TP + FP) \quad (8.31)$$

$$Recall = TP / (TP + FN) \quad (8.32)$$

F1-score merupakan rata-rata harmonik keduanya, sehingga hanya bernilai tinggi bila presisi dan *recall* sama-sama tinggi:

$$F1 = 2 \cdot (Precision \times Recall) / (Precision + Recall) \quad (8.33)$$

Ketika distribusi kelas timpang, akurasi menjadi metrik yang menyesatkan. Pada dataset dengan komposisi 65 : 35, model naif yang selalu memprediksi kelas mayoritas sudah memperoleh akurasi 65% tanpa mempelajari apa pun. Karena itu digunakan AUC-ROC sebagai metrik utama. Kurva ROC memetakan hubungan antara *True Positive Rate* dan *False Positive Rate* pada seluruh kemungkinan ambang keputusan:

$$TPR(\tau) = TP(\tau) / [\, TP(\tau) + FN(\tau) \,], \quad FPR(\tau) = FP(\tau) / [\, FP(\tau) + TN(\tau) \,] \quad (8.34)$$

Luas daerah di bawah kurva tersebut adalah:

$$AUC = \int 01\, TPR(\, FPR - 1(u) \,)\, du \qquad (8.35)$$

Nilai AUC memiliki tafsir probabilistik yang intuitif: AUC menyatakan peluang bahwa sebuah sampel positif yang dipilih acak memperoleh skor lebih tinggi daripada sampel negatif yang dipilih acak. Nilai 1,0 menandakan pemisahan sempurna, sedangkan 0,5 setara dengan tebakan acak. Karena AUC dihitung atas seluruh ambang sekaligus, metrik ini tidak terpengaruh oleh pemilihan ambang maupun oleh proporsi kelas.

Terakhir, karena evaluasi dilakukan melalui validasi silang, hasil dilaporkan sebagai rata-rata dan simpangan baku antar lipatan:

$$\mu = (1/K)\, \Sigma k = 1K\, mk \;, \quad \sigma = \sqrt{[\, (1/K)\, \Sigma k = 1K\, (mk - \mu)2 \,]} \quad (8.36)$$

Simpangan baku yang kecil menandakan bahwa performa model tidak sensitif terhadap pembagian data tertentu, yang merupakan indikator kestabilan dan generalisasi yang baik.

➢ Metodologi

Alur Umum Metodologi

Metodologi disusun dalam lima tahapan berurutan yang membentuk satu alur tertutup: eksplorasi dan persiapan data, pemilihan dan perancangan arsitektur model, pelatihan dan validasi, evaluasi performa, serta interpretasi model menggunakan pendekatan XAI. Urutan ini bukan sekadar konvensi administratif. Setiap tahap menghasilkan keputusan yang membatasi ruang pilihan pada tahap berikutnya: karakteristik data menentukan strategi praproses; strategi praproses membatasi arsitektur yang masuk akal; arsitektur menentukan lapisan mana yang dapat dijadikan target *Grad-CAM*.



Gambar 8.5 Alur Metodologi Klasifikasi Jenis Kelamin Berbasis Citra Panoramik Gigi

Perlu ditegaskan bahwa tahap kelima, interpretasi model, bukanlah pelengkap kosmetik yang ditambahkan di akhir. Pada domain medis-legal, interpretasi berfungsi sebagai mekanisme falsifikasi: apabila peta aktivasi menunjukkan bahwa model memusatkan perhatian pada penanda perangkat, teks anotasi, atau tepi citra, maka nilai akurasi setinggi apa pun harus dianggap tidak sahih karena model mempelajari korelasi semu, bukan pola biologis.

Deskripsi Dataset

Dataset yang digunakan berupa citra radiografi panoramik gigi yang dikumpulkan dari Rumah Sakit Gigi dan Mulut (RSGM) melalui prosedur perizinan yang berlaku. Dataset terdiri atas **5.203 citra** berformat JPEG yang dilengkapi informasi jenis kelamin dan usia pasien. Informasi tersebut dikodekan langsung pada nama berkas dengan konvensi [id]_[gender]_[usia].jpg. Sebagai contoh, berkas bernama 00001_F_53.81.jpg menunjukkan pasien perempuan berusia 53,81 tahun. Selain itu tersedia berkas metadata.csv yang memuat tiga kolom utama.

Tabel 8.1 Struktur Metadata Dataset

Kolom	**Tipe Data**	**Deskripsi**	**Contoh Nilai**
id	Integer	Identitas unik pasien (anonim)	00001
gender	String	Jenis kelamin (F / M)	F
age	Float	Usia pasien dalam tahun	53.81

Pengodean informasi pada nama berkas memiliki konsekuensi teknis yang perlu diantisipasi. Rekonstruksi nama berkas dari metadata harus mereproduksi format persis seperti yang tersimpan di sistem berkas, termasuk jumlah digit desimal pada usia. Sebagaimana dibahas pada Subbab 8.4.2, perbedaan sepele antara

51.0 dan 51.00 sudah cukup untuk menggagalkan pembacaan seluruh dataset.

Struktur direktori dataset terdiri atas folder *image/* yang memuat seluruh berkas citra dan berkas metadata.csv pada direktori utama. Dataset disimpan sebagai dataset privat pada platform Kaggle sehingga dapat diakses secara terkendali oleh anggota tim peneliti.

Eksplorasi dan Analisis Data

Eksplorasi data (*exploratory data analysis*, EDA) dilakukan sebelum pemodelan untuk memahami karakteristik dataset dan meng-identifikasi permasalahan yang dapat memengaruhi proses pelatihan. Metode yang digunakan bersifat deskriptif, mencakup visualisasi distribusi frekuensi kelas dan kelompok usia, analisis dimensi citra, analisis statistik intensitas piksel, serta inspeksi visual terhadap sampel citra.

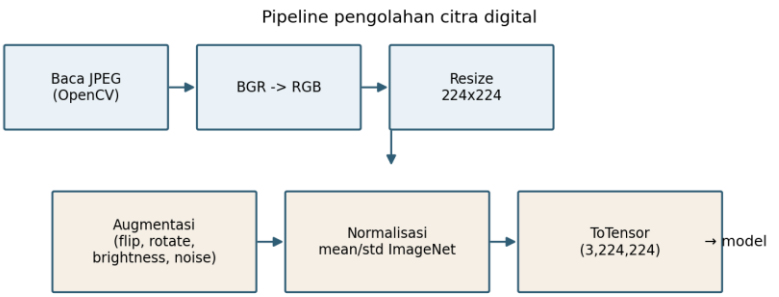
Pemilihan pendekatan deskriptif ini disengaja. Pada tahap awal, tujuannya bukan menguji hipotesis statistik, melainkan membangun intuisi terhadap data: seberapa timpang distribusi kelasnya, apakah usia berkorelasi dengan kelas sehingga berpotensi menjadi variabel pengganggu, dan apakah terdapat perbedaan karakteristik akuisisi antar kelas yang dapat dipelajari model sebagai pintasan (*shortcut learning*). Ketiga pertanyaan tersebut secara langsung menentukan rancangan validasi dan pemilihan metrik.

Empat aspek yang dianalisis adalah:

a. Distribusi kelas. Menentukan apakah dataset seimbang; hasilnya menentukan pilihan metrik evaluasi dan strategi pembagian data.
b. Distribusi usia per kelas. Menguji apakah usia dapat menjadi variabel pengganggu; bila median usia kedua kelas berbeda jauh, model berpotensi mempelajari penanda usia alih-alih penanda jenis kelamin.
c. Dimensi dan resolusi citra. Menentukan kebutuhan penyeragaman ukuran masukan.
d. Kecerahan dan kontras per kelas. Menguji apakah terdapat perbedaan sistematis dalam karakteristik akuisisi antar kelas yang dapat dieksploitasi model sebagai pintasan.

Praproses dan Augmentasi Citra

Praproses merupakan tahap kritis yang menjembatani karakteristik data mentah dengan asumsi yang dibangun ke dalam arsitektur model. *Pipeline* praproses diimplementasikan menggunakan pustaka Albumentations dan dipisahkan menjadi dua jalur: jalur pelatihan yang memuat augmentasi acak, dan jalur validasi yang hanya memuat transformasi deterministik.



Gambar 8.6 *Pipeline* Pengolahan Citra Digital dari Pembacaan Berkas hingga Tensor Masukan Model

Pada jalur pelatihan, langkah yang diterapkan berturut-turut adalah: penyeragaman ukuran citra ke 224 × 224 piksel; pencerminan horizontal acak dengan probabilitas 0,5; rotasi acak pada rentang −10° hingga +10°; penyesuaian kecerahan dan kontras acak; penambahan derau Gaussian; serta normalisasi menggunakan statistik *ImageNet*. Normalisasi dilakukan per kanal mengikuti:

$$x' = (x \,/\, 255 - \mu) \,/\, \sigma \qquad (8.37)$$

dengan $\mu = [0{,}485; 0{,}456; 0{,}406]$ dan $\sigma = [0{,}229; 0{,}224; 0{,}225]$. Penggunaan statistik *ImageNet*, bukan statistik dataset sendiri, merupakan konsekuensi logis dari *transfer learning*: bobot pra-latih dioptimalkan pada distribusi masukan tersebut, sehingga menyimpang darinya akan menggeser aktivasi lapisan awal ke rentang yang tidak dikenali model.

Pemilihan jenis augmentasi didasarkan pada pertimbangan validitas anatomis, bukan sekadar mengikuti praktik umum pada citra alami. Pencerminan horizontal dapat dibenarkan karena posisi kepala pasien memang bervariasi dan lengkung rahang bersifat

kurang lebih simetris bilateral. Rotasi dibatasi hingga 10° karena rotasi besar akan mendistorsi hubungan spasial antar struktur anatomis secara tidak realistis dan justru menghasilkan citra yang tidak pernah muncul dalam praktik klinis. Penyesuaian kecerahan dan kontras mensimulasikan variasi pengaturan eksposur antar perangkat dan antar operator. Sebaliknya, pencerminan vertikal, distorsi elastis, dan pemotongan acak yang agresif tidak digunakan karena berpotensi menghilangkan atau merusak informasi morfologis yang justru menjadi dasar klasifikasi.

Tabel 8.2 Konfigurasi Augmentasi dan Justifikasi Anatomisnya

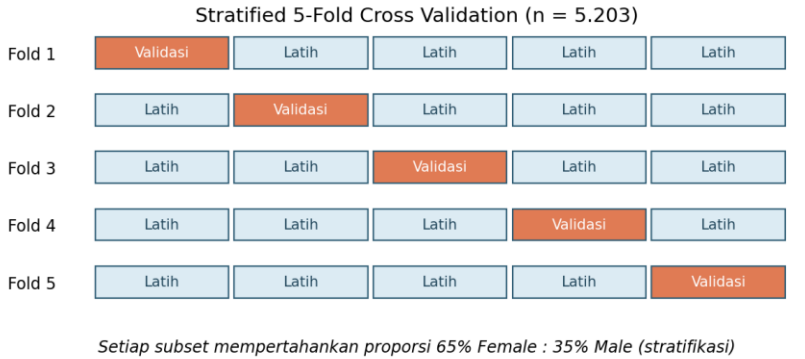
Transformasi	**Parameter**	**Justifikasi**
Resize	224 × 224	Menyeragamkan dimensi masukan sesuai ekspektasi *DenseNet169* pra-latih
Horizontal Flip	p = 0,5	Simetri bilateral lengkung rahang; variasi posisi kepala pasien
Rotate	±10°, p = 0,5	Variasi penempatan pasien; dibatasi agar tidak mendistorsi anatomi
RandomBrightness Contrast	p = 0,4	Variasi pengaturan eksposur antar perangkat dan operator
GaussNoise	p rendah	Ketahanan terhadap derau akuisisi dan kompresi
Normalize	mean/std *ImageNet*	Menyelaraskan distribusi masukan dengan bobot pra-latih
ToTensorV2	—	Konversi array NumPy (H, W, C) menjadi tensor *PyTorch* (C, H, W)

Pada jalur validasi, hanya penyeragaman ukuran dan normalisasi yang diterapkan. Pemisahan ini bersifat wajib secara metodologis: augmentasi acak pada data validasi akan membuat nilai metrik bervariasi antar evaluasi meskipun modelnya identik, sehingga perbandingan antar *epoch* dan antar lipatan menjadi tidak sahih.

Skema Validasi: *Stratified 5-Fold Cross Validation*

Evaluasi menggunakan pembagian tunggal antara data latih dan data uji (*single train-test split*) memiliki dua kelemahan mendasar pada dataset berukuran sedang. Pertama, estimasi performa yang dihasilkan memiliki ragam yang besar karena bergantung pada satu realisasi pembagian acak. Kedua, tidak tersedia informasi mengenai stabilitas performa, sehingga sulit membedakan model yang benar-benar baik dari model yang kebetulan beruntung pada satu pembagian.

Karena itu digunakan **Stratified 5-Fold Cross *Validation***. Dataset dibagi menjadi lima lipatan; pada setiap iterasi, empat lipatan digunakan untuk pelatihan dan satu lipatan untuk validasi, sehingga setiap sampel tepat satu kali menjadi data validasi. Sifat *stratified* menjamin proporsi kelas pada setiap lipatan mendekati proporsi pada keseluruhan dataset, yaitu sekitar 65% perempuan dan 35% laki-laki. Tanpa stratifikasi, pembagian acak pada dataset timpang berisiko menghasilkan lipatan dengan komposisi kelas yang menyimpang jauh, yang membuat estimasi performa menjadi tidak stabil.



Gambar 8.7 Skema *Stratified 5-Fold Cross Validation* dengan Proporsi Kelas yang dipertahankan pada Setiap Lipatan

Dengan skema ini, setiap lipatan menggunakan 4.162 atau 4.163 sampel untuk pelatihan dan 1.040 atau 1.041 sampel untuk validasi. Hasil akhir dilaporkan sebagai rata-rata dan simpangan baku dari kelima lipatan sesuai Persamaan (8.36), sehingga yang dilaporkan bukan satu titik estimasi melainkan gambaran distribusi performa.

Perancangan Arsitektur Model

Model dibangun di atas *backbone DenseNet169* yang telah dilatih pada *ImageNet*. Bagian ekstraktor fitur dipertahankan seluruhnya, sedangkan kepala klasifikasi diganti agar sesuai dengan tugas biner. Kepala klasifikasi yang baru terdiri atas empat komponen berurutan: *adaptive average pooling* yang meringkas peta fitur menjadi vektor berdimensi 1.664; operasi *flatten*; *dropout* dengan laju 0,4 sebagai regularisasi; dan lapisan *fully connected* dengan dua neuron keluaran.

Tiga pertimbangan mendasari pemilihan arsitektur ini. Pertama, kedalaman *DenseNet169* memadai untuk menangkap pola morfologi tulang rahang yang kompleks tanpa jumlah parameter yang berlebihan. Kedua, mekanisme *dense connectivity* memberikan ketahanan yang lebih baik terhadap *vanishing gradient*, sehingga proses pelatihan lebih stabil. Ketiga, ukuran model yang relatif kompak, sekitar 14 juta parameter, memungkinkan pelatihan yang efisien pada GPU kelas T4 yang tersedia di platform Kaggle, khususnya mengingat skema lima lipatan menuntut lima kali proses pelatihan penuh.

Untuk menempatkan pilihan tersebut dalam konteks, Tabel 8.5 membandingkan beberapa arsitektur CNN yang lazim digunakan pada citra medis. Perbandingan ini bersifat kualitatif dan dimaksudkan sebagai kerangka pengambilan keputusan, bukan sebagai hasil eksperimen; pengujian empiris terhadap alternatif tersebut disarankan sebagai pengembangan lanjutan.

Tabel 8.5 Perbandingan Kualitatif Kandidat *Arsitektur Backbone*

Arsitektur	**Parameter**	**Mekanisme**	**Pertimbangan untuk kasus ini**
*ResNet*50	±25,6 juta	Koneksi residual (penjumlahan)	Sangat mapan, namun parameter lebih banyak dengan kedalaman efektif serupa
DenseNet121	±8,0 juta	*Dense connectivity*	Paling hemat parameter; dilaporkan terbaik

			pada studi Franco dkk. (2022)
DenseNet169	±14,1 juta	*Dense connectivity*	Kapasitas lebih besar dari DenseNet121 dengan parameter masih di bawah *ResNet*50
EfficientNet-B0	±5,3 juta	Penskalaan majemuk	Sangat efisien, namun lebih sensitif terhadap resolusi masukan dan penjadwalan laju belajar
Vision Transformer	±86 juta	*Self-attention*	Menuntut data jauh lebih banyak; kurang sesuai untuk 5.203 sampel

Dari tabel tersebut, *DenseNet169* menempati posisi tengah yang masuk akal: cukup dalam untuk menangkap pola morfologi yang kompleks, namun cukup ringan untuk dilatih lima kali dalam skema validasi silang pada satu sesi GPU. Perlu ditegaskan bahwa argumen ini bersifat *a priori*; pembuktian bahwa *DenseNet169* memang unggul untuk tugas ini menuntut perbandingan langsung dengan konfigurasi hiperparameter yang identik, sebagaimana dirancang pada Subbab 8.7.4.

Nilai *dropout* 0,4 dipilih relatif tinggi secara sengaja. Dengan 5.203 sampel dan 14 juta parameter, rasio data terhadap parameter sangat kecil, sehingga risiko *overfitting* tinggi dan regularisasi yang kuat pada kepala klasifikasi menjadi wajar.

Konfigurasi Pelatihan

Seluruh hiperparameter pelatihan dirangkum pada Tabel 8.3. Konfigurasi ini disimpan dalam satu berkas YAML yang berfungsi sebagai *single source of truth* bagi seluruh eksperimen.

Tabel 8.3 Konfigurasi Hiperparameter Pelatihan

Hiperparameter	**Nilai**	**Keterangan**
Optimizer	AdamW	*Weight decay* terpisah dari gradien adaptif
Learning rate	1×10^{-4}	Kecil, untuk menjaga bobot pra-latih
Weight decay	1×10^{-4}	Regularisasi L2 terdekopel
Batch size	16	Menyesuaikan kapasitas memori GPU T4
Scheduler	OneCycleLR (cosine)	Peluruhan kosinus setelah fase pemanasan
Warmup epochs	3	Mencegah gangguan besar pada bobot pra-latih
Epoch maksimum	50	Batas atas; jarang tercapai karena *early stopping*
Early stopping patience	10	Menghentikan pelatihan bila performa validasi stagnan
Dropout	0,4	Regularisasi pada kepala klasifikasi
Loss	*CrossEntropy* + label *smoothing* 0,1	Mengurangi *overconfidence*
Mixed precision (AMP)	Aktif	Mempercepat komputasi, menghemat memori
Seed	42	Menjamin reproduktifitas

Penggunaan *mixed precision training* perlu dijelaskan singkat. Teknik ini melakukan sebagian besar komputasi dalam presisi 16-bit sambil mempertahankan salinan bobot dalam presisi 32-bit, sehingga kecepatan komputasi meningkat dan konsumsi memori menurun tanpa penurunan kualitas yang berarti. Karena rentang bilangan 16-bit lebih sempit, gradien perlu diskalakan terlebih dahulu agar nilai kecil tidak menjadi nol (*underflow*); mekanisme ini ditangani oleh objek *gradient scaler*.

Lingkungan Implementasi dan Struktur Proyek

Implementasi dilakukan menggunakan bahasa pemrograman Python dengan kerangka kerja *PyTorch*, didukung pustaka torchvision, Albumentations, OpenCV, NumPy, Pandas, dan scikit-learn. Pelatihan dijalankan pada Kaggle Notebook yang menyediakan akses GPU NVIDIA T4.

Kode disusun dalam struktur modular, bukan sebagai satu berkas *notebook* monolitik. Pemisahan ini memberikan tiga manfaat: setiap komponen dapat diuji secara terpisah; perubahan pada satu modul tidak berisiko merusak modul lain; dan logika Metodologi dapat direproduksi melalui satu perintah, bukan melalui eksekusi sel *notebook* secara berurutan yang rawan kesalahan.

Tabel 8.4 Struktur Modul Proyek

Modul	**Tanggung Jawab**
data/splits.py	Rekonstruksi nama berkas, pengodean label, pembagian *stratified K-Fold*
data/dataset.py	Kelas Dataset *PyTorch* untuk pembacaan citra panoramik
data/transforms.py	*Pipeline* augmentasi pelatihan dan transformasi validasi
models/densenet.py	Definisi arsitektur *DenseNet169* dengan kepala klasifikasi baru
training/engine.py	*Loop* pelatihan dan evaluasi *per epoch*
training/losses.py	Konstruksi fungsi *loss* dengan label *smoothing*
training/metrics.py	Perhitungan AUC, akurasi, F1, dan laporan per kelas
xai/gradcam.py	Implementasi *Grad-CAM* dan pembuatan *overlay heatmap*
eda/insights.py	Visualisasi eksplorasi data
utils/seed.py	Penetapan seed global untuk reproduktifitas

➢ Implementasi

Persiapan Data dan Pengodean Label

Tahap pertama adalah menyiapkan kerangka data yang memuat nama berkas, label numerik, dan indeks lipatan. Tiga fungsi bekerja secara berurutan: rekonstruksi nama berkas dari metadata, pengodean label kategorik menjadi bilangan bulat, dan penambahan kolom lipatan.

```
def prepare_dataframe(df, n_splits=5, seed=42):
    # 1. Rekonstruksi nama berkas: [id]_[gender]_[usia].jpg
    df = build_filename(df)
    # 2. Pengodean label: F -> 0 (Female), M -> 1 (Male)
    df = encode_gender(df, label_col='gender')
    # 3. Penambahan kolom fold dengan stratifikasi
    df = add_kfold_column(df, label_col='label',
                          n_splits=n_splits, seed=seed)
    return df
def encode_gender(df, label_col='gender'):
    mapping = {'F': 0, 'M': 1}
    df['label'] = df[label_col].map(mapping)
    # Validasi eksplisit: gagal cepat bila ada nilai tak dikenal
    if df['label'].isna().any():
        unknown = df[df['label'].isna()][label_col].unique()
        raise ValueError(f'Nilai gender tidak dikenal: {unknown}')
    return df
def add_kfold_column(df, label_col='label', n_splits=5,
seed=42):
    df['fold'] = -1
    skf = StratifiedKFold(n_splits=n_splits, shuffle=True,
                          random_state=seed)
    for fold_idx, (_, val_idx) in enumerate(skf.split(df,
df[label_col])):
        df.loc[val_idx, 'fold'] = fold_idx
    return df
```

Kode 8.1 Persiapan Kerangka Data (*data/splits.py*)

Dua rincian pada kode di atas layak diperhatikan. Pertama, validasi eksplisit pada fungsi encode*gender* menerapkan prinsip fail fast: bila terdapat nilai gender di luar {F, M}, program berhenti dengan pesan yang jelas alih-alih meneruskan nilai NaN yang akan menimbulkan kegagalan misterius pada tahap pelatihan. Kedua, penetapan randomstate pada *StratifiedKFold* memastikan pembagian lipatan dapat direproduksi persis pada eksekusi berikutnya.

Rekonstruksi nama berkas merupakan sumber kesalahan yang nyata pada implementasi awal. Berkas citra menggunakan format usia dengan dua digit desimal tetap, misalnya 51.00, sedangkan pemformatan bawaan Python menghasilkan 51.0. Perbedaan satu karakter ini menyebabkan seluruh pembacaan berkas gagal. Penyelesaiannya adalah menggunakan pemformatan eksplisit f'{age:.2f}'. Kasus ini menegaskan pentingnya melakukan uji cepat pada subset kecil data sebelum menjalankan pelatihan penuh.

Kelas Dataset dan Representasi Citra

Kelas Dataset bertanggung jawab membaca citra dari cakram, mengonversi ruang warna, menerapkan transformasi, dan mengembalikan tensor beserta labelnya.

```
class DentalPanoramicDataset(Dataset):
    def __init__(self, df, image_dir, transform=None,
                 image_col='filename', label_col='label'):
        self.df = df.reset_index(drop=True)
        self.image_dir = Path(image_dir)
        self.transform = transform
        self.image_col = image_col
        self.label_col = label_col
    def __len__(self):
        return len(self.df)

    def __getitem__(self, idx: int) -> dict:
        row = self.df.iloc[idx]
        img_path = self.image_dir / row[self.image_col]
        image = cv2.imread(str(img_path))
        if image is None:
```

```
        raise FileNotFoundError(f'Citra tidak ditemukan: {img_path}')
        # OpenCV membaca sebagai BGR; model pra-latih mengharapkan RGB
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        if self.transform is not None:
            image = self.transform(image=image)['image']
        sample = {'image': image, 'filename': str(row[self.image_col])}
        if self.label_col is not None:
            sample['label'] = torch.tensor(int(row[self.label_col]),
                                           dtype=torch.long)
        return sample
```

Kode 8.2 Kelas Dataset *PyTorch* (data/dataset.py)

Konversi BGR ke RGB bukan detail kosmetik. OpenCV membaca citra dengan urutan kanal biru–hijau–merah, sedangkan bobot pra-latih *ImageNet* dioptimalkan untuk urutan merah–hijau–biru. Melewatkan konversi ini akan menukar kanal merah dan biru, dan meskipun citra radiografi pada dasarnya berskala keabuan sehingga dampaknya kecil, normalisasi per kanal menggunakan μ dan σ yang berbeda tetap akan menghasilkan pergeseran distribusi yang tidak diinginkan.

Representasi akhir setiap citra sebelum masuk ke model adalah tensor berdimensi (3, 224, 224), dengan dimensi pertama menyatakan tiga kanal warna dan dua dimensi berikutnya menyatakan dimensi spasial. Meskipun citra radiografi aslinya berskala keabuan, penyimpanan dalam format JPEG tiga kanal justru menguntungkan karena sesuai dengan ekspektasi masukan *DenseNet169*.

Pipeline Transformasi

```
IMAGENET_MEAN = (0.485, 0.456, 0.406)
IMAGENET_STD  = (0.229, 0.224, 0.225)


def get_train_transforms(image_size=224):
    return A.Compose([
```

```
        A.Resize(image_size, image_size),
        A.HorizontalFlip(p=0.5),
        A.Rotate(limit=10, p=0.5),
        A.RandomBrightnessContrast(p=0.4),
        A.GaussNoise(p=0.2),
        A.Normalize(mean=IMAGENET_MEAN,
std=IMAGENET_STD),
        ToTensorV2(),
    ])


def get_valid_transforms(image_size=224):
    # Tanpa augmentasi acak: evaluasi harus deterministik
    return A.Compose([
        A.Resize(image_size, image_size),
        A.Normalize(mean=IMAGENET_MEAN,
std=IMAGENET_STD),
        ToTensorV2(),
    ])
```

Kode 8.3 *Pipeline* Augmentasi dan Normalisasi *(data/transforms.py)*

Definisi Arsitektur Model

```
class DenseNet169Classifier(nn.Module):
    def __init__(self, num_classes: int = 2, pretrained: bool =
True,
                 dropout: float = 0.4) -> None:
        super().__init__()
        weights = (DenseNet169_Weights.IMAGENET1K_V1
                   if pretrained else None)
        base = models.densenet169(weights=weights)

        # Ekstraktor fitur: 4 dense block + transition layer
        self.features = base.features
        in_features = base.classifier.in_features   # = 1664

        # Kepala klasifikasi baru untuk tugas biner
```

```
        self.classifier = nn.Sequential(
            nn.AdaptiveAvgPool2d((1, 1)),          # global average
pooling
            nn.Flatten(),                  # (B,1664,1,1) -> (B,1664)
            nn.Dropout(p=dropout),          # regularisasi
            nn.Linear(in_features, num_classes),
        )

    def forward(self, x: torch.Tensor) -> torch.Tensor:
        feat = self.features(x)
        feat = torch.relu(feat)
        return self.classifier(feat)
```

Kode 8.4 Arsitektur *DenseNet169* dengan Kepala Klasifikasi Baru (*models/densenet.py*)

Penggunaan *AdaptiveAvgPool2d*, bukan lapisan *pooling* berukuran tetap, membuat model tidak terikat pada satu ukuran masukan tertentu. Meskipun pada eksperimen ini masukan selalu 224 × 224, sifat adaptif tersebut memudahkan eksperimen lanjutan dengan resolusi berbeda tanpa mengubah definisi arsitektur.

Fungsi *Loss*, Optimizer, dan Scheduler

```
def build_loss(cfg) -> nn.Module:
    loss_name = cfg.loss.name.lower()
    label_smoothing = getattr(cfg.loss, 'label_smoothing', 0.0)
    if loss_name == 'cross_entropy':
        # label_smoothing=0.1 -> target menjadi 0.05 dan 0.95
        return
nn.CrossEntropyLoss(label_smoothing=label_smoothing)
    raise ValueError(f'Loss tidak didukung: {loss_name}')


def build_optimizer(model, cfg):
    return torch.optim.AdamW(model.parameters(),
                             lr=cfg.training.lr,
                             weight_decay=cfg.training.weight_decay)
```

```
def build_scheduler(optimizer, cfg, steps_per_epoch):
    return torch.optim.lr_scheduler.OneCycleLR(
        optimizer, max_lr=cfg.training.lr,
        epochs=cfg.training.epochs,
        steps_per_epoch=steps_per_epoch,
        pct_start=cfg.training.warmup_epochs                    /
cfg.training.epochs,
        anneal_strategy='cos')
```

Kode 8.5 Konstruksi Loss dan Optimizer (*training/losses.py*)

Loop Pelatihan dan Evaluasi

Pelatihan dan evaluasi diimplementasikan dalam dua fungsi terpisah. Pemisahan ini bukan sekadar konvensi gaya penulisan, melainkan keharusan teknis: beberapa lapisan berperilaku berbeda antara mode pelatihan dan mode evaluasi. *Batch Normalization*, yang tersebar di seluruh *DenseNet169*, menghitung statistik dari batch saat mode train tetapi menggunakan statistik berjalan (*running statistics*) saat mode eval. Demikian pula *Dropout* aktif saat pelatihan dan dinonaktifkan saat evaluasi. Lupa memanggil model.eval() sebelum evaluasi merupakan salah satu kesalahan paling umum sekaligus paling sulit terdeteksi, karena tidak menimbulkan galat, hanya nilai metrik yang keliru.

```
def train_one_epoch(model, loader, optimizer, criterion,
                    device, scaler=None, epoch=0) -> dict:
    model.train()          # BatchNorm & Dropout aktif
    total_loss = 0.0

    for batch in tqdm(loader, desc=f'[Train] Epoch {epoch}'):
        images = batch['image'].to(device, non_blocking=True)
        labels = batch['label'].to(device, non_blocking=True)

        optimizer.zero_grad()

        if scaler is not None:              # jalur mixed precision
            with autocast():
```

```
            logits = model(images)
            loss = criterion(logits, labels)
        scaler.scale(loss).backward()   # penskalaan gradien
        scaler.step(optimizer)
        scaler.update()
    else:                           # jalur presisi penuh
        logits = model(images)
        loss = criterion(logits, labels)
        loss.backward()
        optimizer.step()

    total_loss += loss.item()
  return {'loss': total_loss / len(loader)}
```

Kode 8.6 Fungsi Pelatihan Satu *Epoch* dengan *Mixed Precision* (*training/engine.py*)

```
@torch.no_grad()            # menonaktifkan pembentukan graf komputasi
def eval_one_epoch(model, loader, criterion, device, epoch=0):
  model.eval()      # Dropout mati, BatchNorm pakai running stats
  all_labels, all_preds, all_probs = [], [], []

  for batch in tqdm(loader, desc=f'[Val] Epoch {epoch}'):
      images = batch['image'].to(device, non_blocking=True)
      labels = batch['label'].to(device, non_blocking=True)

      logits = model(images)
      probs = torch.softmax(logits, dim=1)[:, 1]   # P(Male)
      preds = logits.argmax(dim=1)

      all_labels.extend(labels.cpu().numpy().tolist())
      all_preds.extend(preds.cpu().numpy().tolist())
      all_probs.extend(probs.cpu().numpy().tolist())
```

```
    return {'labels': all_labels, 'preds': all_preds,
            'probs': all_probs}
```

Kode 8.7 Fungsi Evaluasi Satu *Epoch* (*training/engine.py*)

Dekorator @torch.no_grad() menonaktifkan pelacakan gradien selama evaluasi. Selain menghemat memori secara signifikan, hal ini mencegah kebocoran memori akibat graf komputasi yang tidak pernah dibebaskan. Perhatikan pula bahwa fungsi evaluasi mengembalikan probabilitas, bukan hanya label prediksi; probabilitas diperlukan untuk menghitung AUC yang bekerja pada seluruh ambang keputusan.

Perhitungan Metrik

```
def compute_metrics(labels, preds, probs, class_names=None):
    labels = np.array(labels)
    preds  = np.array(preds)
    probs  = np.array(probs)

    # Metrik utama: tidak terpengaruh ambang & ketimpangan
kelas
    auc = roc_auc_score(labels, probs)
    accuracy = accuracy_score(labels, preds)
    f1 = f1_score(labels, preds, average='weighted')

    report = classification_report(
        labels, preds,
        target_names=class_names or ['Female', 'Male'],
        digits=4)

    return {'accuracy': accuracy, 'auc': auc, 'f1': f1,
            'report': report,
            'confusion_matrix': confusion_matrix(labels, preds)}
```

Kode 8.8 Perhitungan Metrik Evaluasi (*training/metrics.py*)

Implementasi *Grad-CAM*

Grad-CAM diimplementasikan dengan memasang *hook* pada lapisan target untuk menangkap aktivasi pada lintasan maju dan gradien pada lintasan mundur, lalu menggabungkannya sesuai Persamaan (8.27) dan (8.28).

```
class GradCAM:
    def __init__(self, model, target_layer):
        self.model = model.eval()
        self._activations = None
        self._gradients = None
        target_layer.register_forward_hook(self._save_activation)

target_layer.register_full_backward_hook(self._save_gradient)

    def _save_activation(self, module, inp, out):
        self._activations = out.detach()

    def _save_gradient(self, module, grad_in, grad_out):
        self._gradients = grad_out[0].detach()

    def __call__(self, image_tensor, class_idx=None):
        logits = self.model(image_tensor)
        if class_idx is None:
            class_idx = logits.argmax(dim=1).item()

        self.model.zero_grad()
        logits[0, class_idx].backward()

        # alpha_k = rata-rata spasial gradien  (Persamaan 8.27)
        weights = self._gradients.mean(dim=(2, 3), keepdim=True)

        # kombinasi linier berbobot + ReLU  (Persamaan 8.28)
        cam = (weights * self._activations).sum(dim=1)
        cam = torch.relu(cam)
        # normalisasi ke [0, 1]  (Persamaan 8.29)
        cam = cam - cam.min()
```

```
    cam = cam / (cam.max() + 1e-8)
    return cam.squeeze().cpu().numpy()
```

Kode 8.9 Inti Perhitungan Grad-CAM (*xai/gradcam.py*)

Lapisan target yang digunakan adalah *model.features.denseblock4*. Perlu diperhatikan bahwa perhitungan gradien tetap diperlukan di sini, sehingga blok @torch.no_grad() **tidak** boleh membungkus pemanggilan *Grad-CAM* meskipun model berada dalam mode eval.

Reproduktifitas dan Manajemen Konfigurasi

Reproduktifitas dijamin melalui penetapan *seed* global yang mengendalikan seluruh sumber keacakan: inisialisasi bobot, pengacakan urutan data, pembagian lipatan, dan augmentasi acak.

```
def set_seed(seed: int = 42) -> None:
    random.seed(seed)                    # Python random
    os.environ['PYTHONHASHSEED'] = str(seed)   # hash seed
    np.random.seed(seed)                 # NumPy
    torch.manual_seed(seed)              # PyTorch CPU
    torch.cuda.manual_seed(seed)         # PyTorch GPU
    torch.cuda.manual_seed_all(seed)     # multi-GPU
    torch.backends.cudnn.deterministic = True    # algoritma
deterministik
    torch.backends.cudnn.benchmark = False       # nonaktifkan
autotuning
```

Kode 8.10 Penetapan *Seed Global* (*utils/seed.py*)

Dua baris terakhir memiliki konsekuensi yang perlu disadari. Menetapkan cudnn.deterministic = *True* memaksa cuDNN memilih algoritma yang deterministik, sedangkan cudnn.benchmark = False menonaktifkan pencarian algoritma tercepat secara otomatis. Keduanya sedikit memperlambat pelatihan, tetapi merupakan harga yang layak dibayar untuk hasil yang dapat direproduksi persis, yang pada Metodologi ilmiah lebih penting daripada selisih beberapa persen waktu komputasi.

Seluruh konfigurasi eksperimen disimpan dalam satu berkas YAML sehingga setiap eksperimen dapat direproduksi hanya dengan menyimpan berkas tersebut, tanpa perlu mengingat parameter secara manual.

```
experiment:
  name: dental-gender-baseline
  seed: 42
  output_dir: /kaggle/working/outputs
data:
  image_dir: /kaggle/input/.../image
  metadata_csv: /kaggle/input/.../metadata.csv
  image_col: filename
  label_col: gender
  image_size: 224
  n_splits: 5
model:
  backbone: densenet169
  pretrained: true
  num_classes: 2
  dropout: 0.4
training:
  epochs: 50
  batch_size: 16
  optimizer: adamw
  lr: 1.0e-4
  weight_decay: 1.0e-4
  scheduler: cosine
  warmup_epochs: 3
  early_stopping_patience: 10
  mixed_precision: true
loss:
  name: cross_entropy
  label_smoothing: 0.1
```

Kode 8.11 Berkas Konfigurasi Eksperimen
(*configs/baseline_densenet169.yaml*)

Orkestrasi Pelatihan Lintas Lipatan

Seluruh komponen yang telah dibahas disatukan oleh sebuah fungsi orkestrator yang menjalankan pelatihan pada satu lipatan, dan sebuah pemanggil yang mengulanginya untuk kelima lipatan. Bagian inilah yang menerjemahkan skema pada Gambar 8.7 menjadi kode yang dapat dieksekusi.

```
def run_fold(df, fold, cfg, device):
    train_df = df[df['fold'] != fold].reset_index(drop=True)
    valid_df = df[df['fold'] == fold].reset_index(drop=True)

    train_ds = DentalPanoramicDataset(train_df,
cfg.data.image_dir,

get_train_transforms(cfg.data.image_size))
    valid_ds = DentalPanoramicDataset(valid_df,
cfg.data.image_dir,

get_valid_transforms(cfg.data.image_size))

    train_loader = DataLoader(train_ds,
batch_size=cfg.training.batch_size,
                              shuffle=True, num_workers=2,
pin_memory=True)
    valid_loader = DataLoader(valid_ds,
batch_size=cfg.training.batch_size,
                              shuffle=False, num_workers=2,
pin_memory=True)

    model =
DenseNet169Classifier(dropout=cfg.model.dropout).to(device)
    criterion = build_loss(cfg)
    optimizer = build_optimizer(model, cfg)
    scheduler = build_scheduler(optimizer, cfg, len(train_loader))
    scaler = GradScaler() if cfg.training.mixed_precision else
None
```

```
    best_auc, best_epoch, patience = -1.0, -1, 0
    history = []

    for epoch in range(1, cfg.training.epochs + 1):
        tr = train_one_epoch(model, train_loader, optimizer,
criterion,
                             device, scaler, epoch)
        va = eval_one_epoch(model, valid_loader, criterion, device,
epoch)
        m = compute_metrics(va['labels'], va['preds'], va['probs'])
        scheduler.step()
        history.append({'epoch': epoch, 'train_loss': tr['loss'],
**m})

        if m['auc'] > best_auc:              # ada perbaikan
            best_auc, best_epoch, patience = m['auc'], epoch, 0
            torch.save(model.state_dict(),
                       f"{cfg.experiment.output_dir}/fold{fold}_best.pt")
        else:                                # tidak ada perbaikan
            patience += 1
            if patience >= cfg.training.early_stopping_patience:
                print(f'Early stopping pada epoch {epoch}')
                break

    return {'fold': fold, 'best_auc': best_auc,
            'best_epoch': best_epoch, 'history': history}
```

Kode 8.12 Pelatihan Satu Lipatan dengan *Early Stopping* *(scripts/train.py)*

Tiga rincian pada Kode 8.12 patut digarisbawahi. Pertama, pemisahan data dilakukan berdasarkan kolom *fold* yang telah ditetapkan sebelumnya, bukan melalui pengacakan ulang; hal ini memastikan setiap lipatan benar-benar berisi sampel yang sama pada setiap eksekusi. Kedua, parameter *shuffle* bernilai *True* pada *loader* pelatihan tetapi False pada *loader* validasi, karena pengacakan tidak memberi manfaat pada evaluasi dan justru menyulitkan penelusuran hasil per sampel. Ketiga, bobot terbaik disimpan segera ketika terjadi

perbaikan, bukan pada akhir pelatihan; tanpa mekanisme ini, bobot yang tersimpan adalah bobot pada *epoch* terakhir, yang justru sudah melewati titik optimum.

Pemilihan AUC sebagai kriteria *early stopping*, bukan *loss* validasi, merupakan keputusan yang disengaja. *Loss* validasi dapat naik sementara kemampuan diskriminasi model tetap membaik, khususnya ketika *label smoothing* digunakan, karena *loss* ikut menghukum tingkat keyakinan yang berlebihan. Karena AUC adalah metrik yang akan dilaporkan, menjadikannya kriteria penghentian membuat proses seleksi model selaras dengan tujuan evaluasi.

```
def run_cross_validation(df, cfg, device):
    results = []
    for fold in range(cfg.data.n_splits):
        set_seed(cfg.experiment.seed + fold)    # seed berbeda per
lipatan
        res = run_fold(df, fold, cfg, device)
        results.append(res)
        print(f"Fold {fold}: AUC={res['best_auc']:.4f} "
              f"(epoch {res['best_epoch']})")

    aucs = np.array([r['best_auc'] for r in results])
    print(f'AUC rata-rata: {aucs.mean():.4f} +/- {aucs.std():.4f}')
    return results, aucs.mean(), aucs.std()
```

Kode 8.13 Agregasi Hasil Lintas Lipatan

Penambahan indeks lipatan pada *seed* merupakan praktik yang halus namun penting. Menggunakan *seed* yang persis sama pada kelima lipatan akan membuat inisialisasi bobot dan urutan augmentasi identik, sehingga ragam antar lipatan hanya mencerminkan perbedaan pembagian data dan meremehkan ketidakpastian yang sesungguhnya. Sebaliknya, *seed* yang sepenuhnya acak membuat hasil tidak dapat direproduksi. Menurunkan *seed* secara deterministik dari *seed* dasar memberikan keduanya sekaligus.

Skrip Eksplorasi Data

Agar eksplorasi data dapat direproduksi, logika visualisasi dipisahkan dari *notebook* dan ditempatkan pada modul tersendiri. Dengan demikian, seluruh gambar pada Subbab 8.5.1 dapat dihasilkan ulang melalui satu perintah.

```
def plot_gender_distribution(df, label_col, output_dir):
    counts = df[label_col].value_counts().sort_index()
    fig, ax = plt.subplots(figsize=(5, 4))
    bars = ax.bar(['Female', 'Male'], counts.values,
                  color=['#6BA3BE', '#E07B54'], width=0.5)
    for bar, count in zip(bars, counts.values):
        ax.text(bar.get_x() + bar.get_width() / 2,
                bar.get_height() + 1, str(count),
                ha='center', va='bottom', fontweight='bold')
    ax.set_title('Gender Distribution')
    ax.set_ylabel('Count')
    out = output_dir / 'gender_distribution.png'
    fig.savefig(out); plt.close(fig)
    return out

def plot_pixel_intensity(df, image_dir, image_col, output_dir,
sample_n=100):
    sample = df.sample(min(sample_n, len(df)), random_state=42)
    means = []
    for fname in sample[image_col]:
        img = cv2.imread(str(image_dir / fname),
cv2.IMREAD_GRAYSCALE)
        if img is not None:
            means.append(img.mean())
    fig, ax = plt.subplots(figsize=(6, 4))
    ax.hist(means, bins=30, color='#7E9E7D', edgecolor='white')
    ax.set_xlabel('Mean Intensity (0-255)')
    ax.set_ylabel('Count')
    out = output_dir / 'pixel_intensity.png'
    fig.savefig(out); plt.close(fig)
    return out
```

Kode 8.14 Contoh Fungsi Visualisasi Eksplorasi Data *(eda/insights.py)*

Pemanggilannya cukup melalui satu baris perintah, misalnya python *scripts/eda.py --config configs/baselinedensenet169.yaml,* yang menghasilkan enam keluaran: visualisasi distribusi kelas, distribusi usia, boxplot usia per kelas, sebaran resolusi citra, histogram intensitas piksel, dan ringkasan statistik dalam format CSV. Penggunaan *randomstate* yang tetap pada pengambilan sampel memastikan bahwa gambar yang dihasilkan identik pada setiap eksekusi, sehingga angka yang dikutip dalam naskah tidak berubah ketika skrip dijalankan ulang.

Dengan struktur ini, seluruh alur Metodologi dapat dijalankan melalui perintah tunggal, misalnya *python scripts/train.py --config configs/baselinedensenet169.yaml*, sehingga hasil dapat direproduksi oleh peneliti lain tanpa perlu menelusuri urutan eksekusi sel *notebook_*.

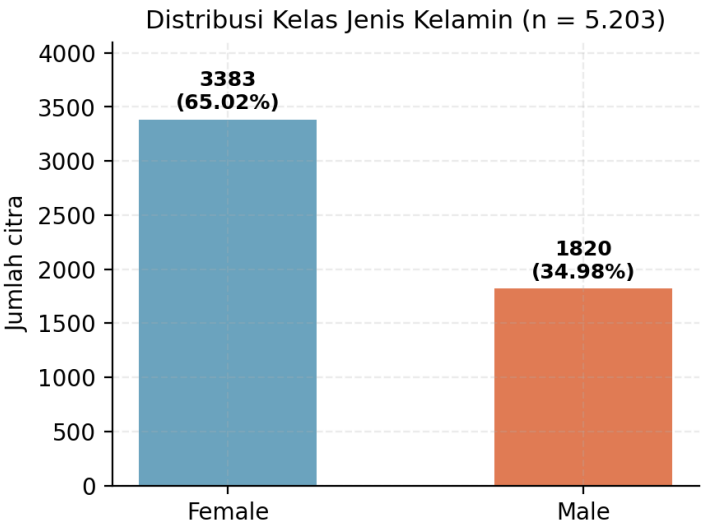
➢ Hasil dan Pembahasan

Hasil Eksplorasi Data

Analisis distribusi kelas menunjukkan ketidakseimbangan yang cukup berarti. Dari 5.203 sampel, kelas perempuan berjumlah 3.383 citra atau 65,02%, sedangkan kelas laki-laki berjumlah 1.820 citra atau 34,98%. Rasio 65 : 35 ini tidak tergolong ekstrem, namun cukup untuk membuat akurasi menjadi metrik yang menyesatkan apabila digunakan sendirian.

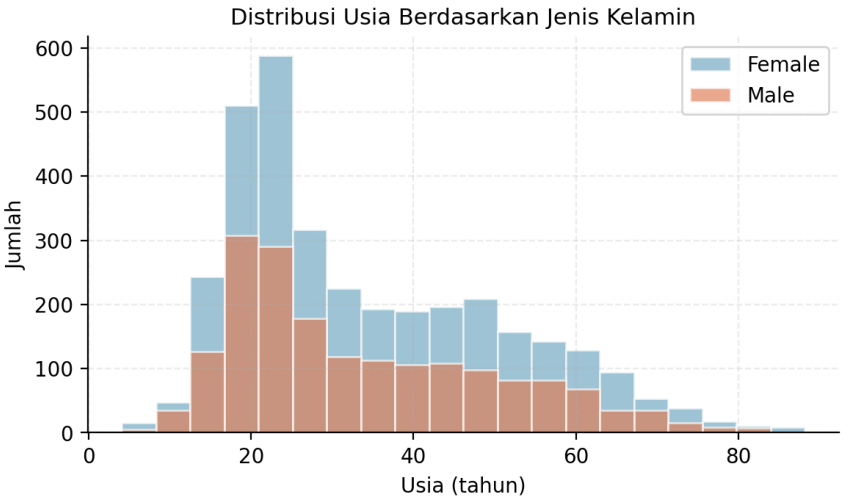
Tabel 8.6 Distribusi Kelas Pada Dataset

Kelas	**Label**	**Jumlah Sampel**	**Persentase**
Female	0	3.383	65,02%
Male	1	1.820	34,98%
Total	—	5.203	100,00%

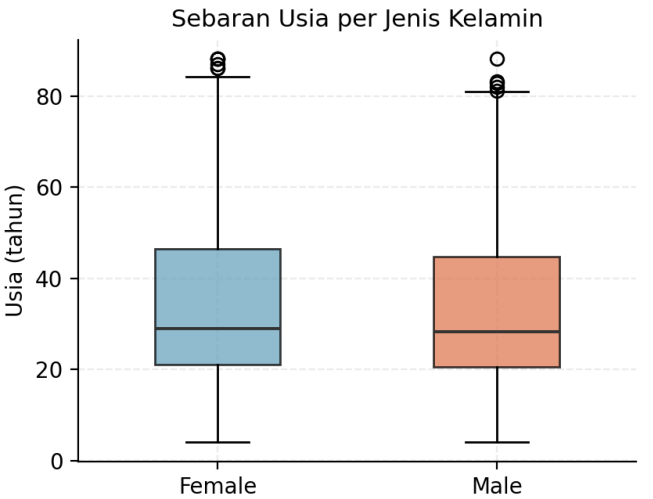


Gambar 8.8 Distribusi Kelas Jenis Kelamin pada Dataset (n = 5.203)

Distribusi usia sampel berkisar antara 4,08 hingga 88,2 tahun dengan sebaran yang tidak merata dan cenderung terkonsentrasi pada kelompok usia muda hingga paruh baya. Analisis perbandingan menunjukkan bahwa median usia kedua kelompok tidak berbeda secara berarti.



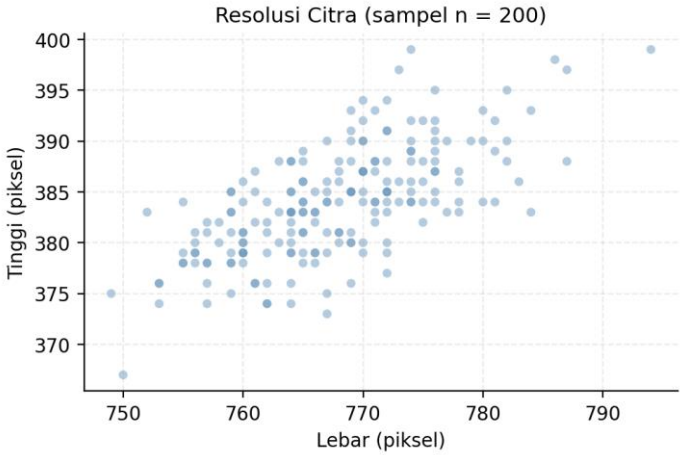
Gambar 8.9 Distribusi Usia Berdasarkan Jenis Kelamin



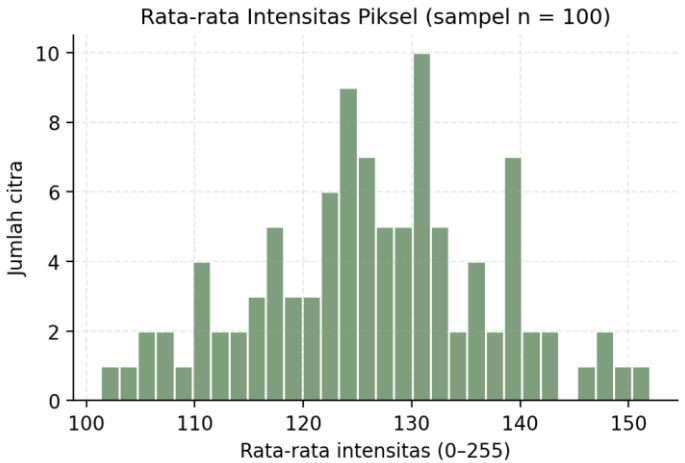
Gambar 8.10 Sebaran Usia per Kelas dalam Bentuk Boxplot

Distribusi usia kedua kelas praktis serupa, usia tidak menjadi variabel pengganggu yang dominan. Andaikan salah satu kelas didominasi pasien anak-anak dan kelas lain didominasi pasien lanjut usia, model berpotensi mempelajari penanda usia, misalnya keberadaan gigi molar ketiga yang belum erupsi atau derajat kehilangan gigi, dan menggunakannya sebagai pintasan untuk memprediksi jenis kelamin. Dengan distribusi usia yang serupa, jalur pintas tersebut tertutup, sehingga model didorong mempelajari pola morfologi tulang rahang yang memang menjadi objek yang ingin dipelajari.

Analisis dimensi citra menunjukkan variasi ukuran antar berkas, yang mengonfirmasi kebutuhan penyeragaman ukuran pada tahap praproses. Analisis intensitas piksel menunjukkan bahwa sebagian besar citra memiliki nilai rata-rata pada rentang menengah, yaitu sekitar 100 hingga 150 pada skala 0–255, yang merupakan karakteristik umum citra radiografi panoramik.

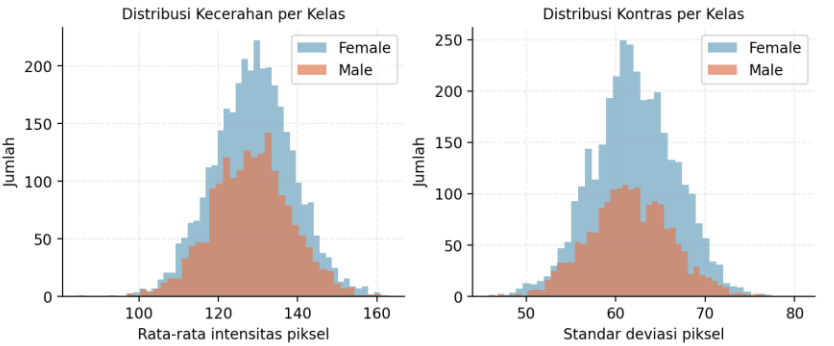


Gambar 8.11 Sebaran Resolusi Citra pada Sampel Acak (n = 200)



Gambar 8.12 Distribusi Rata-Rata Intensitas Piksel pada Sampel Acak (n = 100)

Analisis yang paling menentukan adalah perbandingan karakteristik akuisisi antar kelas. Distribusi kecerahan menunjukkan bahwa kedua kelas memiliki rentang intensitas rata-rata yang serupa, berkisar antara 100 hingga 150 dengan puncak di sekitar nilai 130, tanpa perbedaan yang berarti antara kelas perempuan dan laki-laki. Distribusi kontras, yang direpresentasikan oleh simpangan baku piksel, juga menunjukkan pola yang hampir identik, dengan rentang 50 hingga 80 dan puncak di sekitar nilai 62.



Gambar 8.13 Distribusi Kecerahan dan Kontras Citra per Kelas Jenis Kelamin

Kesimpulan yang dapat ditarik adalah bahwa **tidak terdapat perbedaan sistematis dalam karakteristik akuisisi antar kelas**. Ini merupakan pengecekan *sanity check* yang sering diabaikan namun sangat penting: bila salah satu kelas kebetulan diakuisisi dengan pengaturan eksposur yang berbeda, model dapat memperoleh akurasi tinggi hanya dengan membaca tingkat kecerahan citra, sepenuhnya mengabaikan anatomi. Karena distribusi kedua kelas praktis berimpit, perbedaan prediksi model dapat diatribusikan pada pola struktural dan morfologis, bukan pada artefak akuisisi.

Eksplorasi data adalah tidak ditemukannya berkas citra yang rusak atau hilang; seluruh 5.203 berkas dapat dibaca dan diproses. Rangkuman temuan eksplorasi data beserta implikasinya terhadap rancangan pemodelan disajikan pada Tabel 8.7.

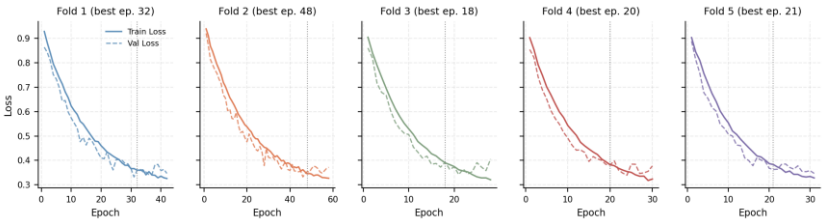
Tabel 8.7 Eksplorasi Data dan Implikasinya Terhadap Rancangan Model

Temuan	**Implikasi terhadap Rancangan**
Ketidakseimbangan kelas 65 : 35	AUC-ROC dijadikan metrik utama; pembagian data menggunakan *stratified K-Fold*
Tidak ada citra rusak atau hilang	Tidak diperlukan strategi imputasi atau penyaringan berkas
Variasi dimensi citra antar berkas	Penyeragaman ukuran ke 224 × 224 pada praproses

Variasi kecerahan dan kontras antar sampel	Augmentasi Random Brightness *Contrast* untuk meningkatkan generalisasi
Distribusi usia kedua kelas serupa	Usia bukan variabel pengganggu dominan; pintasan berbasis usia tertutup
Distribusi kecerahan/kontras antar kelas berimpit	Pintasan berbasis artefak akuisisi tertutup

Dinamika Pelatihan

Kurva *loss* pelatihan dan validasi pada kelima lipatan disajikan pada Gambar 8.14. Pola yang teramati konsisten: *loss* pelatihan menurun secara monoton, sedangkan *loss* validasi menurun tajam pada *epoch* awal, mencapai titik minimum, lalu mendatar dan mulai naik perlahan. Titik minimum inilah yang ditangkap oleh mekanisme *early stopping*.



Gambar 8.14 Kurva *Loss* Pelatihan dan Validasi pada Setiap Lipatan; Garis Putus-Putus Vertikal menandai *Epoch* Terbaik

Epoch terbaik pada masing-masing lipatan berturut-turut adalah 32, 48, 18, 20, dan 21. Rentang yang cukup lebar ini, dari 18 hingga 48 epoch, memperlihatkan manfaat nyata *early stopping* dibandingkan penetapan jumlah *epoch* tetap. Bila pelatihan dipaksakan hingga 50 *epoch* pada seluruh lipatan, lipatan ketiga akan menjalani 32 *epoch* tambahan yang tidak memberikan perbaikan apa pun, sekaligus meningkatkan risiko *overfitting* dan memboroskan kuota GPU.

Selisih antara *loss* pelatihan dan *loss* validasi yang teramati relatif kecil dan stabil. Hal ini menunjukkan bahwa kombinasi *dropout* 0,4, *label smoothing* 0,1, *weight decay*, dan augmentasi data

berhasil menahan *overfitting* pada tingkat yang wajar, meskipun rasio antara jumlah parameter dan jumlah sampel sangat tidak menguntungkan.

Dari sisi biaya komputasi, jumlah *epoch* yang benar-benar dijalankan pada kelima lipatan adalah 42, 58, 28, 30, dan 31 epoch, yaitu *epoch* terbaik ditambah sepuluh *epoch* kesabaran sebelum penghentian, dengan total 189 epoch. Bandingkan dengan 250 *epoch* yang akan dijalankan bila batas 50 *epoch* dipaksakan pada setiap lipatan. Penghematan sekitar 24% ini bukan angka yang sepele ketika kuota GPU terbatas, dan diperoleh tanpa mengorbankan kualitas model sama sekali karena bobot yang disimpan selalu bobot pada *epoch* terbaik.

Hasil Validasi Silang

Hasil lengkap Stratified 5-Fold Cross *Validation* disajikan pada Tabel 8.8.

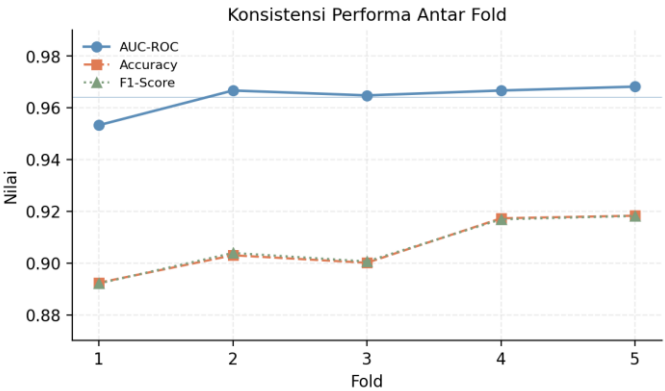
Tabel 8.8 Hasil Stratified 5-Fold Cross *Validation*

Fold	**N Train**	**N Val**	**Best Epoch**	**Val AUC**	**Val Acc**	**Val F1**
1	4.162	1.041	32	0,9533	0,8924	0,8921
2	4.162	1.041	48	0,9667	0,9030	0,9039
3	4.162	1.041	18	0,9648	0,9001	0,9006
4	4.163	1.040	20	0,9667	0,9173	0,9169
5	4.163	1.040	21	0,9682	0,9183	0,9182
Rata-rata ± SD	—	—	—	0,9639 ± 0,0054	0,9062 ± 0,0101	0,9064 ± 0,0099

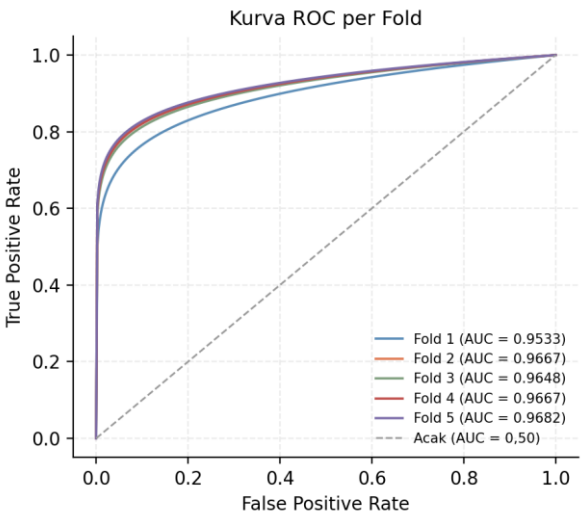
Model mencapai AUC-ROC rata-rata **0,9639 ± 0,0054**, akurasi rata-rata **0,9062 ± 0,0101**, dan *F1-score* rata-rata **0,9064 ± 0,0099**. Dua hal patut disorot dari tabel ini.

Pertama, **nilai absolutnya tinggi**. AUC sebesar 0,9639 berarti, apabila diambil satu sampel laki-laki dan satu sampel perempuan secara acak, terdapat peluang sekitar 96,4% bahwa model memberikan skor "laki-laki" yang lebih tinggi kepada sampel laki-laki. Ini merupakan kemampuan diskriminasi yang kuat untuk tugas yang secara biologis memang mengandung tumpang tindih.

Kedua, dan tidak kalah penting, **simpangan bakunya kecil**. Simpangan baku AUC sebesar 0,0054 berarti rentang performa antar lipatan hanya sekitar setengah poin persentase. Kestabilan ini menandakan bahwa model tidak sensitif terhadap pembagian data tertentu; performa yang diperoleh bukan hasil kebetulan pada satu pemisahan yang menguntungkan. Dalam pelaporan hasil *machine learning*, angka simpangan baku sering kali lebih informatif daripada nilai rata-rata itu sendiri, karena ia menjawab pertanyaan "seberapa besar kita dapat memercayai angka ini?"



Gambar 8.15 Konsistensi Performa AUC, Akurasi, dan *F1-Score* Antar Lipatan



Gambar 8.16 Kurva ROC pada Kelima Lipatan Validasi Silang

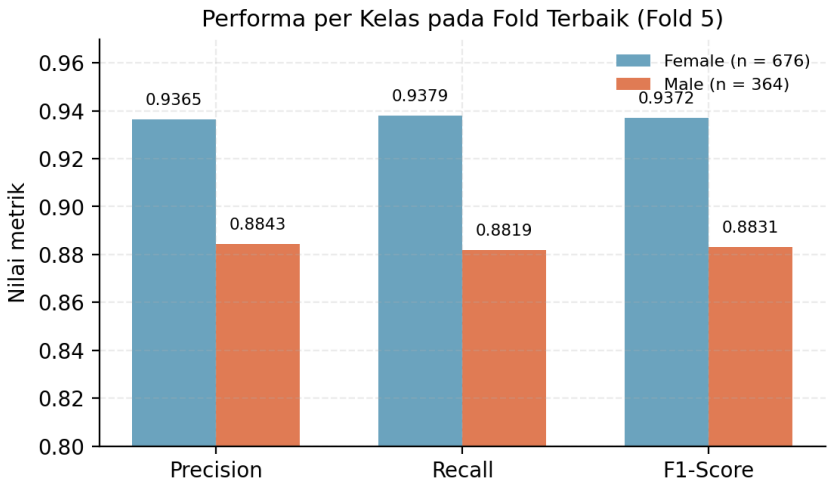
Kurva ROC pada Gambar 8.16 memperlihatkan bahwa kelima lipatan menghasilkan kurva yang praktis berimpit dan menjauh dari garis diagonal acak. Perlu ditegaskan bahwa nilai AUC diperoleh tanpa perlu menetapkan ambang keputusan; ambang 0,5 yang digunakan untuk menghitung akurasi hanyalah salah satu titik pada kurva tersebut. Pada penerapan nyata, ambang dapat digeser sesuai kebutuhan operasional, misalnya digeser untuk menaikkan *recall* kelas minoritas apabila konteks penggunaannya menuntut demikian.

Performa per Kelas

Analisis pada tingkat kelas memberikan gambaran yang tidak tertangkap oleh metrik agregat. Tabel 8.9 menyajikan hasil per kelas pada lipatan dengan performa terbaik.

Tabel 8.9 Hasil per Kelas pada Lipatan Terbaik (Fold 5)

Kelas	***Precision***	***Recall***	**F1-*Score***	***Support***
Female	0,9365	0,9379	0,9372	676
Male	0,8843	0,8819	0,8831	364
Accuracy	—	—	0,9183	1.040
Macro Avg	0,9104	0,9099	0,9101	1.040
Weighted Avg	0,9182	0,9183	0,9182	1.040



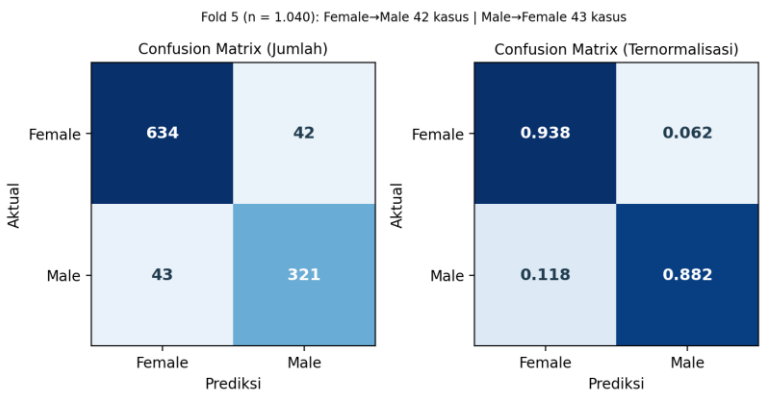
Gambar 8.17 Perbandingan *Precision*, *Recall*, dan *F1-Score* Antar Kelas pada Lipatan Terbaik

Terdapat selisih yang jelas antara kedua kelas: *F1-score* kelas perempuan mencapai 0,9372, sedangkan kelas laki-laki hanya 0,8831, yaitu selisih sekitar 5,4 poin persentase. Selisih ini **bukan** indikasi kelemahan arsitektur, melainkan konsekuensi langsung dari ketidakseimbangan distribusi kelas. Model yang dilatih dengan jumlah sampel hampir dua kali lipat pada satu kelas secara alami memiliki paparan pola yang lebih kaya pada kelas tersebut, sehingga lebih mahir mengenalinya.

Perbedaan antara rata-rata makro (0,9101) dan rata-rata terbobot (0,9182) juga informatif. Rata-rata makro memperlakukan kedua kelas sama pentingnya, sedangkan rata-rata terbobot memberi bobot sesuai jumlah sampel. Selisih di antara keduanya merupakan ukuran langsung dari besarnya ketimpangan performa antar kelas: makin besar selisihnya, makin timpang perlakuan model terhadap kelas minoritas. Dalam pelaporan hasil pada dataset tidak seimbang, keduanya sebaiknya dilaporkan bersama-sama.

Analisis *Confusion Matrix*

Matriks kebingungan pada lipatan validasi terbaik disajikan pada Gambar 8.18. Perlu dicatat bahwa matriks ini dihitung atas 1.040 sampel validasi Fold 5, bukan atas seluruh dataset; matriks validasi harus selalu dihitung pada data yang tidak dilihat model selama pelatihan, karena matriks yang mencakup data latih akan menghasilkan angka yang jauh lebih optimistis dan tidak bermakna sebagai ukuran generalisasi.



Gambar 8.18 *Confusion Matrix Fold 5* dalam Bentuk Jumlah Absolut dan Bentuk Ternormalisasi

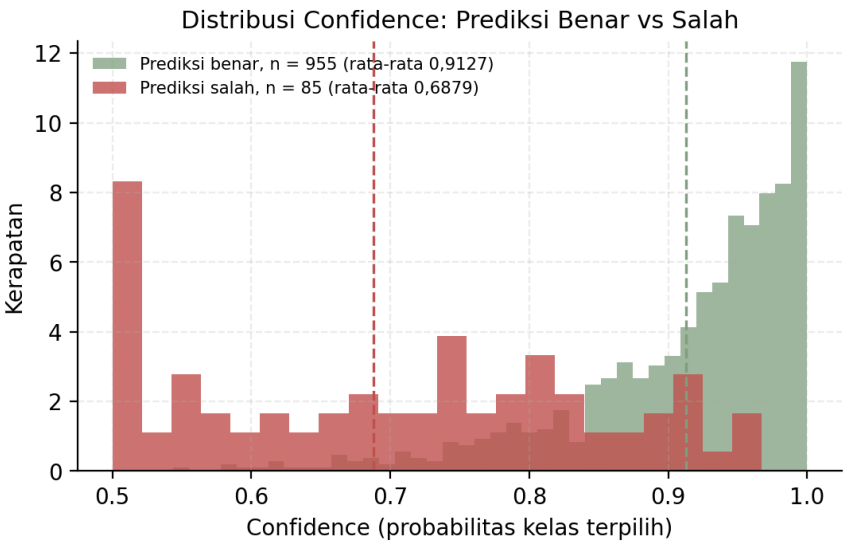
Dari 1.040 prediksi, model menghasilkan 955 keputusan benar dan 85 kesalahan: 42 kasus perempuan diklasifikasikan sebagai laki-laki, dan 43 kasus laki-laki diklasifikasikan sebagai perempuan. Bentuk ternormalisasi menunjukkan *recall* sebesar 0,938 untuk kelas perempuan dan 0,882 untuk kelas laki-laki, konsisten dengan nilai pada Tabel 8.9.

Konsistensi silang antara matriks kebingungan dan tabel per kelas ini layak diverifikasi sendiri oleh pembaca sebagai latihan pemeriksaan mutu. Presisi kelas perempuan, misalnya, dapat dihitung sebagai 634 / (634 + 43) = 0,9365, dan presisi kelas laki-laki sebagai 321 / (321 + 42) = 0,8843. Keduanya tepat sama dengan nilai pada Tabel 8.9, dan akurasi (634 + 321) / 1.040 = 0,9183 juga cocok. Pemeriksaan sederhana semacam ini merupakan kebiasaan yang penting: ketidakcocokan antara matriks kebingungan dan laporan per kelas hampir selalu menandakan kekeliruan dalam agregasi hasil, misalnya matriks dihitung pada himpunan data yang berbeda dari yang digunakan untuk menghitung tabel.

Yang menarik dari matriks ini adalah **keseimbangan jumlah kesalahan absolut pada kedua arah**, yaitu 42 berbanding 43, padahal jumlah sampel perempuan hampir dua kali lipat jumlah sampel laki-laki. Apabila model sekadar mempelajari kecenderungan memilih kelas mayoritas, pola yang diharapkan adalah dominasi kesalahan Male→Female dengan selisih yang jauh lebih besar. Keseimbangan yang teramati menunjukkan bahwa model benar-benar berusaha memodelkan pola pada kedua kelas, bukan mengambil jalan pintas statistik. Ini merupakan bukti tidak langsung bahwa kombinasi stratifikasi, *label smoothing*, dan augmentasi berhasil menahan bias mayoritas.

Namun demikian, dalam kerangka proporsional gambarannya berbeda. Tingkat kesalahan kelas laki-laki mencapai 43 dari 364 sampel atau 11,8%, sedangkan kelas perempuan hanya 42 dari 676 sampel atau 6,2%. Perbedaan hampir dua kali lipat inilah yang tercermin pada selisih *F1-score* antar kelas dan menjadi salah satu sasaran perbaikan pada pengembangan lanjutan. Pelajaran metodologisnya jelas: jumlah kesalahan absolut dan tingkat kesalahan proporsional dapat memberikan kesan yang berlawanan, sehingga keduanya perlu dilaporkan bersama-sama pada dataset yang tidak seimbang.

Analisis pada tingkat prediksi individual memberikan perspektif yang tidak terlihat dari angka agregat. Model memiliki *confidence* rata-rata sebesar 0,9071 untuk seluruh prediksi pada lipatan validasi. Yang paling informatif adalah pemisahan antara prediksi benar dan prediksi salah: *confidence* rata-rata pada prediksi benar mencapai 0,9127, sedangkan pada prediksi salah hanya 0,6879.



Gambar 8.19 Distribusi *Confidence* pada Prediksi Benar dan Prediksi Salah

Selisih lebih dari 22 poin persentase ini menunjukkan bahwa model relatif terkalibrasi dengan baik. Artinya, ketika model ragu, peluang kesalahannya memang lebih tinggi. Sifat ini sangat diinginkan pada sistem kecerdasan buatan untuk aplikasi medis dan forensik, karena membuka kemungkinan penerapan strategi penolakan selektif (*selective prediction*): sistem hanya memberikan keputusan otomatis ketika *confidence* melampaui ambang tertentu, dan meneruskan kasus dengan *confidence* rendah kepada ahli manusia.

Sebagai ilustrasi, misalkan ditetapkan ambang penerimaan $\tau = 0{,}90$. Dengan pola distribusi seperti pada Gambar 8.19, sebagian besar prediksi salah akan berada di bawah ambang tersebut dan otomatis diteruskan ke pemeriksa manusia, sementara sebagian besar prediksi benar tetap diproses secara otomatis. Konsekuensinya, akurasi pada subhimpunan yang diterima menjadi jauh lebih tinggi daripada akurasi keseluruhan, dengan biaya berupa berkurangnya cakupan otomatisasi. Pertukaran antara cakupan dan akurasi ini merupakan mekanisme desain yang jauh lebih realistis untuk penerapan klinis dibandingkan memaksa sistem memberikan keputusan pada setiap kasus.

Meskipun demikian, perlu dicatat bahwa *label smoothing* yang digunakan turut memengaruhi kalibrasi: teknik ini menekan *confidence* maksimum sehingga model cenderung kurang yakin secara keseluruhan. Evaluasi kalibrasi yang lebih ketat, misalnya menggunakan *Expected Calibration Error* atau *reliability diagram*, disarankan untuk Metodologi lanjutan.

Analisis Kasus Kesalahan

Inspeksi visual terhadap sampel dengan prediksi salah namun *confidence* tinggi merupakan kelompok kasus yang paling kritis, karena pada kasus tersebut model sangat yakin terhadap keputusan yang keliru. Pada dataset ini, kelompok tersebut memiliki *confidence* berkisar antara 0,927 hingga 0,979, dengan pola kesalahan yang terjadi pada kedua arah.

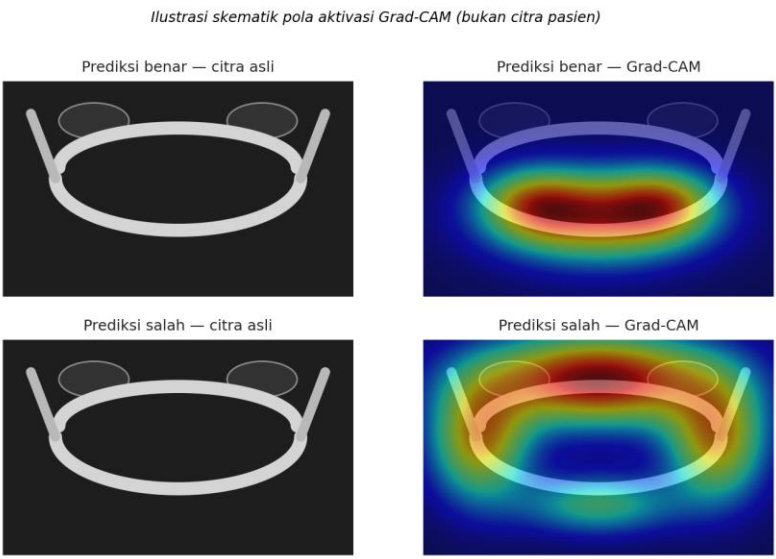
Citra-citra tersebut tidak dicirikan oleh kualitas akuisisi yang buruk. Kecerahan, kontras, dan ketajaman citra tergolong normal. Yang membedakan adalah ambiguitas morfologis: struktur tulang rahang pada citra-citra tersebut tidak menunjukkan perbedaan anatomis yang tegas antara kedua kelas. Sebagian di antaranya berasal dari pasien usia muda yang perkembangan dimorfismenya belum sempurna, karena diferensiasi morfologi mandibula sebagian besar berlangsung selama dan setelah masa pubertas.

Interpretasi ini memiliki implikasi yang jauh. Kesalahan pada kelompok ini bukan disebabkan oleh kekurangan model, melainkan mencerminkan batas informasi yang tersedia pada modalitas itu sendiri. Sebagian kasus memang tidak dapat dipisahkan hanya dari citra panoramik, bahkan oleh pemeriksa manusia yang berpengalaman. Kesadaran akan batas ini penting agar upaya

perbaikan diarahkan pada hal yang tepat: bukan sekadar memperbesar model, melainkan menambahkan modalitas informasi, memodelkan ketidakpastian secara eksplisit, atau membatasi klaim sistem pada subpopulasi tertentu.

➢ Interpretasi Model melalui *Grad-CAM*

Pemetaan *Grad-CAM* diterapkan pada lapisan denseblock4 untuk memverifikasi apakah model memanfaatkan struktur anatomis yang relevan secara forensik. Gambar 8.20 menyajikan ilustrasi skematik pola aktivasi yang teramati pada prediksi benar dan prediksi salah.



Gambar 8.20 Ilustrasi Skematik Pola Aktivasi *Grad-Cam* pada Prediksi Benar (Kiri) dan Prediksi Salah (Kanan)

Pada prediksi yang benar dengan *confidence* tinggi, model secara konsisten memusatkan perhatian pada *region* mandibula, khususnya korpus mandibula dan area molar. *Region* inilah yang secara klinis memang diketahui menunjukkan perbedaan morfologi antar jenis kelamin, meliputi ketebalan tulang kortikal, tinggi tulang alveolar, dan kontur tepi bawah mandibula. Kesesuaian antara area perhatian model dan pengetahuan anatomis yang mapan merupakan bukti pendukung bahwa model mempelajari sinyal yang bermakna secara biologis.

Sebaliknya, pada prediksi yang salah, distribusi aktivasi cenderung lebih tersebar dan kurang terfokus, atau justru terpusat pada *region* yang kurang relevan secara anatomis, misalnya area sinus maksilaris pada bagian atas citra. Pola ini konsisten dengan interpretasi bahwa kegagalan prediksi terjadi ketika sinyal anatomis yang relevan tidak cukup kuat, sehingga model "terpaksa" bersandar pada *region* lain yang informasinya lebih lemah.

Dua kesimpulan dapat ditarik dari analisis ini. Pertama, dari sudut pandang validasi ilmiah, *Grad-CAM* berfungsi sebagai mekanisme falsifikasi: seandainya peta aktivasi menunjukkan pemusatan pada tepi citra, penanda perangkat, atau teks anotasi, maka nilai AUC 0,9639 harus dicurigai sebagai hasil pembelajaran korelasi semu. Karena yang teramati adalah pemusatan pada *region* mandibula, hasil tersebut memperoleh dukungan tambahan.

Kedua, dari sudut pandang pertanggungjawaban medis-legal, kemampuan menunjukkan dasar visual keputusan mengubah keluaran sistem dari sekadar angka menjadi argumen yang dapat diperiksa. Ahli forensik dapat menilai sendiri apakah *region* yang disorot masuk akal untuk kasus yang sedang ditangani, dan dapat menolak keluaran sistem apabila tidak.

Meskipun demikian, terdapat batasan yang perlu disadari. *Grad-CAM* menghasilkan peta lokalisasi yang kasar karena resolusi spasial peta aktivasi pada lapisan terakhir jauh lebih rendah daripada resolusi citra asli; peta 7 × 7 yang diperbesar ke 224 × 224 tidak dapat menunjukkan struktur halus. Selain itu, *Grad-CAM* menunjukkan *di mana* model memusatkan perhatian, bukan *mengapa region* tersebut mengarah pada kelas tertentu. Kesesuaian dengan pengetahuan anatomis merupakan bukti pendukung, bukan bukti kausal. Validasi yang lebih kuat menuntut perbandingan sistematis antara peta aktivasi dan penilaian ahli forensik odontologi pada sejumlah kasus yang memadai.

➢ Keterbatasan dan Arah Pengembangan

Pemahaman kritis terhadap keterbatasan sistem merupakan bagian integral dari pendekatan ilmiah yang bertanggung jawab. Bagian ini menguraikan keterbatasan yang teridentifikasi beserta arah perbaikan yang dapat ditempuh.

Keterbatasan Data

Ketidakseimbangan kelas. Rasio 65: 35 berpotensi memperkenalkan bias sistematis. Meskipun stratifikasi membantu memitigasi dampaknya selama pelatihan, pada penerapan nyata distribusi kelas yang berbeda dapat menggeser performa secara berarti. Perbaikan yang dapat ditempuh mencakup penggunaan *class-weighted loss* dengan bobot berbanding terbalik terhadap frekuensi kelas, *oversampling* terarah pada kelas minoritas, atau *focal loss* yang secara otomatis memberi bobot lebih pada sampel yang sulit.

Sumber data tunggal. Seluruh citra berasal dari satu fasilitas kesehatan, sehingga kemampuan generalisasi terhadap citra dari perangkat akuisisi berbeda belum terverifikasi. Variasi protokol akuisisi, kalibrasi perangkat, dan karakteristik demografis populasi dapat memengaruhi performa secara substansial. Fenomena ini dikenal sebagai *domain shift* dan merupakan salah satu penyebab paling umum kegagalan model citra medis saat dipindahkan dari lingkungan pengembangan ke lingkungan operasional. Validasi eksternal pada dataset multi-pusat merupakan prasyarat sebelum penerapan nyata.

Cakupan populasi. Dataset berasal dari populasi Indonesia pada satu wilayah. Karena *dimorfisme* seksual *dentomaksilofasial* diketahui bervariasi antar populasi dan etnis, model belum tentu berlaku pada populasi lain tanpa penyesuaian.

Keterbatasan Metodologis

Klasifikasi biner tanpa opsi abstain. Model dipaksa memilih salah satu dari dua kelas untuk setiap masukan, termasuk pada kasus yang secara biologis ambigu. Penerapan estimasi ketidakpastian, misalnya melalui *Monte Carlo Dropout* atau pendekatan Bayesian, akan memungkinkan sistem menyatakan "tidak dapat menentukan" dan meneruskan kasus tersebut kepada ahli.

Tanpa mekanisme atensi eksplisit. Fokus model pada *region* tertentu muncul sebagai hasil sampingan pelatihan, bukan sebagai sesuatu yang dirancang. Penambahan modul atensi seperti *Convolutional Block Attention Module* (CBAM) atau mekanisme *self-attention* dapat membantu model secara eksplisit memberi bobot lebih pada *region* yang paling diskriminatif.

Tugas tunggal. Model hanya memprediksi jenis kelamin, padahal metadata usia tersedia. Pendekatan *multitask learning* yang memprediksi jenis kelamin dan usia secara bersamaan dapat memanfaatkan korelasi antar tugas untuk memperkaya representasi fitur, sekaligus berfungsi sebagai regularisasi tambahan. Fungsi *loss* gabungannya berbentuk:

$$\mathcal{L}total = \lambda 1\, \mathcal{L}gender + \lambda 2\, \mathcal{L}age \quad (8.38)$$

Interpretasi *Grad-CAM* belum divalidasi ahli. Kesesuaian antara peta aktivasi dan pengetahuan anatomis dinilai secara kualitatif oleh peneliti, belum melalui penilaian terstruktur oleh ahli forensik odontologi. Validasi semacam ini merupakan langkah lanjutan yang penting.

Tabel 8.10 Ringkasan Keterbatasan dan Arah Pengembangan

Keterbatasan	**Dampak**	**Solusi yang Diusulkan**
Ketidakseimbangan kelas 65 : 35	Bias terhadap kelas mayoritas; F1 kelas minoritas lebih rendah	*Class-weighted loss*, *focal loss*, *oversampling* terarah
Sumber data tunggal	Generalisasi lintas perangkat belum teruji	Dataset multi-pusat, domain *adaptation*, validasi eksternal
Klasifikasi biner tanpa abstain	Kasus ambigu tetap dipaksa mendapat label	*Uncertainty estimation* (MC *Dropout*, Bayesian)
Tanpa mekanisme atensi	Fokus *region* tidak terkendali secara eksplisit	CBAM, *self-attention*, atau supervisi atensi
Tugas tunggal	Informasi usia yang tersedia tidak dimanfaatkan	Multitask learning: jenis kelamin + estimasi usia
Grad-CAM belum divalidasi ahli	Klaim interpretabilitas masih kualitatif	Studi terstruktur bersama ahli forensik odontologi



Gambar 8.21 *Roadmap* Pengembangan Sistem dari Baseline Menuju Kesiapan Penerapan Klinis

Rancangan Studi Ablasi yang Disarankan

Hasil yang dilaporkan pada bab ini berasal dari satu konfigurasi tunggal. Konsekuensinya, kontribusi masing-masing komponen terhadap performa akhir belum dapat dipisahkan: tidak diketahui berapa besar sumbangan *label smoothing*, augmentasi, maupun *dropout* secara terpisah. Studi ablasi menjawab pertanyaan tersebut dengan menonaktifkan satu komponen pada satu waktu sambil mempertahankan seluruh komponen lain, lalu mengukur selisih performanya.

Rancangan yang disarankan disajikan pada Tabel 8.11. Setiap konfigurasi dijalankan dengan skema validasi, pembagian lipatan, dan *seed* yang identik, sehingga selisih yang teramati dapat diatribusikan pada komponen yang diubah.

Tabel 8.11 Rancangan Studi Ablasi yang Disarankan

No.	**Konfigurasi**	**Pertanyaan yang Dijawab**
A0	Konfigurasi penuh (*baseline* bab ini)	Titik acuan pembanding
A1	Tanpa bobot pra-latih (pelatihan dari awal)	Berapa besar sumbangan *transfer learning*?
A2	Tanpa augmentasi acak	Apakah augmentasi benar-benar meningkatkan generalisasi?
A3	Tanpa label *smoothing* ($\varepsilon = 0$)	Bagaimana pengaruhnya terhadap kalibrasi dan AUC?

A4	*Dropout* 0 dan 0,2 sebagai pembanding	Apakah 0,4 merupakan pilihan yang tepat?
A5	*Loss* berbobot kelas	Apakah selisih F1 antar kelas dapat dipersempit?
A6	*Backbone* alternatif dengan hiperparameter identik	Apakah *DenseNet169* memang pilihan terbaik?

Khusus untuk konfigurasi A5, bobot kelas ditetapkan berbanding terbalik terhadap frekuensi kemunculan:

$$wc = N / (C \cdot nc) \quad (8.39)$$

dengan N jumlah seluruh sampel, C jumlah kelas, dan n(c) jumlah sampel pada kelas c. Untuk dataset ini diperoleh bobot kelas Female sebesar 5.203 / (2 × 3.383) ≈ 0,769 dan bobot kelas Male sebesar 5.203 / (2 × 1.820) ≈ 1,429, yaitu kelas minoritas memperoleh bobot hampir dua kali lipat kelas mayoritas. Alternatif lain adalah *focal loss*, yang secara otomatis menurunkan bobot sampel yang sudah mudah diklasifikasikan:

$$\mathcal{L}focal = -\Sigma i\ qi\ (1 - pi)\gamma\ log\ pi \quad (8.40)$$

Parameter γ mengendalikan seberapa agresif penurunan bobot tersebut; nilai γ = 0 mengembalikan fungsi ke bentuk *cross entropy* baku. Pendekatan ini sering lebih efektif daripada pembobotan kelas sederhana ketika kesulitan klasifikasi tidak terdistribusi merata di dalam satu kelas, sebagaimana terjadi pada kasus ini di mana kesalahan terkonsentrasi pada pasien usia muda.

Perlu ditekankan bahwa Tabel 8.11 merupakan rancangan, bukan hasil. Menyajikan rancangan eksperimen yang belum dijalankan sebagai bagian dari laporan merupakan praktik yang sehat: ia menandai secara jujur batas antara apa yang telah dibuktikan dan apa yang masih berupa dugaan, sekaligus memberikan peta kerja yang konkret bagi Metodologi lanjutan.

Perbandingan dengan Metodologi Terdahulu

Untuk menempatkan hasil dalam konteks, Tabel 8.12 membandingkan Metodologi ini dengan sejumlah studi terdahulu pada tugas yang serupa.

Tabel 8.12 Perbandingan dengan Metodologi terdahulu

Metodologi	**Jumlah Data**	**Arsitektur**	**Hasil**
Ciconelle dkk. (2023)	207.946	CNN standar, *ResNet*	Akurasi 95,02%
Franco dkk. (2022) — from scratch	Terbatas	Beragam CNN	Akurasi 71%
Franco dkk. (2022) — *transfer learning*	Terbatas	DenseNet121 terbaik	Akurasi 82%
Bab ini	5.203	*DenseNet169* + TL	AUC 0,9639; akurasi 90,62%

Perbandingan ini perlu dibaca dengan hati-hati karena dataset, populasi, dan protokol evaluasi berbeda-beda, sehingga angka tidak dapat dibandingkan secara langsung. Meskipun demikian, dua pola dapat *dice*rmati. Pertama, hasil pada bab ini secara konsisten melampaui Franco dkk. (2022) yang menggunakan jumlah data serupa, sekaligus menegaskan bahwa *transfer learning* jauh mengungguli pelatihan dari awal, dengan selisih 11 poin persentase. Kedua, Ciconelle dkk. (2023) yang mencapai akurasi lebih tinggi menggunakan data sekitar 40 kali lipat lebih banyak, yang mengindikasikan bahwa penambahan data masih merupakan jalur peningkatan yang paling menjanjikan pada tugas ini.

➢ Aspek Etika, Privasi, dan Kelayakan Penerapan

Sistem yang dibahas pada bab ini bekerja dengan data medis pasien yang bersifat sensitif dan menghasilkan keluaran yang berpotensi digunakan dalam konteks hukum. Karena itu, pembahasan teknis tidak lengkap tanpa peninjauan aspek etis dan legal.

Perizinan dan anonimisasi. Penggunaan data radiografi pasien mensyaratkan perizinan dari institusi penyedia data. Anonimisasi harus dilakukan sebelum data digunakan, mencakup penghapusan identitas pasien dari metadata maupun dari citra itu sendiri, karena sebagian perangkat radiografi menuliskan nama pasien dan tanggal akuisisi langsung pada citra. Penggunaan identitas berupa nomor urut tanpa kaitan ke rekam medis, sebagaimana diterapkan pada dataset ini, merupakan praktik yang tepat.

Positioning sistem. Sistem harus diposisikan sebagai alat bantu penyaringan awal (*triage*), bukan sebagai penentu keputusan akhir. Pada konteks forensik, keluaran sistem sebaiknya diperlakukan sebagai satu masukan di antara berbagai jenis bukti, bukan sebagai kesimpulan yang berdiri sendiri. Nilai AUC 0,9639 memang menjanjikan, namun tingkat kesalahan sekitar 9% pada ambang baku berarti kira-kira satu dari sebelas kasus akan salah diklasifikasikan, angka yang tidak dapat diterima sebagai dasar tunggal pengambilan keputusan hukum.

Transparansi keterbatasan. Setiap penerapan sistem harus disertai pernyataan eksplisit mengenai populasi tempat model dilatih, perangkat akuisisi yang digunakan, serta rentang usia yang terwakili. Menggunakan model pada populasi atau perangkat di luar cakupan tersebut tanpa validasi ulang merupakan penyalahgunaan yang berpotensi menghasilkan kesimpulan keliru.

Kategori jenis kelamin. Model dilatih pada label biner administratif. Perlu disadari bahwa label tersebut merepresentasikan jenis kelamin biologis sebagaimana tercatat dalam rekam medis dan tidak mencakup keragaman identitas gender maupun kondisi interseks. Klaim sistem harus dibatasi secara tepat pada apa yang sebenarnya diprediksi, yaitu pola morfologi *dentomaksilofasial* yang berkorelasi dengan jenis kelamin biologis tercatat.

Potensi pemanfaatan. Dengan batasan-batasan tersebut, sistem tetap memiliki nilai praktis pada dua ranah. Pada ranah forensik, sistem dapat mempercepat penyaringan awal ketika jumlah korban besar dan waktu terbatas, sehingga sumber daya ahli dapat difokuskan pada kasus yang sulit. Pada ranah klinis, sistem dapat berfungsi sebagai mekanisme verifikasi silang terhadap kesesuaian citra dengan rekam medis pasien, khususnya pada sistem informasi yang belum terintegrasi penuh.

➢ Ringkasan

Bab ini memaparkan pengembangan sistem klasifikasi jenis kelamin berbasis citra radiografi panoramik gigi menggunakan arsitektur *DenseNet169* dengan pendekatan *transfer learning*, secara utuh dari eksplorasi data hingga refleksi kritis. Butir-butir pokoknya adalah sebagai berikut:

a. Gigi merupakan sumber bukti biologis yang tahan terhadap suhu ekstrem dan dekomposisi, sehingga citra panoramik gigi menjadi modalitas yang bernilai untuk identifikasi forensik ketika metode lain tidak dapat diterapkan.
b. *Dimorfisme* seksual *dentomaksilofasial* bersifat statistik, bukan deterministik, sehingga terdapat batas atas teoretis bagi akurasi model mana pun pada tugas ini.
c. *Dense connectivity* pada DenseNet mengganti penjumlahan fitur dengan konkatenasi, sehingga informasi dari lapisan awal tetap tersedia secara eksplisit dan memungkinkan penggunaan ulang fitur yang efisien.
d. *Transfer learning* dari *ImageNet* terbukti efektif meskipun domain sumber dan sasaran berbeda secara semantik, karena fitur tingkat rendah berupa tepi, tekstur, dan gradien tetap relevan bagi citra radiografi.
e. Eksplorasi data bukan formalitas: verifikasi bahwa distribusi usia dan karakteristik akuisisi kedua kelas berimpit menutup jalur pembelajaran pintasan dan memperkuat validitas hasil.
f. *Stratified K-Fold Cross Validation* menghasilkan estimasi performa yang lebih andal, dan simpangan baku antar lipatan sering kali lebih informatif daripada nilai rata-rata.
g. Pada dataset tidak seimbang, AUC-ROC lebih tepat dijadikan metrik utama daripada akurasi, dan performa per kelas harus dilaporkan bersama metrik agregat.
h. Model mencapai AUC 0,9639 ± 0,0054, akurasi 0,9062 ± 0,0101, dan *F1-score* 0,9064 ± 0,0099, dengan kestabilan tinggi antar lipatan.
i. Perbedaan *confidence* antara prediksi benar (0,9127) dan salah (0,6879) menunjukkan kalibrasi yang memadai dan membuka peluang penerapan strategi penolakan selektif.
j. *Grad-CAM* memperlihatkan pemusatan perhatian model pada *region* mandibula dan molar pada prediksi benar, yang konsisten dengan pengetahuan anatomis forensik, sehingga berfungsi sebagai mekanisme falsifikasi sekaligus sarana pertanggung-jawaban.

k. Sistem harus diposisikan sebagai alat bantu, bukan pengganti penilaian ahli, dan penerapannya menuntut validasi eksternal, transparansi keterbatasan, serta kepatuhan pada kaidah etika Metodologi medis.

➢ Latihan

Soal Konseptual

a. Jelaskan perbedaan mendasar antara koneksi residual pada *ResNet* (Persamaan 8.9) dan *dense connectivity* pada DenseNet (Persamaan 8.10). Mengapa perbedaan operasi penjumlahan dan konkatenasi berdampak besar pada aliran informasi?
b. Sebuah model diuji pada dataset dengan komposisi kelas 90 : 10 dan memperoleh akurasi 90%. Mengapa angka ini belum tentu menunjukkan model yang baik? Metrik apa yang sebaiknya dilaporkan dan mengapa?
c. Mengapa augmentasi berupa pencerminan vertikal tidak digunakan pada citra panoramik gigi, padahal pencerminan horizontal digunakan? Jelaskan dari sudut pandang validitas anatomis.
d. Jelaskan mengapa fungsi eval_one_*epoch* pada Kode 8.7 wajib memanggil model.eval(), dan apa yang terjadi bila pemanggilan tersebut terlewat.
e. Mengapa ReLU pada Persamaan (8.28) bersifat esensial bagi *Grad-CAM*? Apa yang akan terlihat pada heatmap bila ReLU dihilangkan?

Soal Perhitungan

a. Sebuah lapisan konvolusi menerima masukan berukuran 224 × 224 dengan 64 kanal, menggunakan 128 kernel berukuran 3 × 3, padding 1, dan stride 1. Hitunglah dimensi spasial keluaran menggunakan Persamaan (8.6) dan jumlah parameter lapisan tersebut menggunakan Persamaan (8.8).
b. Sebuah dense block memiliki 12 lapisan. Hitunglah jumlah koneksi langsung di dalamnya menggunakan Persamaan (8.11), lalu bandingkan dengan jumlah koneksi pada CNN konvensional berkedalaman sama.

c. Diketahui matriks kebingungan Fold 5 dengan kelas Male sebagai kelas positif: TP = 321, TN = 634, FP = 42, dan FN = 43. Hitunglah akurasi, presisi, *recall*, dan *F1-score* menggunakan Persamaan (8.30) sampai (8.33), lalu bandingkan hasilnya dengan Tabel 8.9.
d. Untuk klasifikasi biner dengan label *smoothing* $\varepsilon = 0{,}1$, hitunglah nilai target yang dilunakkan menggunakan Persamaan (8.20). Kemudian hitung nilai *loss* pada Persamaan (8.21) bila model menghasilkan probabilitas $p = (0{,}2;\ 0{,}8)$ dan kelas benar adalah kelas kedua.
e. Diberikan nilai AUC pada lima lipatan sebesar 0,9533; 0,9667; 0,9648; 0,9667; dan 0,9682. Hitunglah rata-rata dan simpangan baku menggunakan Persamaan (8.36), lalu tafsirkan makna nilai simpangan baku yang diperoleh.

Soal Implementasi

a. Modifikasi Kode 8.5 agar menggunakan *class-weighted cross entropy* dengan bobot berbanding terbalik terhadap frekuensi kelas. Hitung terlebih dahulu bobot yang sesuai untuk komposisi 3.383 berbanding 1.820.
b. Implementasikan strategi penolakan selektif: tulis fungsi yang menerima daftar probabilitas dan label, lalu mengembalikan akurasi serta cakupan pada berbagai nilai ambang *confidence* $\tau \in \{0{,}5;\ 0{,}6;\ 0{,}7;\ 0{,}8;\ 0{,}9\}$.
c. Ubah Kode 8.4 menjadi arsitektur multitask yang menghasilkan dua keluaran: logit klasifikasi jenis kelamin dan nilai skalar estimasi usia. Rancang pula fungsi *loss* gabungannya mengikuti Persamaan (8.38).
d. Implementasikan Monte Carlo *Dropout*: aktifkan *dropout* saat inferensi, lakukan 30 kali lintasan maju untuk setiap citra, lalu gunakan simpangan baku prediksi sebagai ukuran ketidakpastian.
e. Bandingkan performa *DenseNet169* dengan setidaknya satu arsitektur alternatif, misalnya *EfficientNet*-B0 atau *ResNet*50, menggunakan skema validasi dan konfigurasi hiperparameter yang identik. Laporkan hasilnya dalam format rata-rata ± simpangan baku.

➢ Kunci Jawaban Soal Perhitungan

Bagian ini menyajikan penyelesaian ringkas untuk kelima soal perhitungan pada Subbab 8.10 agar pembaca dapat memeriksa pemahamannya secara mandiri.

a. **Soal 1 — Dimensi keluaran dan jumlah parameter**
Dengan W = 224, K = 3, P = 1, dan S = 1, Persamaan (8.6) memberikan:
O = ⌊ (224 − 3 + 2·1) / 1 ⌋ + 1 = 223 + 1 = 224 (—)
Dimensi keluaran adalah 224 × 224 × 128, yaitu dimensi spasial tidak berubah karena *padding* 1 tepat mengimbangi kernel 3 × 3 pada *stride* 1. Jumlah parameter menurut Persamaan (8.8):
N_{param} = (3 × 3 × 64 + 1) × 128 = 577 × 128 = 73.856 (—)
Perhatikan bahwa jumlah parameter sepenuhnya tidak bergantung pada ukuran 224 × 224. Inilah sifat yang membuat CNN dapat diterapkan pada citra beresolusi tinggi tanpa ledakan jumlah parameter.

b. **Soal 2 — Jumlah koneksi dalam dense block**
Dengan L = 12, Persamaan (8.11) memberikan:
$N_{koneksi}$ = 12 × 13 / 2 = 78 (—)
CNN konvensional dengan kedalaman yang sama hanya memiliki 12 koneksi langsung, sehingga dense block memuat 6,5 kali lebih banyak jalur informasi. Perlu dicatat bahwa penambahan jalur ini tidak berarti penambahan parameter yang sebanding, karena konkatenasi sendiri tidak memiliki parameter; yang bertambah adalah jumlah kanal masukan bagi setiap lapisan berikutnya.

c. **Soal 3 — Metrik dari matriks kebingungan**
Dengan kelas Male sebagai kelas positif dan TP = 321, TN = 634, FP = 42, FN = 43:
Accuracy = (321 + 634) / 1.040 = 955 / 1.040 = 0,9183 (—)
Precision = 321 / (321 + 42) = 321 / 363 = 0,8843 (—)
Recall = 321 / (321 + 43) = 321 / 364 = 0,8819 (—)
F1 = 2 × (0,8843 × 0,8819) / (0,8843 + 0,8819) = 0,8831 (—)
Keempat nilai tersebut tepat sama dengan baris Male pada Tabel 8.9, yang mengonfirmasi konsistensi internal pelaporan hasil.

d. **Soal 4 — Label *smoothing***
Untuk C = 2 dan ε = 0,1, Persamaan (8.20) memberikan target yang dilunakkan sebesar 0,05 untuk kelas yang salah dan 0,95 untuk kelas yang benar. Dengan p = (0,2; 0,8) dan kelas benar adalah kelas kedua, Persamaan (8.21) menghasilkan:
$\mathscr{L} = -[\ 0{,}05 \cdot \ln 0{,}2 + 0{,}95 \cdot \ln 0{,}8\ ] = -[\ -0{,}0805 - 0{,}2120\ ] = 0{,}2925$ (—)
Sebagai pembanding, tanpa *label smoothing* nilai *loss* adalah $-\ln 0{,}8 = 0{,}2231$. *Loss* yang lebih besar menunjukkan bahwa pelunakan target memberikan hukuman tambahan atas keyakinan yang berlebihan, sehingga model didorong untuk tidak memaksakan probabilitas mendekati satu.

e. **Soal 5 — Rata-rata dan simpangan baku**
Dengan lima nilai AUC 0,9533; 0,9667; 0,9648; 0,9667; dan 0,9682, Persamaan (8.36) memberikan:
$\mu = 4{,}8197 / 5 = 0{,}9639$ (—)
$\sigma = \sqrt{[\ (1/5)\ \Sigma\ (m_k - \mu)^2\ ]} = 0{,}0054$ (—)
Tafsirannya: rentang performa antar lipatan hanya sekitar setengah poin persentase di sekitar nilai tengah. Ini menandakan bahwa model tidak sensitif terhadap pembagian data tertentu, sehingga nilai 0,9639 dapat diperlakukan sebagai estimasi yang stabil, bukan sebagai hasil satu pembagian yang kebetulan menguntungkan. Perhatikan pula bahwa nilai lipatan pertama, 0,9533, terletak lebih dari dua simpangan baku di bawah rata-rata, sehingga layak diperiksa apakah lipatan tersebut memuat sampel yang secara sistematis lebih sulit.

➢ Daftar Pustaka

Ata, I. (2022). Human Gender Prediction Based on Deep *Transfer Learning* from Panoramic Dental Radiograph Images.

Carter, E. (2025). Forensic Odontology: Identifying Disaster Victims Through Dental Records, 16, 7–9.

Ciconelle, A. C. M., da Silva, R. L. B., Kim, J. H., Rocha, B. A., Dos Santos, D. G., Vianna, L. G. R., Gomes Ferreira, L. G., Pereira Dos Santos, V. H., Costa, J. O., & Vicente, R. (2023). *Deep*

learning for sex determination: Analyzing over 200,000 panoramic radiographs. Journal of Forensic Sciences, 68(6), 2057–2064.

Faragalli, A., Ferrante, L., Angelakopoulos, N., & Cameriere, R. (2025). Do machine learning methods solve the main pitfall of linear regression in dental age estimation? Forensic Science International, 367, 112353.

Franco, A., Porto, L., Heng, D., Murray, J., Lygate, A., Franco, R., Bueno, J., Sobania, M., Costa, M. M., Paranhos, L. R., Manica, S., & Abade, A. (2022). Diagnostic performance of convolutional neural networks for dental sexual dimorphism. Scientific Reports, 12(1), 1–12.

Franklin, D., O'Higgins, P., Oxnard, C. E., & Dadour, I. (2008). Discriminant function sexing of the mandible of Indigenous South Africans. Forensic Science International, 179(1), 84.e1–84.e5.

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep learning*. Nature, 521, 436–444.

Loshchilov, I., & Hutter, F. (2019). Decoupled Weight Decay Regularization. International Conference on Learning Representations.

Maysanjaya, I. D. (2020). Klasifikasi *Pneumonia* pada Citra X-rays Paru-paru dengan Convolutional Neural Network, 9(2), 190–195.

Pan, S. J., & Yang, Q. (2009). A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345–1359.

Prajapati, G., Sarode, S. C., Sarode, G. S., Shelke, P., Awan, K. H., & Patil, S. (2018). Role of forensic odontology in the identification of victims of major mass disasters across the world: A systematic review. PLoS ONE, 13(6), 1–12.

Pretty, I. A., & Sweet, D. (2001). A look at forensic dentistry — Part 1: The role of teeth in the determination of human identity. British Dental Journal, 190(7), 359–366.

Qiu, G. (2020). Metastatic Cancer Image Classification Based on *Deep Learning* Method. arXiv preprint.

Rangel, G., Cuevas-Tello, J. C., Silva-Trujillo, A. G., Nunez-Varela, J., & Puente, C. (2024). A Survey on Convolutional Neural Networks and Their Performance Limitations in Image Recognition Tasks. Journal of Sensors, 2024.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. Proceedings of the IEEE International Conference on Computer Vision.

Shivananda, S., Doddawad, V. G., Anand, A., Patil, K., Cs, V., & Suresh, J. (2025). Effects on heat-exposed teeth in relation to forensic identification — a pilot study.

Standish, S. M., & Stimson, P. G. (1977). The Scope of Forensic Dentistry. Dental Clinics of North America, 21(1), 3–5.

Su, S. P., Kim, Y. J., Kim, J. K. J., & Lee, K. H. S. (2024). Automatic and robust estimation of sex and chronological age from panoramic radiographs using a multi-task *deep learning* network: a study on a South Korean population. International Journal of Legal Medicine, 1741–1757.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

BAB 9
SISTEM ANALISIS PROPORSI MORFOLOGI SEL DARAH MERAH PADA APUSAN DARAH TEPI MENGGUNAKAN *MOBILENETV2* UNTUK PERINGATAN DINI TALASEMIA

Bab ini membahas secara utuh perancangan, pembangunan, pengujian, dan penerapan sebuah sistem peringatan dini talasemia berbasis analisis proporsi morfologi sel darah merah pada citra apusan darah tepi. Sistem yang dibahas diberi nama *Thalassemia Morphology Screener* (TMS) dan dikembangkan sebagai proof-of-concept, yaitu pembuktian kelayakan konsep, bukan sebagai alat diagnosis yang telah tervalidasi secara klinis. Pembahasan disusun mengikuti alur kerja Metodologi yang sesungguhnya, mulai dari pemahaman biologis mengenai talasemia dan morfologi *eritrosit*, penyiapan data citra mikroskopis, pembangunan model klasifikasi berbasis *transfer learning* dengan arsitektur *MobileNetV2*, verifikasi keputusan model menggunakan *Grad*-CAM, penerapan model pada citra apusan utuh melalui *pipeline* segmentasi, hingga pengemasan seluruh alur menjadi aplikasi antarmuka pengguna berbasis *Streamlit*.

Selain membahas sistem TMS, bab ini juga menyajikan sebuah contoh studi kasus pelengkap pada ranah sains data tabular, yaitu pengenalan pola status rawat inap pasien berdasarkan profil hematologi menggunakan algoritma *Random Forest* dan interpretasi model dengan SHAP. Studi kasus pelengkap tersebut dimasukkan karena memperlihatkan sisi lain dari pemanfaatan data hematologi: apabila TMS bekerja pada data citra, maka studi kasus kedua bekerja pada data numerik hasil pemeriksaan laboratorium. Keduanya bersama-sama membentuk gambaran yang lebih utuh mengenai bagaimana kecerdasan buatan dapat mendukung proses skrining dan pengambilan keputusan klinis awal.

Setelah membaca bab ini, pembaca diharapkan mampu:

a. Menjelaskan patofisiologi dasar talasemia serta kaitannya dengan perubahan morfologi sel darah merah yang dapat diamati pada sediaan apusan darah tepi.

b. Menguraikan sembilan kategori morfologi sel darah merah yang digunakan dalam sistem, beserta ciri visual pembeda masing-masing kategori.
c. Menjelaskan konsep *deep learning*, convolutional neural network, arsitektur *MobileNetV2*, serta prinsip *transfer learning* dua tahap.
d. Merancang dan melaksanakan *pipeline* pengolahan citra untuk segmentasi sel darah merah dari citra apusan darah tepi, meliputi ekstraksi kanal hijau, *thresholding Otsu*, operasi morfologi, dan segmentasi *watershed*.
e. Menerapkan strategi penanganan ketidakseimbangan kelas serta menjelaskan alasan teknis di balik pemilihan setiap parameter pelatihan.
f. Mengevaluasi kinerja model klasifikasi citra medis menggunakan *accuracy, precision, recall, F1-score*, dan *confusion matrix*, serta memverifikasi penalaran model menggunakan *Grad*-CAM.
g. Menghitung proporsi jenis sel pada sebuah apusan dan menerjemahkannya menjadi indikator risiko talasemia berbasis aturan ambang batas.
h. Mengemas model dan *pipeline* menjadi aplikasi berbasis *Streamlit* yang dapat dioperasikan oleh pengguna non-teknis.
i. Menyampaikan keterbatasan sebuah sistem berbasis kecerdasan buatan secara jujur dan bertanggung jawab, khususnya pada ranah kesehatan.

➢ Pendahuluan

Talasemia merupakan salah satu kelainan darah genetik yang paling banyak ditemukan di dunia. Kelainan ini disebabkan oleh gangguan pada sintesis rantai globin sehingga tubuh memproduksi hemoglobin secara tidak normal. Hemoglobin adalah protein di dalam sel darah merah yang bertugas mengangkut oksigen dari paru-paru ke seluruh jaringan tubuh. Ketika produksi hemoglobin terganggu, sel darah merah menjadi rapuh, mudah pecah, dan berumur lebih pendek daripada semestinya. Akibatnya penderita mengalami anemia kronis, kelelahan berkepanjangan, pembesaran limpa, gangguan pertumbuhan pada anak, serta berbagai komplikasi serius lain apabila tidak ditangani sejak dini.

Indonesia termasuk negara dengan beban talasemia yang tinggi. Berdasarkan data Kementerian Kesehatan Republik Indonesia, prevalensi pembawa sifat (*carrier*) talasemia di Indonesia diperkirakan berkisar antara 3 sampai 10 persen dari total populasi. Angka tersebut berarti terdapat jutaan penduduk yang membawa gen talasemia tanpa menunjukkan gejala berarti, namun berpotensi menurunkan kelainan tersebut kepada keturunannya. Apabila dua orang pembawa sifat menikah, terdapat peluang sebesar 25 persen pada setiap kehamilan untuk melahirkan anak dengan talasemia mayor, yaitu bentuk paling berat yang memerlukan transfusi darah rutin seumur hidup. Karena itulah deteksi dini menjadi kunci utama dalam menekan angka kelahiran penderita talasemia mayor.

Persoalannya, diagnosis pasti talasemia umumnya memerlukan pemeriksaan laboratorium lanjutan seperti *High Performance Liquid Chromatography* (HPLC) dan analisis DNA. Kedua pemeriksaan tersebut memerlukan biaya tinggi, peralatan khusus, serta tenaga ahli yang tidak selalu tersedia, terutama di fasilitas kesehatan dengan sumber daya terbatas seperti puskesmas dan rumah sakit daerah. Konsekuensinya, banyak kasus talasemia tidak terdeteksi sejak dini dan baru diketahui setelah pasien menunjukkan gejala klinis yang berat.

Di sisi lain, terdapat satu pemeriksaan awal yang relatif murah dan hampir selalu tersedia di laboratorium klinik mana pun, yaitu pengamatan apusan darah tepi (*peripheral blood smear*). Pada penderita talasemia, sel darah merah menunjukkan kelainan morfologi yang cukup khas, seperti munculnya sel target (*target cell*), sel tetesan air mata (*teardrop cell*), sel hipokromik, dan sel pensil (*pencil cell*) dalam proporsi tertentu. Sayangnya pembacaan morfologi secara manual bersifat subjektif, memerlukan tenaga analis yang terlatih, dan memakan waktu cukup lama untuk setiap sediaan. Konsistensi hasil antar-pembaca juga sulit dijaga, apalagi ketika beban kerja tinggi dan pembaca mengalami kelelahan. Karena itu pembacaan manual sulit dijadikan alat skrining massal yang konsisten.

Perkembangan kecerdasan buatan, khususnya pembelajaran mendalam (*deep learning*) pada bidang visi komputer, membuka peluang untuk membantu proses pembacaan morfologi tersebut secara otomatis dan objektif. Berbagai Metodologi terdahulu telah berhasil mengklasifikasikan jenis-jenis sel darah merah dengan

akurasi yang tinggi. Meskipun demikian, sebagian besar Metodologi tersebut hanya bekerja pada citra sel tunggal yang sudah dipotong dan disegmentasi terlebih dahulu, belum pada citra apusan darah tepi yang utuh. Selain itu hasil Metodologi tersebut umumnya berhenti pada tahap laporan akurasi, belum tersedia dalam bentuk alat yang sederhana, terbuka, dan siap dicoba di lapangan oleh tenaga kesehatan.

Berangkat dari kesenjangan tersebut, bab ini memaparkan pengembangan *Thalassemia Morphology Screener* (TMS), yaitu sebuah sistem skrining dini berbasis morfologi sel darah merah yang menerima masukan berupa citra apusan darah tepi utuh, melakukan segmentasi sel secara otomatis, mengklasifikasikan setiap sel ke dalam sembilan kategori morfologi, menghitung proporsi sel abnormal, dan akhirnya memberikan indikator risiko pada sebuah antarmuka pengguna yang sederhana. Sistem ini diposisikan sebagai alat bantu skrining tahap awal, bukan sebagai pengganti pemeriksaan baku emas.

Berdasarkan latar belakang di atas, permasalahan yang hendak dijawab pada bab ini dirumuskan sebagai berikut:

a. Bagaimana merancang *pipeline* pengolahan citra yang mampu memisahkan sel darah merah individual dari citra apusan darah tepi utuh yang selnya banyak saling bersentuhan dan bertumpuk?
b. Bagaimana membangun model klasifikasi sembilan morfologi sel darah merah yang akurat meskipun jumlah data latih terbatas dan distribusi antar-kelasnya sangat timpang?
c. Bagaimana memastikan bahwa keputusan model didasarkan pada ciri morfologi sel yang benar dan bukan pada artefak citra yang kebetulan berkorelasi dengan label?
d. Bagaimana menerjemahkan hasil klasifikasi tingkat sel menjadi indikator risiko talasemia pada tingkat subjek yang mudah dipahami oleh tenaga medis?
e. Bagaimana mengemas seluruh alur tersebut menjadi aplikasi yang dapat dioperasikan tanpa keahlian pemrograman?

Tujuan yang hendak dicapai melalui pengembangan sistem yang dibahas pada bab ini meliputi:

a. Menghasilkan *pipeline* segmentasi sel darah merah dari citra apusan darah tepi utuh yang konsisten dengan metodologi sumber dataset.
b. Menghasilkan model klasifikasi sembilan morfologi sel darah merah berbasis *transfer learning MobileNetV2* dengan akurasi yang memadai dan kinerja per kelas yang dilaporkan secara jujur.
c. Menerapkan *Grad*-CAM sebagai metode *Explainable Artificial Intelligence* untuk memverifikasi penalaran model.
d. Menghasilkan mekanisme perhitungan proporsi sel dan indikator risiko tiga tingkat (hijau, kuning, merah) pada tingkat subjek.
e. Menghasilkan prototipe aplikasi *Streamlit* yang terintegrasi dan dapat langsung dicoba dengan mengunggah satu citra apusan.

Manfaat yang diharapkan dari sistem ini bersifat ganda. Dari sisi praktis, sistem dapat mempercepat proses skrining awal, membantu tenaga laboratorium mengidentifikasi kelainan morfologi sel secara lebih objektif, serta mengurangi beban kerja manual yang rentan terhadap kelelahan dan perbedaan interpretasi antar-pembaca. Sistem berbasis perangkat lunak ringan seperti ini juga berpotensi diintegrasikan sebagai alat bantu skrining jarak jauh (*telemedicine*) untuk daerah dengan akses terbatas terhadap ahli hematologi. Dari sisi akademik, sistem ini menunjukkan bagaimana rangkaian teknik pengolahan citra digital, visi komputer, dan *explainable artificial intelligence* dapat dirangkai menjadi satu alur kerja yang utuh dan dapat dipertanggungjawabkan secara metodologis.

Agar pembahasan tetap terarah, perlu ditegaskan sejak awal batasan-batasan berikut. Pertama, sistem ini adalah *proof-of-concept* dan bukan alat diagnosis. Seluruh keluaran sistem berupa indikator risiko harus dibaca sebagai sinyal untuk menindaklanjuti pemeriksaan, bukan sebagai kesimpulan diagnostik. Kedua, jumlah subjek apusan yang digunakan sangat terbatas, yaitu hanya lima subjek yang terdiri atas empat pasien talasemia dan satu individu sehat. Ketiga, tidak tersedia label kebenaran per sel pada citra apusan utuh, sehingga proporsi yang dihitung bersifat estimasi berdasarkan prediksi model, bukan berdasarkan anotasi pakar. Keempat, aturan

ambang batas risiko yang digunakan bersifat demonstratif dan belum dikalibrasi maupun divalidasi bersama pakar hematologi. Kelima, sistem hanya menangani sembilan kategori morfologi yang tersedia pada dataset dan tidak mencakup kelainan sel darah putih maupun trombosit.

➢ Landasan Teori

Talasemia

Talasemia merupakan kelainan darah bawaan yang ditandai dengan produksi hemoglobin abnormal akibat gangguan sintesis rantai globin. Hemoglobin dewasa normal (HbA) tersusun atas dua rantai alfa dan dua rantai beta. Apabila produksi rantai alfa terganggu, kondisi tersebut disebut talasemia alfa; apabila yang terganggu adalah rantai beta, disebut talasemia beta. Ketidakseimbangan jumlah rantai globin menyebabkan rantai yang berlebih mengendap di dalam sel darah merah, merusak membran sel, dan memicu penghancuran sel darah merah lebih awal baik di dalam sumsum tulang maupun di limpa.

Secara klinis, talasemia dibagi menjadi tiga tingkat keparahan. Talasemia minor atau pembawa sifat umumnya tidak menunjukkan gejala berarti, paling banter berupa anemia ringan yang sering disalahartikan sebagai anemia defisiensi besi. Talasemia intermedia menimbulkan anemia sedang yang kadang memerlukan transfusi. Talasemia mayor menimbulkan anemia berat sejak usia dini dan mengharuskan penderita menjalani transfusi darah rutin seumur hidup disertai terapi kelasi besi untuk membuang kelebihan zat besi akibat transfusi. Beban biaya, beban psikologis, dan beban sosial pada keluarga penderita talasemia mayor sangat besar, sehingga pencegahan melalui skrining pembawa sifat jauh lebih efisien dibanding penanganan kuratif.

Pada tingkat sediaan apusan darah tepi, gangguan sintesis hemoglobin ini meninggalkan jejak visual yang khas. Sel darah merah tampak lebih pucat karena kandungan hemoglobinnya berkurang, ukurannya lebih kecil dari normal, dan bentuknya beragam akibat ketidakstabilan membran. Jejak visual inilah yang menjadi objek analisis pada sistem yang dibahas dalam bab ini.

Sel Darah Merah (*Eritrosit*)

Sel darah merah atau *eritrosit* adalah komponen darah dengan jumlah terbanyak. Fungsi utamanya adalah mengangkut oksigen dari paru-paru ke seluruh jaringan tubuh serta membawa karbon dioksida kembali ke paru-paru untuk dikeluarkan. Bentuk *eritrosit* normal menyerupai cakram bikonkaf, yaitu cekung pada kedua sisinya. Bentuk khas ini memberikan dua keuntungan sekaligus: fleksibilitas untuk melewati pembuluh kapiler yang diameternya bahkan lebih kecil daripada diameter sel itu sendiri, serta luas permukaan yang maksimal untuk pertukaran gas.

Pada kondisi normal, *eritrosit* memiliki ukuran yang relatif seragam dengan diameter rata-rata sekitar 7,5 mikrometer. Warnanya merah muda merata dengan area pucat konsentris di bagian tengah yang mencakup kira-kira sepertiga diameter sel. Area pucat tersebut merupakan konsekuensi langsung dari bentuk bikonkaf: bagian tengah sel lebih tipis sehingga cahaya lebih banyak diteruskan. *Eritrosit* matang tidak memiliki inti sel, sehingga seluruh volume internalnya dapat dipakai untuk menampung hemoglobin.

Pada berbagai kondisi patologis, termasuk talasemia, bentuk, ukuran, dan warna *eritrosit* dapat berubah secara signifikan. Perubahan bentuk disebut poikilositosis, perubahan ukuran disebut anisositosis, dan perubahan warna akibat kekurangan hemoglobin disebut hipokromia. Ketiga perubahan tersebut dapat diamati langsung melalui mikroskop pada sediaan apusan darah tepi yang telah diwarnai, dan sampai hari ini pengamatan tersebut masih merupakan metode standar dan fundamental dalam identifikasi kelainan hematologis.

Morfologi Sel Darah Merah pada Talasemia

Sistem yang dibahas pada bab ini mengklasifikasikan sel darah merah ke dalam sembilan kategori morfologi. Delapan di antaranya merupakan bentuk abnormal dengan relevansi klinis yang berbeda-beda, sedangkan satu kategori mewakili sel normal. Uraian ringkas masing-masing kategori disajikan berikut ini.

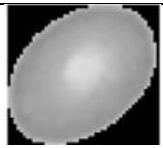
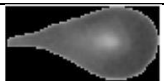
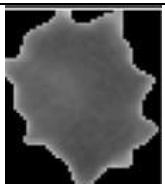
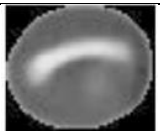
a. *Sel target *(target cell).** Sel yang memiliki zona pucat di bagian tengah dengan bintik hemoglobin di tengah zona tersebut serta cincin hemoglobin di tepinya, sehingga penampakannya menyerupai sasaran tembak. Sel target terbentuk akibat kelebihan luas membran sel dibandingkan volume hemoglobin

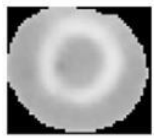
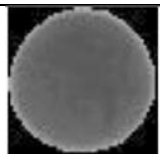
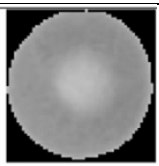
di dalamnya. Sel ini merupakan salah satu penanda paling khas pada talasemia dan hemoglobinopati lain.

b. *Sel hipokromik (hypochromic cell).* Sel yang tampak jauh lebih pucat dari normal karena kekurangan hemoglobin intraseluler, dengan zona pucat pusat yang meluas melebihi sepertiga diameter sel. Hipokromia merupakan penanda utama talasemia sekaligus anemia defisiensi besi.
c. *Sel tetesan air mata *(teardrop cell atau dacrocyte).** Sel berbentuk seperti tetesan air mata atau buah pir dengan salah satu ujung meruncing. Bentuk ini terjadi akibat deformasi sel ketika melewati sumsum tulang yang mengalami fibrosis atau ketika melewati limpa yang membesar.
d. *Sel eliptosit atau ovalosit *(elliptocyte/ovalocyte).** Sel berbentuk oval atau elips, memanjang dari bentuk bikonkaf normal. Umumnya ditemukan pada talasemia minor, anemia defisiensi besi, dan anemia megaloblastik.
e. *Sel pensil *(pencil cell).** Sel yang memanjang seperti pensil atau cerutu dengan ujung meruncing. Sel pensil dianggap sebagai variasi ekstrem dari eliptosit dan sering dijumpai pada talasemia serta anemia defisiensi besi.
f. *Sel stomatosit *(*stomatocyte*).* Sel dengan area pucat berbentuk celah memanjang menyerupai mulut *(*stoma*) di bagian tengah. Terjadi akibat gangguan keseimbangan ion dan air pada membran sel.
g. *Sel akantosit *(acanthocyte).** Sel dengan tonjolan mirip duri yang tidak teratur pada permukaan membrannya. Terbentuk akibat perubahan komposisi lipid membran, dijumpai pada penyakit hati, abetalipoproteinemia, dan beberapa gangguan genetik.
h. *Sferosit *(spherocyte).** Sel berbentuk bola penuh, lebih kecil dan lebih padat dibanding sel normal, tanpa area pucat di bagian tengah. Terjadi akibat hilangnya kelebihan membran sehingga sel kehilangan fleksibilitasnya.
i. *Sel normal *(normal cell).** Sel bikonkaf berdiameter sekitar 7,5 mikrometer dengan warna merah muda merata dan area pucat konsentris di tengah yang mencakup sekitar sepertiga diameter sel.

Ringkasan kesembilan kategori tersebut beserta contoh citranya disajikan pada Tabel 9.1 berikut. Contoh citra diambil langsung dari dataset yang digunakan dalam Metodologi ini sehingga mencerminkan kondisi data yang sesungguhnya dihadapi model, termasuk keterbatasan resolusi dan variasi kontras yang ada.

Tabel 9.1 Sembilan Kategori Morfologi Sel Darah Merah pada Dataset Rbcdataset

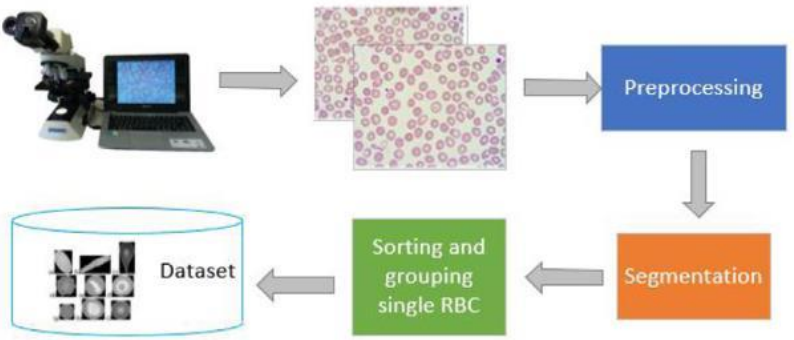
Jenis Sel	**Deskripsi Morfologi**	**Contoh Citra**
Sel eliptosit / ovalosit (*Elliptocyte*)	Sel berbentuk oval atau elips, memanjang dari bentuk bikonkaf normal. Umumnya ditemukan pada talasemia minor, anemia defisiensi besi, dan anemia megaloblastik.	
Sel pensil (*Pencil Cell*)	Sel memanjang seperti pensil atau cerutu dengan ujung meruncing. Sering dijumpai pada talasemia dan anemia defisiensi besi.	
Sel tetesan air mata (*Teardrop Cell*)	Sel berbentuk tetesan air mata dengan salah satu ujung meruncing. Penanda khas talasemia, mielofibrosis, dan anemia hemolitik.	
Sel akantosit (*Acanthocyte*)	Sel dengan proyeksi mirip duri yang tidak teratur di permukaan membran. Dijumpai pada abetalipoproteinemia, penyakit hati, dan gangguan genetik tertentu.	
Sel stomatosit (*Stomatocyte*)	Sel dengan area pucat berbentuk celah menyerupai mulut di bagian tengah. Ditemukan pada hereditary stomatocytosis, penyakit hati, dan alkoholisme.	

Sel target (*Target Cell*)	Sel dengan penampilan menyerupai sasaran tembak: cincin hemoglobin di tepi, area pucat, dan bintik hemoglobin di tengah. Khas pada talasemia.	
Sferosit (*Spherocyte*)	Sel berbentuk bola penuh, lebih kecil dan lebih padat dari sel normal, tanpa area pucat di tengah. Ditemukan pada anemia hemolitik autoimun.	
Sel hipokromik (*Hypochromic Cell*)	Sel dengan area pucat yang meluas melebihi sepertiga diameter sel akibat kekurangan hemoglobin. Penanda utama talasemia dan anemia defisiensi besi.	
Sel normal (*Normal Cell*)	Sel bikonkaf berdiameter sekitar 7,5 mikrometer, warna merah muda merata, area pucat konsentris mencakup sekitar sepertiga diameter sel.	

Citra Mikroskopis Apusan Darah Tepi

Citra mikroskopis sel darah merah diperoleh menggunakan mikroskop cahaya dengan perbesaran tinggi, umumnya seribu kali dengan bantuan minyak imersi, pada sediaan apusan darah tepi yang telah diwarnai dengan pewarnaan Wright-Giemsa atau May-Grünwald-Giemsa. Pewarnaan bertujuan memberikan kontras yang memadai antara sel darah merah, sel darah putih, dan latar belakang plasma sehingga morfologi sel dapat diamati dengan jelas.

Dataset yang digunakan pada Metodologi ini bersumber dari RBCdataset yang dikembangkan oleh Tyas dan rekan-rekan pada tahun 2022. Alur perolehan dataset tersebut penting untuk dipahami karena berimplikasi langsung pada keputusan teknis di tahap penerapan model, sebagaimana akan dibahas pada subbab hasil. Alur tersebut ditunjukkan pada Gambar 9.1.



Gambar 9.1 Alur Perolehan Dataset Citra Sel Darah Merah dari Citra Apusan Darah Tepi (diadaptasi dari Tyas dkk., 2022)

Berdasarkan Gambar 9.1, alur perolehan dataset terdiri atas empat tahap utama. Pertama, akuisisi citra dilakukan menggunakan mikroskop *Olympus* CX21 dengan kamera *Optilab Advance Plus* 8 MP pada perbesaran seribu kali sehingga menghasilkan citra apusan darah tepi beresolusi tinggi sebesar 4100 kali 3075 piksel. Kedua, tahap pra-pemrosesan mencakup konversi citra ke kanal hijau, penerapan *median filtering* untuk mereduksi derau, serta pengubahan ukuran citra ke resolusi 800 kali 600 piksel guna mengurangi beban komputasi. Ketiga, tahap segmentasi menggunakan metode *watershed* untuk memisahkan setiap sel darah merah secara individual dari latar belakang maupun dari sel-sel lain yang saling bertumpuk. Keempat, tahap penyortiran dan pengelompokan sel tunggal dilakukan untuk menyeleksi sel yang memenuhi kriteria kualitas, yaitu memiliki luas minimal 500 piksel dan tidak berada di tepi citra, lalu mengklasifikasikannya ke dalam sembilan kategori sesuai label yang diberikan oleh tenaga laboratorium klinis. Hasil akhir dari keseluruhan proses ini adalah dataset yang memuat 7.108 citra sel darah merah individual dalam sembilan kelas morfologi.

Catatan penting
RBCdataset bukanlah dataset yang berdiri sendiri. Ia merupakan hasil pemotongan sel individual dari citra apusan ThalassemiaPBS menggunakan *pipeline* pra-pemrosesan tertentu. Konsekuensinya, model yang dilatih dengan RBCdataset hanya akan bekerja dengan benar pada apusan utuh

apabila apusan tersebut diproses menggunakan *pipeline* yang persis sama. Pemahaman ini menjadi kunci dari keseluruhan keputusan teknis pada tahap analisis apusan.

Pembelajaran Mendalam (*Deep Learning*)

Pembelajaran mendalam merupakan subbidang kecerdasan buatan yang menggunakan arsitektur jaringan saraf tiruan berlapis-lapis untuk mempelajari representasi hierarkis dari data secara otomatis, tanpa memerlukan rekayasa fitur secara manual. Berbeda dengan metode pembelajaran mesin konvensional yang sangat bergantung pada fitur yang didefinisikan secara eksplisit oleh perancang sistem, pembelajaran mendalam memungkinkan model mengekstraksi pola langsung dari data mentah melalui proses pelatihan yang dioptimasi dengan teknik *backpropagation* dan *gradient descent*.

Inti proses pelatihan adalah meminimalkan sebuah fungsi kerugian (*loss function*) yang mengukur selisih antara keluaran model dan label sebenarnya. Untuk tugas klasifikasi multikelas, fungsi kerugian yang lazim digunakan adalah *categorical cross-entropy*, yang dirumuskan sebagai berikut.

$$L = -\Sigma\ y_k \cdot log(\hat{y}_k)\ ,\ k = 1\ \ldots\ K \qquad (9.1)$$

Keterangan: L adalah nilai kerugian; K adalah jumlah kelas; y_k bernilai 1 apabila sampel benar-benar termasuk kelas k dan bernilai 0 untuk kelas lainnya; $\hat{y}_k$ adalah probabilitas keluaran model untuk kelas k.

Probabilitas keluaran tersebut dihasilkan oleh fungsi aktivasi *softmax* pada lapisan terakhir jaringan, yang mengubah skor mentah menjadi distribusi probabilitas dengan jumlah total sama dengan satu.

$$\hat{y}_k = exp(z_k) / \Sigma\ exp(z\_j)\ ,\ j = 1\ \ldots\ K \qquad (9.2)$$

Keterangan: z_k adalah skor mentah (*logit*) kelas k yang dihasilkan lapisan *fully-connected* terakhir.

Pembaruan bobot dilakukan menggunakan aturan penurunan gradien, di mana setiap bobot digeser berlawanan arah dengan gradien fungsi kerugian terhadap bobot tersebut, dengan besar langkah ditentukan oleh laju pembelajaran.

$$w \leftarrow w - \eta \cdot \partial L/\partial w \qquad (9.3)$$

Keterangan: w adalah bobot jaringan; η adalah laju pembelajaran (*learning rate*); ∂L/∂w adalah gradien fungsi kerugian terhadap bobot. Nilai η yang terlalu besar membuat pelatihan tidak stabil, sedangkan nilai yang terlalu kecil membuat konvergensi sangat lambat.

Convolutional Neural Network (CNN)

Convolutional Neural Network merupakan arsitektur pembelajaran mendalam yang dirancang khusus untuk memproses data dengan topologi berbentuk grid seperti citra dua dimensi. Arsitektur ini bekerja melalui operasi konvolusi yang memungkinkan pembelajaran fitur spasial secara lokal dan hierarkis. Operasi konvolusi diskret dua dimensi dirumuskan sebagai berikut.

$$y(i,j) = \Sigma\,\Sigma\; x(i-m, j-n) \cdot k(m,n) \quad (9.4)$$

Keterangan: x(i, j) adalah nilai piksel citra masukan pada koordinat (i, j); k(m, n) adalah kernel konvolusi yang dapat dipelajari; y(i, j) adalah nilai *feature map* hasil ekstraksi fitur. Penjumlahan dilakukan terhadap seluruh indeks m dan n pada rentang kernel.

Arsitektur CNN umumnya terdiri atas tiga jenis lapisan utama yang disusun berurutan. Lapisan konvolusi mengekstraksi fitur lokal melalui sekumpulan filter yang dapat dipelajari. Lapisan *pooling* melakukan *downsampling* representasi spasial sehingga model menjadi lebih invarian terhadap pergeseran kecil posisi objek sekaligus mengurangi beban komputasi. Lapisan *fully-connected* melakukan agregasi fitur global untuk menghasilkan keputusan klasifikasi akhir. Di antara lapisan-lapisan tersebut biasanya disisipkan fungsi aktivasi non-linear, yang paling umum adalah *Rectified Linear Unit* dengan rumusan sederhana berikut.

$$ReLU(z) = max(0, z) \quad (9.5)$$

Fungsi ini meneruskan nilai positif apa adanya dan memetakan seluruh nilai negatif menjadi nol, sehingga memperkenalkan non-linearitas dengan biaya komputasi yang sangat murah.

Keunggulan utama CNN dibandingkan jaringan saraf tiruan konvensional terletak pada dua mekanisme, yaitu *parameter sharing* dan *local connectivity*. Karena kernel yang sama digeser ke seluruh bagian citra, jumlah parameter yang harus dilatih berkurang drastis dibandingkan jaringan yang menghubungkan setiap piksel ke setiap neuron. Pada saat yang sama, kemampuan model untuk mengenali pola spasial tetap terjaga bahkan meningkat, karena model dipaksa mempelajari fitur yang berguna di mana pun posisinya pada citra.

Arsitektur *MobileNetV2*

MobileNetV2 merupakan arsitektur CNN yang dikembangkan oleh Sandler dan rekan-rekan pada tahun 2018 dengan tujuan utama menghadirkan kemampuan pengenalan citra yang kompetitif pada platform dengan sumber daya komputasi dan memori terbatas, seperti perangkat bergerak dan sistem tertanam. Arsitektur ini memperkenalkan dua inovasi struktural utama.

Inovasi pertama adalah *inverted residual block*. Pada blok residual konvensional seperti pada *ResNet*, koneksi lompatan menghubungkan lapisan dengan jumlah kanal yang besar, sedangkan bagian tengah blok justru dipersempit. *MobileNetV2* membalik pola tersebut: koneksi residual menghubungkan lapisan dengan jumlah kanal yang sedikit, sementara ekspansi kanal dilakukan hanya di dalam blok untuk keperluan komputasi. Pola terbalik ini menghemat memori secara signifikan karena tensor yang perlu disimpan untuk koneksi lompatan berukuran kecil.

Inovasi kedua adalah *linear bottleneck*, yaitu penghilangan fungsi aktivasi non-linear pada lapisan proyeksi akhir setiap blok. Alasannya, fungsi ReLU yang diterapkan pada ruang representasi berdimensi rendah berpotensi menghancurkan informasi karena seluruh nilai negatif dipaksa menjadi nol. Dengan membiarkan lapisan proyeksi bersifat linear, informasi pada ruang berdimensi rendah tetap terjaga.

Efisiensi *MobileNetV2* pada dasarnya berasal dari penggunaan *depthwise separable convolution*, yang memecah konvolusi standar menjadi dua tahap, yaitu konvolusi kedalaman per kanal dan konvolusi titik berukuran satu kali satu. Perbandingan biaya komputasi antara konvolusi standar dan konvolusi terpisah dapat dinyatakan sebagai berikut.

$$Rasio\ biaya\ =\ (1\ /\ N)\ +\ (1\ /\ D^2{}_k) \qquad (9.6)$$

Keterangan: N adalah jumlah kanal keluaran dan D_k adalah ukuran kernel. Untuk kernel berukuran tiga kali tiga dan jumlah kanal yang besar, rasio tersebut mendekati sepersembilan, artinya konvolusi terpisah hanya memerlukan sekitar sebelas persen biaya komputasi konvolusi standar untuk hasil yang setara.

Dalam Metodologi ini, *MobileNetV2* dipilih sebagai arsitektur dasar berdasarkan tiga pertimbangan. Pertama, arsitektur ini ringan sehingga dapat dilatih dengan cepat pada lingkungan komputasi terbatas. Kedua, ukuran model yang kecil membuatnya praktis dipasang pada aplikasi *Streamlit* tanpa memerlukan pemroses grafis kelas atas saat inferensi. Ketiga, citra sel hasil segmentasi relatif sederhana karena hanya berisi satu objek dengan latar gelap, sehingga arsitektur ringan sudah cukup mampu menangkap pola pembedanya tanpa memerlukan kapasitas model yang sangat besar.

Transfer Learning

Transfer learning adalah paradigma pelatihan yang memanfaatkan pengetahuan yang telah diperoleh dari penyelesaian satu tugas sumber untuk mempercepat dan meningkatkan kinerja model pada tugas target yang berkaitan. Paradigma ini dimotivasi oleh kenyataan bahwa representasi fitur tingkat rendah dan menengah, seperti pendeteksi tepi, tekstur, dan bentuk dasar, yang dipelajari dari dataset berskala besar cenderung bersifat umum dan dapat dimanfaatkan lintas domain.

Pendekatan ini sangat relevan pada konteks pencitraan medis, di mana perolehan data berlabel dalam jumlah besar sering terhalang oleh biaya, isu privasi, dan keterbatasan ketersediaan pakar untuk melakukan anotasi. Terdapat dua strategi utama dalam *transfer learning*. Strategi pertama adalah *feature extraction*, di mana seluruh bobot model dasar dibekukan dan hanya lapisan klasifikasi baru yang dilatih. Strategi kedua adalah *fine-tuning*, di mana sebagian atau seluruh lapisan model dasar dilatih kembali dengan laju pembelajaran yang jauh lebih kecil agar fitur yang ada dapat diadaptasi pada karakteristik domain target.

Pada buku ini kedua strategi tersebut diterapkan secara berurutan. Tahap pertama menggunakan strategi *feature extraction*, tahap kedua menggunakan strategi *fine-tuning* terbatas pada tiga puluh lapisan terakhir. Pendekatan dua tahap ini terbukti secara empiris menghasilkan konvergensi yang lebih stabil dan akurasi akhir yang lebih tinggi dibandingkan pelatihan model dari nol pada ukuran dataset yang terbatas.

Penanganan Ketidakseimbangan Kelas

Ketidakseimbangan kelas merupakan kondisi yang sangat umum dijumpai pada dataset citra medis, di mana distribusi jumlah sampel antar kelas sangat tidak proporsional. Kondisi ini menyebabkan model yang dilatih secara naif memiliki bias kuat terhadap kelas mayoritas, karena secara statistik menebak kelas mayoritas adalah strategi paling aman untuk meminimalkan kerugian. Implikasinya serius pada konteks skrining klinis: kelas-kelas morfologi yang justru paling relevan sebagai penanda patologis, seperti sel pensil, cenderung hadir dalam jumlah paling sedikit pada dataset.

Untuk mengatasi permasalahan tersebut, Metodologi ini menerapkan teknik pembobotan kelas pada fungsi kerugian selama pelatihan. Setiap kelas diberi bobot yang berbanding terbalik dengan frekuensi kemunculannya, sehingga kesalahan pada kelas langka dikenakan penalti lebih besar. Bobot kelas seimbang dihitung menggunakan formula berikut.

$$w_{\mathrm{k}} = n\_total \,/\, (\, n\_kelas \cdot n_{\mathrm{k}}\,) \qquad (9.7)$$

Keterangan: n_total adalah jumlah total sampel data latih; n_kelas adalah banyaknya kelas; n_k adalah jumlah sampel pada kelas k. Bobot ini kemudian dikalikan pada kontribusi kerugian setiap sampel sesuai kelasnya.

Pada dataset yang sangat timpang, formula di atas dapat menghasilkan bobot yang ekstrem, misalnya mencapai puluhan kali lipat untuk kelas dengan hanya beberapa sampel. Bobot seekstrem itu berisiko membuat gradien dari satu-dua sampel langka mendominasi arah pembaruan bobot secara tidak proporsional. Untuk itu Metodologi ini juga menggunakan varian bobot yang lebih lembut, yaitu akar kuadrat dari bobot seimbang.

$$w\_lembut,_{\mathrm{k}} = \sqrt{(\, w_{\mathrm{k}}\,)} \qquad (9.8)$$

Varian ini meredam efek pembobotan ekstrem sambil tetap mempertahankan perhatian lebih terhadap kelas minoritas, sehingga menjadi kompromi antara keadilan antar-kelas dan kestabilan pelatihan.

Segmentasi Citra untuk Skrining Apusan

Segmentasi citra merupakan proses membagi citra menjadi region-*region* bermakna berdasarkan karakteristik piksel, sehingga setiap objek individual dapat diekstraksi dan dianalisis secara terpisah. Dalam konteks analisis apusan darah, segmentasi menjadi tahap krusial karena memungkinkan klasifikasi dilakukan pada tingkat sel tunggal, bukan pada citra apusan secara keseluruhan. *Pipeline* segmentasi yang digunakan pada Metodologi ini terdiri atas lima tahap berurutan.

Tahap pertama adalah ekstraksi kanal hijau dari citra berwarna. Pemilihan kanal hijau bukan tanpa alasan: pada citra apusan darah dengan pewarnaan Giemsa atau Wright, sel darah merah menyerap cahaya hijau secara preferensial sehingga kontras antara badan sel dan latar belakang plasma menjadi paling tegas pada kanal tersebut.

Tahap kedua adalah penerapan *thresholding* Otsu, yaitu metode segmentasi adaptif yang secara otomatis menentukan nilai ambang optimal dengan memaksimalkan variansi antar-kelas pada histogram intensitas piksel citra. Kriteria yang dimaksimalkan dirumuskan sebagai berikut.

$$\sigma^2\_b(t) = \omega_0(t) \cdot \omega_1(t) \cdot [\mu_0(t) - \mu_1(t)]^2 \qquad (9.9)$$

Keterangan: $\sigma^2\_b(t)$ adalah variansi antar-kelas pada ambang t; $\omega_0(t)$ dan $\omega_1(t)$ berturut-turut adalah proporsi piksel kelas latar belakang dan kelas sel pada ambang t; $\mu_0(t)$ dan $\mu_1(t)$ adalah rata-rata intensitas piksel kedua kelas tersebut. Nilai ambang optimal t* adalah nilai t yang memaksimalkan $\sigma^2\_b(t)$.

Tahap ketiga adalah operasi morfologi, meliputi *opening* untuk menghilangkan derau berupa bercak kecil serta pengisian lubang untuk menutup rongga di dalam mask biner sel. Operasi *opening* sendiri merupakan gabungan dari erosi yang diikuti dilasi, yang dirumuskan sebagai berikut.

$$A \circ B = (A \ominus B) \oplus B \quad (9.10)$$

Keterangan: A adalah citra biner masukan; B adalah elemen penstruktur (*structuring element*); simbol $\ominus$ menyatakan erosi dan $\oplus$ menyatakan dilasi. Erosi menghapus objek yang lebih kecil dari elemen penstruktur, sedangkan dilasi mengembalikan ukuran objek yang tersisa.

Tahap keempat adalah penghitungan *distance transform*, yaitu peta jarak Euclidean dari setiap piksel objek ke piksel latar belakang terdekat. Piksel dengan jarak terjauh dari tepi merupakan kandidat kuat sebagai pusat sel dan dijadikan penanda awal.

$$D(p) = min \| p - q \|_2 \ \ untuk\ semua\ q \in latar\ belakang \quad (9.11)$$

Keterangan: D (p) adalah nilai *distance transform* pada piksel p; q adalah piksel latar belakang; $\|\cdot\|_2$ menyatakan jarak Euclidean.

Tahap kelima adalah penerapan algoritma *watershed*, yaitu teknik segmentasi berbasis gradien yang menggunakan analogi topografi. Citra diperlakukan sebagai bentang alam dengan lembah dan punggungan; air diibaratkan mengalir dari setiap penanda dan membentuk genangan yang meluas hingga bertemu genangan lain, dan garis pertemuan itulah yang menjadi batas antar objek. Teknik ini sangat efektif untuk memisahkan sel-sel yang saling bersinggungan atau bertumpuk, kondisi yang hampir selalu terjadi pada apusan darah tepi.

Setelah kelima tahap tersebut, dilakukan seleksi akhir berdasarkan dua kriteria: luas *region* minimal 500 piksel untuk menyingkirkan derau, dan posisi *region* yang tidak menyentuh tepi citra untuk memastikan sel yang diproses tidak terpotong.

Grad-CAM sebagai Metode *Explainable* AI

Gradient-weighted Class Activation Mapping atau Grad-CAM adalah metode interpretasi model visi komputer yang menghasilkan peta panas menunjukkan bagian citra mana yang paling berpengaruh terhadap keputusan klasifikasi. Metode ini bekerja dengan menghitung gradien skor kelas terhadap peta fitur pada lapisan konvolusi terakhir, lalu merata-ratakan gradien tersebut untuk memperoleh bobot kepentingan setiap kanal fitur.

$$\alpha_k = (1 / Z) \cdot \Sigma\Sigma\ \partial y\hat{}c / \partial A\hat{}k(i, j) \quad (9.12)$$

Keterangan: α_k adalah bobot kepentingan kanal fitur ke-k; y^c adalah skor kelas c sebelum softmax; A^k adalah peta fitur kanal ke-k; Z adalah jumlah piksel pada peta fitur.

Peta panas akhir kemudian dibentuk dengan menjumlahkan peta fitur yang telah dibobot, lalu memangkas nilai negatif menggunakan ReLU agar hanya bagian yang mendukung kelas tersebut yang ditampilkan.

$$M^c = ReLU(\Sigma\ \alpha_k \cdot A^k) \qquad (9.13)$$

Keterangan: M^c adalah peta panas Grad-CAM untuk kelas c. Peta ini kemudian dinormalisasi dan diperbesar kembali ke ukuran citra masukan agar dapat ditumpangkan di atas citra asli.

Grad-CAM tidak digunakan untuk meningkatkan akurasi, melainkan untuk memverifikasi bahwa model membuat keputusan berdasarkan alasan yang benar. Pada konteks Metodologi ini, alasan yang benar berarti model memperhatikan bentuk dan kontur sel, bukan tekstur latar belakang atau artefak pewarnaan yang kebetulan berkorelasi dengan label pada data latih. Verifikasi semacam ini penting dilakukan sebelum model dipercaya untuk dipakai menganalisis apusan dari pasien sungguhan.

Metrik Evaluasi Model Klasifikasi

Evaluasi kinerja model klasifikasi dilakukan menggunakan serangkaian metrik yang saling melengkapi agar diperoleh gambaran yang holistik dan tidak bias terhadap distribusi kelas. Seluruh metrik berikut diturunkan dari empat besaran dasar, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN), yang seluruhnya dapat dibaca langsung dari *confusion matrix*.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \qquad (9.14)$$

Accuracy mengukur proporsi prediksi yang benar terhadap keseluruhan data. Metrik ini mudah dipahami namun dapat menyesatkan pada dataset yang sangat timpang.

$$Precision = TP / (TP + FP) \qquad (9.15)$$

Precision mengukur ketepatan prediksi positif, yaitu berapa proporsi sampel yang diprediksi sebagai kelas tertentu yang memang benar termasuk kelas tersebut.

$$Recall = TP / (TP + FN) \qquad (9.16)$$

Recall atau sensitivitas mengukur kelengkapan deteksi, yaitu berapa proporsi anggota sebuah kelas yang berhasil ditemukan oleh model.

$$F1 - Score = 2 \cdot (Precision \times Recall) / (Precision + Recall) \quad (9.17)$$

F1-Score merupakan rata-rata harmonik dari *precision* dan *recall*, memberikan estimasi kinerja yang seimbang terutama pada kondisi ketidakseimbangan kelas.

Confusion matrix sendiri adalah representasi matriks dua dimensi yang mendeskripsikan distribusi lengkap prediksi benar dan salah untuk seluruh pasangan kelas. Diagonal utama matriks menunjukkan prediksi yang benar, sedangkan elemen di luar diagonal menunjukkan pola kekeliruan secara granular. Membaca *confusion matrix* memberikan informasi yang jauh lebih kaya daripada sekadar melihat angka akurasi, karena memperlihatkan kelas mana yang sering tertukar dengan kelas mana.

Pada buku ini, evaluasi difokuskan tidak hanya pada akurasi agregat, tetapi secara khusus pada *recall* kelas normal. Alasannya berkaitan dengan keamanan klinis: *recall* kelas normal yang tinggi berarti sistem jarang salah mengklasifikasikan sel normal sebagai sel patologis, sehingga risiko memicu penyelidikan klinis yang tidak perlu dapat ditekan.

Aturan Indikator Risiko

Setelah seluruh sel pada sebuah apusan diklasifikasikan, hasil pada tingkat sel perlu diagregasi menjadi kesimpulan pada tingkat subjek. Agregasi dilakukan dengan menghitung proporsi setiap jenis morfologi terhadap total sel yang terdeteksi pada apusan tersebut.

$$p_{\mathrm{k}} = n_{\mathrm{k}} / N \times 100\,\% \quad (9.18)$$

Keterangan: p_k adalah proporsi kelas k pada apusan; n_k adalah jumlah sel yang diklasifikasikan sebagai kelas k; N adalah total sel yang berhasil disegmentasi dan diklasifikasikan pada apusan tersebut.

Proporsi tersebut kemudian dibandingkan dengan tiga ambang batas morfologi yang mengacu pada karakteristik apusan darah penderita talasemia. Setiap ambang yang terlampaui dihitung sebagai satu suara, dan jumlah suara menentukan warna indikator risiko sebagaimana dirumuskan berikut ini.

$$S = 1[\,p\_target > 10\%\,] + 1[\,p\_hipokromik > 15\%\,] + 1[\,p\_teardrop > 5\%\,] \qquad (9.19)$$

Keterangan: S adalah jumlah penanda yang terlampaui; $1[\cdot]$ adalah fungsi indikator yang bernilai 1 apabila kondisi di dalamnya terpenuhi dan 0 apabila tidak.

Berdasarkan nilai S, indikator risiko ditetapkan dalam tiga kategori. Kategori HIJAU atau risiko rendah diberikan apabila S sama dengan nol, artinya tidak ada penanda morfologi yang terlampaui. Kategori KUNING atau risiko sedang diberikan apabila S sama dengan satu. Kategori MERAH atau risiko tinggi diberikan apabila S bernilai dua atau lebih. Pendekatan berbasis ambang dan pemungutan suara sederhana ini sengaja dipilih, alih-alih membangun model pembelajaran mesin kedua, karena tujuan tahap ini adalah menghasilkan keputusan yang mudah ditelusuri dan mudah dijelaskan secara klinis.

Peringatan metodologis
Ambang batas 10 persen, 15 persen, dan 5 persen yang digunakan bersifat demonstratif dan belum divalidasi secara klinis. Angka tersebut perlu dikalibrasi lebih lanjut bersama pakar hematologi menggunakan data yang jauh lebih besar sebelum dapat dipakai pada konteks nyata. Seluruh keluaran indikator risiko pada bab ini harus dibaca dalam kerangka pembuktian konsep semata.

Streamlit sebagai Kerangka Aplikasi

Streamlit merupakan kerangka kerja Python bersifat sumber terbuka yang dikembangkan khusus untuk membangun aplikasi web berbasis data secara cepat menggunakan bahasa Python murni, tanpa memerlukan pengetahuan mengenai HTML, CSS, maupun JavaScript. Mekanisme kerjanya didasarkan pada paradigma skrip reaktif, di mana setiap perubahan pada komponen antarmuka pengguna secara otomatis memicu eksekusi ulang seluruh skrip dari

atas ke bawah sehingga tampilan aplikasi selalu mencerminkan keadaan terkini dari data dan model.

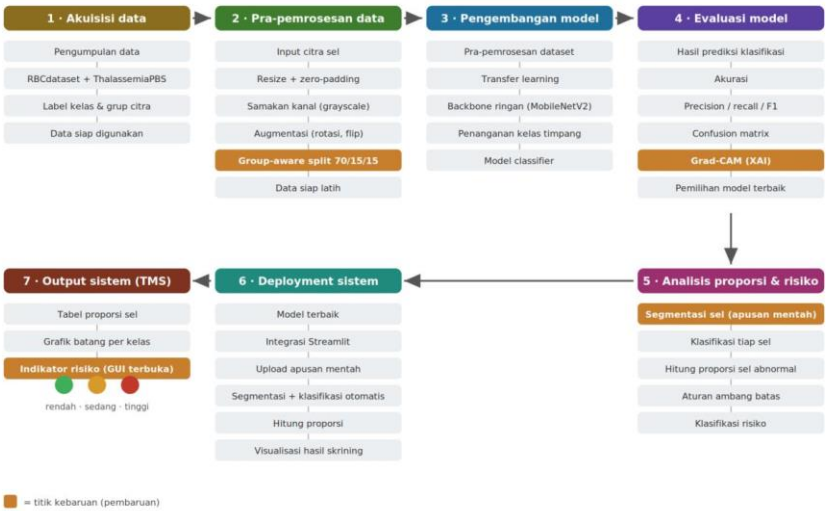
Pendekatan ini mereduksi kompleksitas pengembangan antarmuka secara signifikan dan memungkinkan peneliti memprototi pekan serta mendistribusikan aplikasi analitik berbasis pembelajaran mesin dengan siklus pengembangan yang jauh lebih singkat dibandingkan pengembangan aplikasi web konvensional. Dalam konteks Metodologi ini, *Streamlit* dipilih karena memungkinkan seluruh *pipeline* Python, mulai dari segmentasi dengan OpenCV, klasifikasi dengan *TensorFlow*, perhitungan proporsi, hingga keputusan risiko, langsung dibungkus menjadi antarmuka web interaktif yang dapat dicoba oleh pengguna non-teknis hanya melalui unggah citra dan satu klik.

➢ Metodologi

Subbab ini menguraikan kerangka kerja penyelesaian proyek secara berurutan. Seluruh metodologi disusun mengikuti delapan tahap yang saling terkait, di mana keluaran setiap tahap menjadi masukan bagi tahap berikutnya. Penyusunan yang berurutan ini penting karena beberapa keputusan teknis pada tahap akhir merupakan konsekuensi langsung dari temuan pada tahap awal, sebagaimana akan terlihat pada pembahasan mengenai kesenjangan domain.

Kerangka Kerja Sistem

Kerangka kerja keseluruhan sistem *Thalassemia Morphology Screener* ditunjukkan pada Gambar 9.2. Diagram tersebut memperlihatkan tujuh blok utama beserta rinciannya, dengan blok berwarna oranye menandai titik-titik yang menjadi pembaruan atau kebaruan dari Metodologi ini dibandingkan pendekatan terdahulu.



Gambar 9.2 Diagram Alur Sistem *Thalassemia Morphology Screener* (TMS)

Sebagaimana terlihat pada Gambar 9.2, terdapat empat titik kebaruan yang ditandai secara khusus. Pertama, penerapan *group-aware split* pada tahap pembagian data untuk mencegah kebocoran data. Kedua, penerapan *Grad*-CAM sebagai lapisan verifikasi penalaran model. Ketiga, kemampuan sistem menerima masukan berupa apusan mentah yang belum tersegmentasi. Keempat, penyajian indikator risiko tiga warna melalui antarmuka pengguna yang terbuka dan mudah dioperasikan. Keempat titik inilah yang membedakan sistem ini dari sebagian besar Metodologi terdahulu yang berhenti pada klasifikasi sel tunggal.

Studi Literatur

Tahap studi literatur dilakukan untuk memahami konsep, metode, dan Metodologi terdahulu yang berkaitan dengan pengolahan citra medis, pembelajaran mendalam, *transfer learning*, klasifikasi morfologi sel darah merah, serta skrining talasemia. Studi literatur menjadi dasar pemilihan metode sekaligus memastikan pendekatan yang diambil sesuai dengan permasalahan yang diteliti. Penelusuran literatur juga secara khusus mencakup makalah sumber dataset, karena pemahaman mengenai bagaimana dataset dibangun terbukti menjadi kunci keberhasilan pada tahap penerapan model.

Akuisisi Data

Pengumpulan data dilakukan menggunakan dua dataset citra mikroskopis sel darah merah yang saling berkaitan. Dataset pertama adalah RBCdataset, terdiri atas 7.108 citra sel darah merah tunggal dalam format skala keabuan yang dikelompokkan ke dalam sembilan kelas morfologi, yaitu *elliptocyte*, *spherocyte*, *target cell*, *acanthocyte*, *hypochromic*, *normal*, *pencil*, *stomatocyte*, dan *teardrop*. Dataset kedua adalah *ThalassemiaPBS*, terdiri atas 108 citra apusan darah tepi utuh dari lima subjek, yang mencakup empat pasien talasemia berkode T1 hingga T4 serta satu individu sehat sebagai kontrol.

Pada tahap ini juga disusun sebuah *manifest* data, yaitu tabel yang memuat jalur berkas setiap citra, label kelas yang diperoleh dari nama folder, serta identitas grup citra yang diperoleh dari nama berkas. Penyusunan identitas grup ini merupakan langkah yang tampak sepele namun sangat menentukan, karena grup tersebut nantinya dipakai untuk memastikan bahwa seluruh potongan sel yang berasal dari satu citra sumber diperlakukan sebagai satu kesatuan pada tahap pembagian data.

Pra-pemrosesan Data

Tahap pra-pemrosesan bertujuan menyiapkan citra agar siap diolah oleh model. Setiap citra sel diubah ukurannya menjadi 96 kali 96 piksel menggunakan teknik pengubahan ukuran yang disertai *zero-padding*, bukan pengubahan ukuran biasa. Pemilihan teknik ini memiliki alasan yang jelas: pengubahan ukuran biasa akan meregangkan atau menggepengkan sel sehingga bentuknya berubah, padahal bentuk membran sel, baik lonjong, bulat, maupun bergerigi, justru merupakan ciri pembeda utama antar kelas morfologi. Dengan *zero-padding*, rasio aspek asli sel tetap terjaga sehingga model belajar dari bentuk yang sebenarnya.

Seluruh citra kemudian disamakan menjadi tiga kanal agar sesuai dengan format masukan *MobileNetV2* yang dilatih pada citra berwarna. Untuk memperkaya variasi data dan mengurangi risiko *overfitting*, diterapkan augmentasi berupa pembalikan citra secara horizontal dan vertikal serta rotasi acak. Augmentasi ini diperlukan karena sel darah pada citra mikroskop tidak memiliki orientasi baku; sel yang sama dapat muncul menghadap arah mana pun pada slide.

Dengan melatih model agar tetap mengenali kelas yang sama meskipun citra diputar atau dibalik, model menjadi tahan terhadap rotasi dan lebih mampu menggeneralisasi dari jumlah data latih yang terbatas.

Tahapan terpenting pada bagian ini adalah pembagian data menggunakan teknik *group-aware split* berbasis grup citra sumber, dengan proporsi tujuh puluh persen data latih, lima belas persen data validasi, dan lima belas persen data uji. Teknik ini dipilih menggantikan pembagian acak biasa karena banyak sel berasal dari potongan citra apusan yang sama. Pembagian acak biasa berisiko menempatkan sel-sel bersaudara, yaitu sel dari slide yang sama dengan warna pewarnaan dan pencahayaan yang mirip, ke dalam himpunan latih dan himpunan uji sekaligus. Apabila hal itu terjadi, model dapat diam-diam menghafal ciri slide tersebut alih-alih ciri morfologi sel, sehingga menghasilkan akurasi uji yang terlihat tinggi namun sesungguhnya menyesatkan.

Pengembangan Model

Model klasifikasi dibangun dengan pendekatan *transfer learning* menggunakan *MobileNetV2* sebagai tulang punggung arsitektur. Di atas tulang punggung tersebut ditempelkan kepala klasifikasi baru yang terdiri atas *Global Average Pooling*, lapisan *Dropout* dengan laju 0,3, dan lapisan *Dense* dengan sembilan neuron beraktivasi *softmax*.

Proses pelatihan dilakukan dalam dua tahap. Pada tahap pertama yang disebut *head warm-up*, seluruh lapisan tulang punggung dibekukan dan hanya kepala klasifikasi yang dilatih menggunakan laju pembelajaran yang relatif besar. Alasan pembekuan ini bersifat teknis: kepala klasifikasi diinisialisasi secara acak sehingga pada awal pelatihan gradiennya besar dan berisik. Apabila seluruh tulang punggung ikut dilatih sejak awal, gradien acak tersebut berisiko merusak bobot hasil pra-pelatihan yang sudah baik, sebuah fenomena yang dikenal sebagai *catastrophic forgetting*.

Pada tahap kedua yang disebut *fine-tuning*, sebanyak tiga puluh lapisan terakhir tulang punggung dicairkan dan dilatih ulang menggunakan laju pembelajaran yang seratus kali lebih kecil. Pembatasan hanya pada tiga puluh lapisan terakhir juga memiliki alasan tersendiri. Lapisan-lapisan awal *MobileNetV2* menangkap fitur generik tingkat rendah seperti tepi, sudut, dan gradasi, yang

relevan untuk hampir semua jenis citra termasuk citra sel, sehingga fitur tersebut sebaiknya tidak diubah agar tetap tangguh. Sebaliknya, lapisan-lapisan akhir menangkap fitur tingkat tinggi yang lebih spesifik terhadap tugas, dan inilah bagian yang paling perlu disesuaikan agar model peka terhadap ciri morfologi sel darah merah.

Untuk menangani distribusi kelas yang sangat timpang, diterapkan pembobotan kelas bertingkat. Pada tahap pertama digunakan bobot seimbang penuh, sedangkan pada tahap kedua digunakan bobot lembut berupa akar kuadrat dari bobot seimbang. Ringkasan lengkap konfigurasi kedua tahap pelatihan disajikan pada Tabel 9.2.

Tabel 9.2 Ringkasan Konfigurasi Pelatihan Tahap 1 dan Tahap 2 pada Model Akhir

Parameter	**Tahap 1: Head Warm-up**	**Tahap 2: *Fine-tuning***
Status tulang punggung	Dibekukan penuh (*trainable* = *False*)	30 lapisan akhir dicairkan
Optimizer	Adam	Adam
Laju pembelajaran	1×10^{-3}	1×10^{-5} (stabil, tanpa ReduceLR)
Epoch maksimum	20	30
EarlyStopping (monitor, patience)	val_loss, patience = 6	val_loss, patience = 8
restore_best_weights	Ya	Ya
ModelCheckpoint (save_best_*only*)	tms_model_terbaik.keras	tms_model_finetuned.keras
Pembobotan kelas	cw_balanced (penuh)	cw_lembut (akar dari cw_balanced)
Akurasi validasi tercapai	± 86–90 %	± 95 %

Evaluasi Model

Evaluasi model dilakukan pada himpunan data uji yang belum pernah dilihat model selama pelatihan, menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, dan *confusion matrix*. Pelaporan *precision* dan *recall* per kelas, bukan sekadar akurasi keseluruhan, merupakan keputusan metodologis yang disengaja. Pada dataset yang tidak seimbang, akurasi keseluruhan dapat menyembunyikan kinerja buruk pada kelas minoritas, karena model yang malas dan selalu menebak kelas mayoritas tetap dapat memperoleh akurasi keseluruhan yang tampak tinggi.

Selain metrik kuantitatif, diterapkan pula *Grad*-CAM untuk memvisualisasikan area citra yang paling memengaruhi keputusan model. Verifikasi visual ini dilakukan sebelum model dipercaya untuk dipakai pada tahap analisis apusan, sehingga apabila ditemukan bahwa model ternyata memperhatikan latar belakang alih-alih tubuh sel, proses dapat dihentikan dan model diperbaiki terlebih dahulu.

Analisis Proporsi dan Risiko

Pada tahap ini model yang telah dilatih diterapkan pada citra apusan darah tepi utuh. Prosesnya terdiri atas empat langkah terintegrasi. Pertama, segmentasi sel individual dari citra apusan menggunakan *pipeline* lima tahap yang telah diuraikan pada landasan teori. Kedua, klasifikasi setiap sel yang tersegmentasi menggunakan model yang telah dilatih. Ketiga, penghitungan proporsi setiap jenis morfologi terhadap total sel yang terdeteksi. Keempat, penetapan indikator risiko berdasarkan aturan ambang batas.

Satu hal yang perlu ditegaskan pada tahap ini adalah keharusan menyamakan *pipeline* segmentasi persis dengan yang digunakan saat membangun dataset latih. Keharusan tersebut merupakan konsekuensi langsung dari fakta bahwa RBCdataset dibuat dengan memotong sel dari citra ThalassemiaPBS menggunakan *pipeline* tertentu. Apabila saat inferensi *pipeline* segmentasinya berbeda, distribusi citra sel yang masuk ke model, baik dari segi ukuran potongan, kontras, maupun derau latar, akan berbeda dari distribusi yang dipelajari model saat pelatihan. Inilah yang disebut kesenjangan domain atau *domain gap*.

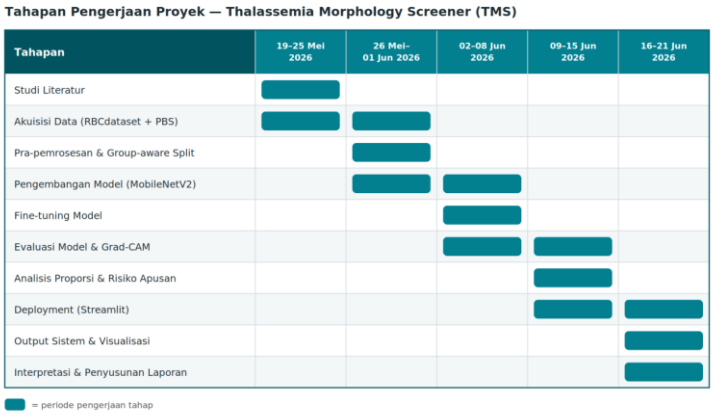
Deployment dan Keluaran Sistem

Seluruh alur pemrosesan dikemas menjadi sebuah aplikasi antarmuka pengguna berbasis *Streamlit* agar dapat digunakan oleh pengguna non-teknis. Model terbaik yang telah disimpan diintegrasikan ke dalam aplikasi sehingga pengguna cukup mengunggah citra apusan darah tepi mentah. Sistem kemudian melakukan segmentasi dan klasifikasi sel secara otomatis, menghitung proporsi tiap jenis sel, dan menampilkan hasilnya.

Keluaran sistem disajikan dalam tiga bentuk yang saling melengkapi, yaitu tabel proporsi sel, grafik batang proporsi tiap kelas, serta indikator risiko yang ditandai dengan warna hijau, kuning, atau merah. Ketiga bentuk keluaran tersebut memberikan gambaran menyeluruh mengenai komposisi morfologi sel darah merah pada apusan yang dianalisis sekaligus tingkat indikasi risiko talasemia, dengan tetap menekankan bahwa sistem berfungsi sebagai alat bantu pengambilan keputusan dan skrining awal.

Lini Masa Pelaksanaan

Keseluruhan tahapan dikerjakan dalam rentang waktu lima minggu dengan pembagian sebagaimana ditunjukkan pada Gambar 9.3. Sejumlah tahap sengaja dijadwalkan saling tumpang tindih, misalnya evaluasi model dan *Grad*-CAM yang berjalan beriringan dengan analisis proporsi apusan, karena temuan pada satu tahap sering kali menuntut peninjauan ulang tahap sebelumnya.



Gambar 9.3 Lini Masa Pengerjaan Proyek *Thalassemia Morphology Screener*

➢ Hasil dan Pembahasan

Subbab ini memaparkan hasil pengerjaan proyek secara berurutan mengikuti tahapan pada diagram alur Metodologi. Setiap tahap dijelaskan dari sisi masukan, proses yang dilakukan, hasil yang diperoleh, serta alasan mengapa hasil tersebut menentukan langkah berikutnya. Seluruh angka dan gambar yang ditampilkan merupakan hasil sebenarnya dari proses pelatihan dan pengujian yang dilakukan pada lingkungan Kaggle Notebook dengan pemroses grafis Tesla T4.

Hasil Pengumpulan Data

Proses diawali dengan menyiapkan kedua dataset. Sistem menyusun *manifest* data dengan menyaring hanya sembilan kelas resmi. Penyaringan ini penting agar label yang tidak konsisten atau folder sisa dari percobaan lama tidak ikut tercampur ke dalam data latih, yang dapat merusak proses pelatihan tanpa disadari. Hasil yang diperoleh berupa *manifest* bersih berisi 7.108 citra, sembilan kelas, dan 1.776 grup citra sumber. Karena *manifest* sudah bersih dan jumlah kelas tepat sembilan, proses dapat dilanjutkan ke tahap pembagian data.

Pemahaman mengenai asal-usul RBCdataset. Berdasarkan makalah sumber, diketahui bahwa RBCdataset sesungguhnya merupakan potongan sel individual yang diambil dari citra apusan ThalassemiaPBS yang sama. Citra apusan asli berukuran 4100 kali 3075 piksel diperkecil menjadi 800 kali 600 piksel, lalu diekstraksi kanal hijaunya, dan disegmentasi menggunakan rangkaian operasi morfologi serta *watershed*. Pemahaman ini menjadi kunci penting pada tahap analisis apusan yang akan dibahas kemudian.

Hasil Pra-pemrosesan dan Pembagian Data

Hasil pembagian data menggunakan teknik *group-aware split* menunjukkan komposisi sebanyak 5.016 sel untuk data latih, 1.014 sel untuk data validasi, dan 1.078 sel untuk data uji. Yang terpenting, pemeriksaan kebocoran grup antar himpunan menghasilkan nilai nol pada ketiga pasangan himpunan, yang biasa ditulis sebagai 0/0/0.

Karena tidak ada satu pun grup yang bocor antar himpunan, akurasi yang dihasilkan model dapat dipercaya sebagai ukuran kinerja yang jujur, bukan akurasi semu akibat kebocoran data. Inilah alasan teknik ini dijadikan salah satu titik kebaruan Metodologi.

Perlu ditekankan bahwa banyak Metodologi klasifikasi citra medis melaporkan akurasi sangat tinggi namun tidak memeriksa aspek ini, sehingga angka yang dilaporkan sulit dibandingkan secara adil.

Hasil Pengembangan Model

Pada tahap pertama pelatihan, dengan tulang punggung dibekukan dan hanya kepala klasifikasi yang dilatih, akurasi validasi mencapai kisaran 86 sampai 90 persen. Karena hasil tersebut masih dapat ditingkatkan, proses dilanjutkan ke tahap *fine-tuning*.

Pada percobaan awal *fine-tuning*, ditemukan bahwa pelatihan yang berhenti terlalu dini hanya mencapai akurasi sekitar 91 persen. Setelah kesabaran pelatihan diperbesar, yaitu dengan menaikkan nilai *patience* dan menjaga laju pembelajaran tetap stabil, akurasi validasi meningkat secara konsisten hingga mencapai sekitar 95 persen. Perbandingan lengkap kedua percobaan disajikan pada Tabel 9.3.

Tabel 9.3 Perbandingan Jadwal *Epoch* pada Percobaan Awal dan Model Akhir

Aspek	**Percobaan Awal**	**Percobaan Final (Model Akhir)**
Epoch maksimum tahap 1	15	20
Patience EarlyStopping tahap 1	4	6
Epoch maksimum *fine-tuning*	25	30
Patience EarlyStopping fine-tuning	7	8
Pengatur laju pembelajaran tambahan	ReduceLROnPlateau (faktor 0,5; patience 3; min_lr 1×10^{-7})	Tidak digunakan — LR tetap stabil 1×10^{-5}
Akurasi validasi terbaik	± 91–94 %	± 95 %

Untuk memahami mengapa perubahan jadwal ini berpengaruh besar, perlu dipahami terlebih dahulu cara kerja mekanisme *EarlyStopping*, *patience*, dan *ModelCheckpoint*. Mekanisme *EarlyStopping* dengan parameter pemantauan *val_loss* dan *patience* bernilai N akan menghentikan pelatihan apabila *val_loss* tidak

membaik selama N *epoch* berturut-turut, dengan tujuan mencegah *overfitting* sekaligus menghemat waktu komputasi.

Pada percobaan awal, nilai *patience* yang digunakan masih terlalu kecil, yaitu empat dan tujuh, sehingga pelatihan berhenti terlalu cepat ketika *val_loss* sempat stagnan sejenak. Padahal apabila dilanjutkan beberapa *epoch* lagi, model masih dapat membaik karena *val_loss* kerap berfluktuasi naik-turun kecil sebelum benar-benar menemukan arah yang lebih baik. Pada percobaan final, *patience* diperbesar menjadi enam dan delapan sehingga pelatihan menjadi lebih sabar menunggu sebelum berhenti, memberi model lebih banyak kesempatan keluar dari dataran stagnan sementara.

Parameter *restore_best_weights* yang diaktifkan turut memastikan bahwa bobot yang dipakai pada akhir pelatihan adalah bobot dari *epoch* dengan *val_loss* terbaik, bukan sekadar bobot dari *epoch* terakhir yang dijalankan. Sebagai pengaman tambahan, *ModelCheckpoint* dengan opsi menyimpan hanya model terbaik bekerja dengan menyimpan berkas model pada setiap *epoch* hanya apabila *val_loss*-nya merupakan yang terbaik sejauh ini.

Mekanisme *ReduceLROnPlateau* yang dipakai pada percobaan awal, yang berfungsi memperkecil laju pembelajaran secara otomatis saat *val_loss* mandek, justru terbukti kontraproduktif ketika dipasangkan dengan *patience* yang juga kecil. Begitu laju pembelajaran diperkecil, perubahan bobot per *epoch* menjadi sangat kecil sehingga *val_loss* tampak mandek lebih cepat, dan *EarlyStopping* pun ikut terpicu lebih awal. Kedua mekanisme tersebut saling memperkuat efek berhenti terlalu dini.

Selanjutnya, penanganan ketidakseimbangan kelas juga menjadi pertimbangan penting dalam pelatihan. Sebagai ilustrasi besarnya efek formula pembobotan kelas, Tabel 9.4 menyajikan perhitungan bobot menggunakan sebaran kelas pada data uji sebagai proksi tingkat ketidakseimbangan.

Tabel 9.4 Ilustrasi Perhitungan Bobot Kelas Seimbang dan Bobot Lembut

Kelas	**Jumlah Sampel (ilustrasi)**	**Bobot cw_balanced**	**Bobot cw_lembut (√)**
Elliptocyte	180	0,67 ×	0,82 ×
Spherocyte (SPERO_bulat)	111	1,08 ×	1,04 ×
Target *cell* (TARGETSEL)	130	0,92 ×	0,96 ×
Acanthocyte	38	3,15 ×	1,78 ×
Hypochromic	27	4,44 ×	2,11 ×
Normal	216	0,55 ×	0,74 ×
Pencil	2	59,89 ×	7,74 ×
Stomatocyte	60	2,00 ×	1,41 ×
Teardrop	314	0,38 ×	0,62 ×

Pada Tabel 9.4 terlihat bahwa kelas *pencil* dapat memperoleh bobot hingga sekitar enam puluh kali di bawah skema bobot seimbang, angka yang sangat ekstrem karena sampelnya hanya dua. Pada tahap pertama, ketika hanya kepala klasifikasi yang dilatih dengan parameter yang relatif sedikit dan laju pembelajaran yang lebih besar, bobot setegas itu masih relatif aman dan justru membantu model langsung memperhatikan kelas langka sejak awal.

Namun pada tahap kedua, jauh lebih banyak parameter yang ikut diperbarui, yaitu tiga puluh lapisan tulang punggung ditambah kepala klasifikasi, dengan laju pembelajaran yang sudah sangat kecil. Bobot seekstrem enam puluh kali pada tahap ini berisiko membuat gradien dari satu atau dua sampel langka mendominasi arah pembaruan bobot secara tidak proporsional. Mengakar-kuadratkan bobot menjadi *cw_lembut* meredam efek ekstrem tersebut, dari enam puluh kali menjadi sekitar tujuh koma tujuh kali, sambil tetap mempertahankan perhatian ekstra terhadap kelas minoritas.

Hasil Evaluasi Model

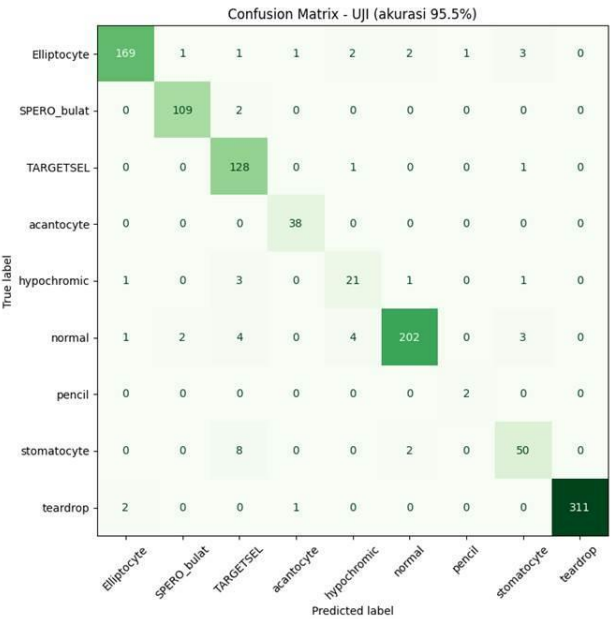
Model akhir dievaluasi pada 1.078 citra sel data uji yang belum pernah dilihat selama pelatihan. Hasil yang diperoleh adalah akurasi sebesar 94,8 persen dengan *recall* kelas normal sekitar 94 persen. Rincian metrik per kelas ditunjukkan pada Tabel 9.5.

Tabel 9.5 Metrik Kinerja Model per Kelas pada Data Uji

Kelas	***Precision***	***Recall***	**F1-score**	***Support***
Elliptocyte	0,977	0,939	0,958	180
Spherocyte (SPERO)	0,973	0,982	0,978	111
Target *cell*	0,877	0,985	0,928	130
Acanthocyte	0,950	1,000	0,974	38
Hypochromic	0,750	0,778	0,764	27
Normal	0,976	0,935	0,955	216
Pencil *	0,667	1,000	0,800	2
Stomatocyte	0,862	0,833	0,847	60
Teardrop	1,000	0,990	0,995	314
Akurasi keseluruhan	—	—	0,948	1.078

Tanda bintang pada kelas *pencil* menunjukkan bahwa kelas tersebut hanya memiliki dua sampel pada data uji, sehingga metriknya tidak dapat diandalkan dan secara jujur dinyatakan sebagai keterbatasan. Nilai *recall* sempurna sebesar 1,000 pada kelas tersebut berarti kedua sampel berhasil ditemukan, namun angka tersebut tidak memberikan informasi statistik yang bermakna mengenai kemampuan model pada kelas ini secara umum.

Distribusi kesalahan klasifikasi secara lengkap dapat dibaca pada *confusion matrix* yang ditunjukkan pada Gambar 9.4.

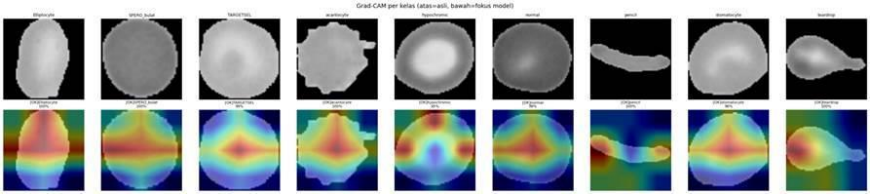


Gambar 9.4 *Confusion Matrix* Hasil Klasifikasi pada Data Uji; Diagonal yang Tebal Menunjukkan Sebagian Besar Sel Terklasifikasi dengan Benar

Berdasarkan Tabel 9.5 dan Gambar 9.4, model menunjukkan kinerja yang sangat kuat pada kelas *teardrop*, *acanthocyte*, dan *spherocyte*. Kelemahan utama terdapat pada kelas *hypochromic* yang cenderung mengalami kelebihan prediksi, serta pada sebagian kekeliruan antara *stomatocyte* dengan *target cell*. Kekeliruan yang terakhir sangat dapat dipahami secara morfologis, karena keduanya sama-sama memiliki pola khas pada bagian tengah sel: *stomatocyte* memiliki celah pucat memanjang sedangkan *target cell* memiliki bintik hemoglobin di tengah zona pucat. Pada resolusi 96 kali 96 piksel, perbedaan halus tersebut memang sulit dibedakan bahkan oleh mata manusia.

Karena kemampuan mengenali sel normal sudah baik dengan *recall* sebesar 94 persen, model dinilai layak digunakan untuk tahap analisis apusan. Sebagaimana telah dijelaskan pada landasan teori, *recall* kelas normal merupakan prasyarat keamanan klinis utama pada sistem skrining.

Untuk memastikan model mengambil keputusan secara beralasan, diterapkan Grad-CAM dengan hasil sebagaimana ditunjukkan pada Gambar 9.5.



Gambar 9.5 Visualisasi Grad-CAM untuk Satu Contoh Tiap Kelas; Baris Atas menunjukkan Citra Sel Asli dan Baris Bawah menunjukkan Area Perhatian Model

Peta panas pada Gambar 9.5 menunjukkan bahwa model memfokuskan perhatian pada tubuh dan kontur sel, bukan pada latar belakang. Pembeda jenis sel terletak pada bentuk membrannya. Dengan demikian, keputusan model dapat ditelusuri dan di-pertanggungjawabkan secara morfologis, bukan sekadar merupakan hasil korelasi statistik yang kebetulan.

Hasil Segmentasi pada Citra Apusan Utuh

Pipeline segmentasi lima tahap diterapkan pada satu citra apusan perwakilan dari masing-masing kelima subjek. Hasil ekstraksi sel disajikan pada Tabel 9.6.

Tabel 9.6 Jumlah Sel yang Berhasil disegmentasi per Subjek

Subjek	**Jumlah Sel Terdeteksi**
T1 (pasien talasemia)	80
T2 (pasien talasemia)	108
T3 (pasien talasemia)	72
T4 (pasien talasemia)	101
Individu sehat (kontrol)	100
Total	461

Hasil pada Tabel 9.6 menunjukkan bahwa *pipeline* pengolahan citra berhasil mengekstraksi rata-rata sekitar 92 sel per citra apusan, dengan rentang antara 72 hingga 108 sel bergantung pada kepadatan sel di masing-masing citra. Variasi jumlah tersebut mencerminkan

perbedaan kepadatan sebaran sel pada tiap apusan, sekaligus menunjukkan bahwa segmentasi *watershed* mampu menangani objek yang saling bertumpuk secara konsisten antar citra.

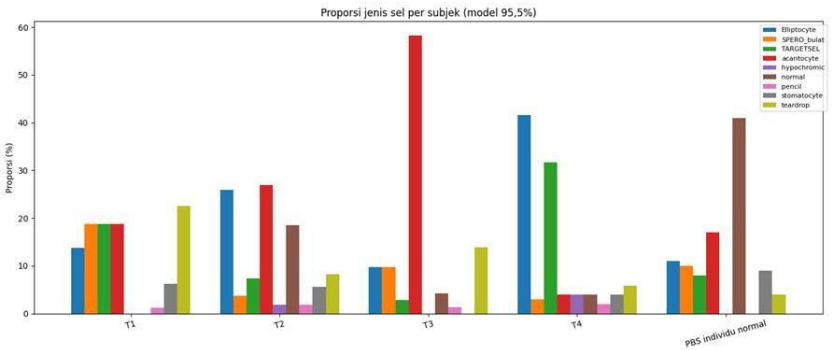
Hasil Analisis Proporsi dan Indikator Risiko

Pada percobaan awal penerapan model ke apusan utuh, klasifikasi sel hasil segmentasi memberi hasil yang tidak masuk akal. Penyelidikan menunjukkan bahwa penyebabnya adalah perbedaan cara pra-pemrosesan antara data latih dan data inferensi, yaitu kesenjangan domain yang telah diantisipasi sejak tahap pemahaman data. Setelah *pipeline* apusan disamakan persis dengan makalah sumber, hasil klasifikasi menjadi wajar. Proporsi jenis sel utama beserta indikator risiko tiap subjek ditunjukkan pada Tabel 9.7.

Tabel 9.7 Proporsi Jenis Sel Utama dan Indikator Risiko per Subjek

Subjek	**Normal**	**Target**	**Teardrop**	**Acanth.**	**Ellip.**	**Risiko**
T1 (pasien)	0 %	18,8 %	22,5 %	18,8 %	13,8 %	MERAH
T2 (pasien)	18,5 %	7,4 %	8,3 %	26,9 %	25,9 %	KUNING
T3 (pasien)	4,2 %	2,8 %	13,9 %	58,3 %	9,7 %	KUNING
T4 (pasien)	4,0 %	31,7 %	5,9 %	4,0 %	41,6 %	MERAH
Individu sehat	41,0 %	8,0 %	4,0 %	17,0 %	11,0 %	HIJAU

Aturan ambang yang digunakan bersifat demonstratif, yaitu *target cell* melebihi sepuluh persen, sel hipokromik melebihi lima belas persen, dan *teardrop* melebihi lima persen. Dua penanda yang terlampaui menghasilkan indikator MERAH, satu penanda menghasilkan KUNING, dan nol penanda menghasilkan HIJAU. Sebaran proporsi seluruh kelas untuk kelima subjek divisualisasikan pada Gambar 9.6.



Gambar 9.6 Proporsi Jenis Sel per Subjek Hasil Skrining Apusan

Hasil pada Tabel 9.7 dan Gambar 9.6 menunjukkan pola yang konsisten dengan harapan klinis. Subjek sehat memiliki proporsi sel normal tertinggi, yaitu 41 persen, dan tergolong risiko rendah dengan indikator HIJAU. Sebaliknya, keempat pasien T1 hingga T4 didominasi sel abnormal dan tergolong risiko sedang hingga tinggi dengan indikator KUNING atau MERAH. Yang paling mencolok adalah subjek T1 yang sama sekali tidak memiliki sel yang diklasifikasikan sebagai normal. Karena pola ini berhasil membedakan subjek sehat dari pasien, konsep skrining terbukti bekerja sebagai pembuktian konsep.

Pembahasan Kesenjangan Domain

Kegagalan pada percobaan awal penerapan apusan layak dibahas secara khusus karena merupakan salah satu pelajaran metodologis paling berharga dari Metodologi ini. Kesenjangan domain terjadi ketika distribusi data pada saat inferensi berbeda dari distribusi data pada saat pelatihan. Perbedaan tersebut dapat berupa hal-hal yang tampak sepele: ukuran potongan citra, tingkat kontras, cara penghitaman latar belakang, bahkan urutan operasi pra-pemrosesan.

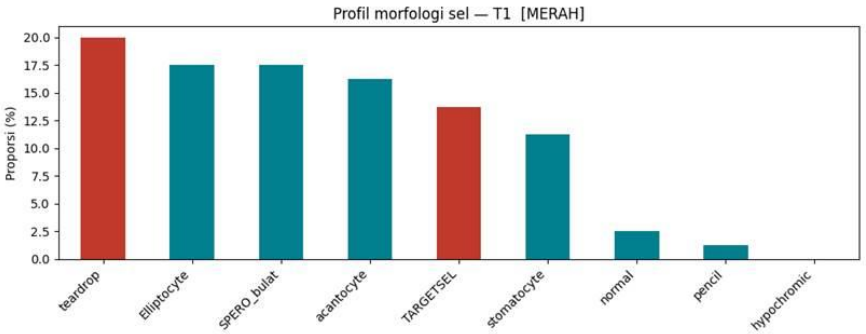
Untuk mengatasi hal tersebut, *pipeline* segmentasi yang dipakai saat membangun RBCdataset direplikasi sepenuhnya, dengan setiap tahapannya memiliki alasan tersendiri. Citra apusan terlebih dahulu diperkecil ke ukuran 800 kali 600 piksel untuk menyamakan skala dengan skala yang dipakai saat RBCdataset dibuat, agar ukuran relatif sel terhadap citra tetap konsisten. Selanjutnya dilakukan ekstraksi kanal hijau karena kanal tersebut memberi kontras terbaik antara badan sel dan latar. Citra kemudian melalui *thresholding* Otsu yang secara otomatis mencari ambang intensitas optimal tanpa perlu menetapkan ambang manual yang bisa berubah-ubah antar citra.

Tahap berikutnya adalah operasi morfologi berupa *opening* dan pengisian lubang, yang bertujuan membersihkan derau kecil dan memastikan area sel terisi penuh sebagai satu *region* utuh. Terakhir, diterapkan *watershed* berbasis *distance transform*, karena banyak sel pada apusan saling bersentuhan atau bertumpuk. *Watershed* diperlukan untuk memisahkan sel-sel yang menempel ini menjadi *instance* individual, mengingat model klasifikasi dilatih untuk mengenali satu sel dalam satu citra, bukan gerombolan sel. Dengan mereplikasi seluruh langkah tersebut, distribusi citra sel hasil segmentasi pada tahap inferensi menjadi sekonsisten mungkin dengan distribusi citra yang dipelajari model selama pelatihan.

Pelajaran metodologis
Sebuah model dengan akurasi tinggi di atas kertas dapat gagal total di lapangan hanya karena perbedaan kecil pada *pipeline* pra-pemrosesan. Konsistensi antara *pipeline* pelatihan dan *pipeline* inferensi bukan sekadar praktik yang baik, melainkan syarat mutlak. Setiap sistem berbasis citra harus mendokumentasikan *pipeline* pra-pemrosesannya secara lengkap agar dapat direplikasi.

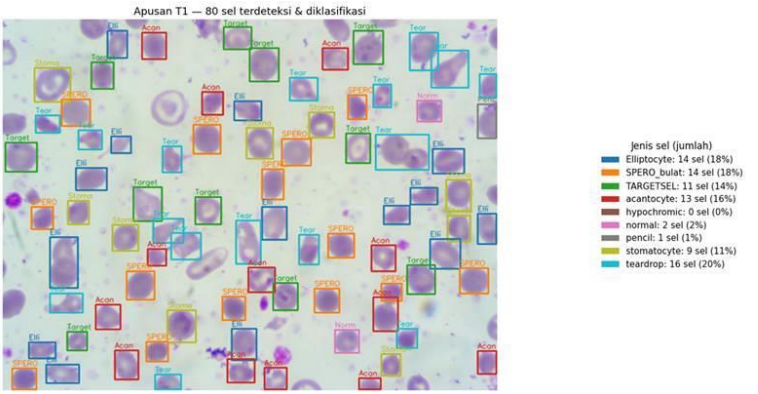
Keluaran Sistem TMS

Sebagai contoh keluaran, Gambar 9.7 menampilkan profil morfologi hasil skrining pada apusan subjek T1 dalam bentuk grafik batang, di mana batang berwarna merah menandai kelas yang proporsinya melampaui ambang penanda risiko.



Gambar 9.7 Contoh Keluaran Skrining pada Apusan T1 dengan 80 Sel Terdeteksi dan Indikator Risiko MERAH

Selain profil agregat, sistem juga menyajikan anotasi visual per sel pada citra apusan asli sebagaimana ditunjukkan pada Gambar 9.8. Setiap sel yang terdeteksi diberi kotak pembatas berwarna sesuai jenis kelas hasil prediksi, disertai keterangan jumlah dan persentase masing-masing kelas pada panel legenda.



Gambar 9.8 Anotasi Tiap Sel pada Apusan T1 dengan Kotak Berwarna Sesuai Jenis Sel Beserta Hitungannya

Pada apusan T1, sistem mendeteksi 80 sel dengan komposisi didominasi *teardrop* sebesar 20 persen, *elliptocyte* dan *spherocyte* masing-masing 17,5 persen, *acanthocyte* 16 persen, dan *target cell* 14 persen. Karena proporsi *target cell* melebihi sepuluh persen dan proporsi *teardrop* melebihi lima persen, subjek ini dikategorikan berisiko tinggi dengan indikator MERAH dan diinterpretasikan perlu menjalani pemeriksaan lanjut untuk konfirmasi.

Anotasi visual per sel memiliki nilai praktis tersendiri di luar sekadar keindahan tampilan. Dengan melihat kotak berwarna langsung di atas citra asli, tenaga laboratorium dapat melakukan pemeriksaan silang secara cepat: apabila sistem menandai sebuah sel sebagai *target cell* padahal secara kasat mata sel tersebut jelas normal, kesalahan tersebut segera terlihat. Kemampuan pemeriksaan silang semacam ini merupakan bentuk transparansi yang penting untuk membangun kepercayaan pengguna terhadap sistem.

➢ Implementasi Perangkat Lunak

Subbab ini menyajikan potongan kode inti dari setiap tahap sistem. Kode ditulis dalam bahasa Python dengan pustaka utama OpenCV untuk pengolahan citra, *TensorFlow* dan Keras untuk pemodelan, serta *Streamlit* untuk antarmuka pengguna. Seluruh potongan kode disertai penjelasan agar dapat diikuti dan direproduksi oleh pembaca.

Pengubahan Ukuran dan Ekstraksi Kanal Hijau

Langkah pertama *pipeline* segmentasi adalah menyamakan skala citra apusan dan mengambil kanal hijau. Perlu diperhatikan bahwa OpenCV membaca citra dalam urutan kanal biru, hijau, merah, sehingga kanal hijau berada pada indeks satu.

```
# ======================================
# RESIZE CITRA + EKSTRAKSI GREEN CHANNEL
# ======================================
import cv2
import numpy as np

img   = cv2.imread(path)
img   = cv2.resize(img, (800, 600))
green = img[:, :, 1]        # kanal hijau (indeks 1 dari BGR)
```

Thresholding Otsu dan Operasi Morfologi

Setelah kanal hijau diperoleh, citra dipisahkan menjadi latar depan dan latar belakang menggunakan metode Otsu. Bendera THRESH_BINARY_INV dipakai karena sel tampak lebih gelap daripada latar pada kanal hijau, sehingga inversi diperlukan agar sel menjadi bernilai putih pada mask biner.

```
# ======================================
# THRESHOLDING OTSU
# ======================================
_, mask = cv2.threshold(
    green, 0, 255,
    cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU
)

# ======================================
# OPERASI MORFOLOGI (OPENING + ISI LUBANG)
# ======================================
from scipy import ndimage as ndi

kernel = np.ones((3, 3), np.uint8)
mask   = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel,
iterations=2)
mask   = ndi.binary_fill_holes(mask > 0).astype(np.uint8) * 255
```

Distance Transform*, Penanda, dan *Watershed

Tahap ini merupakan inti dari pemisahan sel yang saling menempel. Piksel dengan nilai *distance transform* melebihi empat puluh persen dari nilai maksimum dianggap sebagai area yang pasti merupakan bagian dalam sel, dan dijadikan penanda awal bagi algoritma *watershed*.

```
# ======================================
# DISTANCE TRANSFORM + PENANDA (MARKER)
# ======================================
dist = cv2.distanceTransform(mask, cv2.DIST_L2, 5)

# area "pasti sel" = piksel dengan jarak > 40% jarak maksimum
_, pasti = cv2.threshold(dist, 0.4 * dist.max(), 255, 0)
pasti    = np.uint8(pasti)

# area "tidak diketahui" (perbatasan antar sel)
tidak_diketahui = cv2.subtract(
    cv2.dilate(mask, kernel, iterations=3), pasti
)
```

```
# label awal tiap marker
_, markers = cv2.connectedComponents(pasti)
markers    = markers + 1
markers[tidak_diketahui == 255] = 0

# ====================================
# SEGMENTASI WATERSHED
#====================================
markers = cv2.watershed(
   cv2.cvtColor(green, cv2.COLOR_GRAY2BGR), markers
)
```

Ekstraksi *Region of Interes*t per Sel

Setelah setiap sel memperoleh label unik, dilakukan ekstraksi kotak pembatas untuk masing-masing label. Dua aturan penyaringan diterapkan, yaitu *region* yang terlalu kecil disingkirkan karena kemungkinan besar merupakan derau, dan *region* yang menyentuh tepi citra disingkirkan karena selnya terpotong sehingga morfologinya tidak dapat dinilai secara utuh.

```
#
========================================
# EKSTRAKSI ROI PER SEL (BOUNDING BOX)
#
========================================
sel = []
for label in np.unique(markers):
   if label <= 1:
      continue            # label 1 = latar, -1 = batas watershed

   ys, xs = np.where(markers == label)
   if len(xs) < 500:
      continue            # saring region terlalu kecil (noise)

   x, y = xs.min(), ys.min()
   w, h = xs.max() - x, ys.max() - y
```

```
    # saring sel yang menyentuh tepi citra
    if x <= 1 or y <= 1 or x + w >= 799 or y + h >= 599:
        continue

    crop = green[y:y+h, x:x+w] * (
        markers[y:y+h, x:x+w] == label).astype(np.uint8)
    sel.append(crop)

print(f"{len(sel)} sel berhasil diekstraksi")
```

Pembangunan Arsitektur Model

Arsitektur model dibangun dengan menempelkan kepala klasifikasi baru di atas tulang punggung *MobileNetV2* yang sudah dilatih pada *ImageNet*. Lapisan augmentasi ditempatkan di dalam model agar augmentasi hanya aktif saat pelatihan dan otomatis nonaktif saat inferensi.

```
#=======================================
# MEMBANGUN ARSITEKTUR MODEL
#=======================================
import tensorflow as tf
from tensorflow.keras import layers, models

IMG = 96
N   = 9                    # jumlah kelas morfologi

base = tf.keras.applications.MobileNetV2(
    input_shape=(IMG, IMG, 3),
    include_top=False,
    weights="imagenet"
)
base.trainable = False          # backbone dibekukan untuk tahap
1

inp = tf.keras.Input((IMG, IMG, 3))
x   = layers.RandomFlip("horizontal_and_vertical")(inp)
x   = layers.RandomRotation(0.3)(x)
x   = layers.Rescaling(1/127.5, offset=-1)(x)
x   = base(x, training=False)
```

```
x   = layers.GlobalAveragePooling2D()(x)
x   = layers.Dropout(0.3)(x)
out = layers.Dense(N, activation="softmax")(x)

model = models.Model(inp, out)
```

Perhitungan Bobot Kelas

Bobot kelas dihitung sebelum pelatihan dimulai. Dua varian dihasilkan sekaligus, yaitu bobot seimbang penuh untuk tahap pertama dan bobot lembut berupa akar kuadratnya untuk tahap kedua.

```
# ======================================
# PEMBOBOTAN KELAS (CLASS WEIGHT)
#======================================
import numpy as np
from sklearn.utils.class_weight import compute_class_weight

kelas = sorted(df_latih["label"].unique())
k2i   = {c: i for i, c in enumerate(kelas)}
bobot = compute_class_weight(
    "balanced",
    classes=np.array(kelas),
    y=df_latih["label"].values
)
cw_balanced = {k2i[c]: float(w)       for c, w in zip(kelas, bobot)}
cw_lembut   = {k2i[c]: float(np.sqrt(w)) for c, w in zip(kelas, bobot)}
```

Pelatihan Tahap 1 dan Tahap 2

Tahap pertama melatih hanya kepala klasifikasi dengan laju pembelajaran relatif besar dan bobot kelas penuh. Tahap kedua mencairkan tiga puluh lapisan terakhir tulang punggung dengan laju pembelajaran seratus kali lebih kecil dan bobot kelas lembut.

```
# ======================================
# TAHAP 1 - HEAD WARM-UP
# ======================================
model.compile(
    optimizer=tf.keras.optimizers.Adam(1e-3),
    loss="sparse_categorical_crossentropy",
    metrics=["accuracy"]
)

model.fit(
    ds_latih, validation_data=ds_val,
    epochs=20,
    class_weight=cw_balanced,
    callbacks=[
        tf.keras.callbacks.EarlyStopping(
            monitor="val_loss", patience=6,
            restore_best_weights=True),
        tf.keras.callbacks.ModelCheckpoint(
            "tms_model_terbaik.keras",
            monitor="val_loss", save_best_only=True)
    ]
)

# ======================================
# TAHAP 2 - FINE-TUNING
# ======================================
base.trainable = True
for layer in base.layers[:len(base.layers) - 30]:
    layer.trainable = False      # hanya 30 lapisan akhir ikut dilatih

model.compile(
    optimizer=tf.keras.optimizers.Adam(1e-5),
    loss="sparse_categorical_crossentropy",
    metrics=["accuracy"]
)

hist = model.fit(
    ds_latih, validation_data=ds_val,
```

```
    epochs=30,
    class_weight=cw_lembut,
    callbacks=[
        tf.keras.callbacks.EarlyStopping(
            monitor="val_loss", patience=8,
            restore_best_weights=True),
        tf.keras.callbacks.ModelCheckpoint(
            "tms_model_finetuned.keras",
            monitor="val_loss", save_best_only=True)
    ]
)
```

Evaluasi Model

```
# ======================================
# EVALUASI MODEL PADA DATA UJI
# ======================================
from sklearn.metrics import classification_report,
confusion_matrix

y_benar, y_tebak = [], []
for X, Y in ds_uji:
    p = model.predict(X, verbose=0)
    y_tebak.extend(np.argmax(p, axis=1))
    y_benar.extend(Y.numpy())

y_benar, y_tebak = np.array(y_benar), np.array(y_tebak)
akurasi = (y_benar == y_tebak).mean()

print(f"Akurasi uji: {akurasi:.1%}")
print(classification_report(y_benar, y_tebak,
                            target_names=kelas, digits=3))
```

Implementasi *Grad*-CAM

Grad-CAM diimplementasikan dengan memisahkan tulang punggung dan kepala klasifikasi, lalu merekam gradien skor kelas terhadap keluaran konvolusi terakhir menggunakan mekanisme *GradientTape*.

```
# ======================================
# GRAD-CAM - INTERPRETASI VISUAL MODEL
# ======================================
base_layer = next(l for l in model.layers
                  if "mobilenet" in l.name.lower())
rescale    = next(l for l in model.layers
                  if "rescaling" in l.name.lower())
kepala     = model.layers[model.layers.index(base_layer) + 1:]
def grad_cam(img):
    x = rescale(tf.expand_dims(img, 0))
    with tf.GradientTape() as tape:
        conv = base_layer(x, training=False)
        tape.watch(conv)
        h = conv
        for layer in kepala:
            h = layer(h, training=False)
        idx  = tf.argmax(h[0])
        skor = h[:, idx]
    grad  = tape.gradient(skor, conv)
    bobot = tf.reduce_mean(grad, axis=(0, 1, 2))
    peta  = tf.reduce_sum(conv[0] * bobot, axis=-1)
    peta  = tf.maximum(peta, 0) / (tf.reduce_max(peta) + 1e-8)
    return tf.image.resize(peta[..., None], (96, 96)).numpy()[..., 0],
int(idx)
```

Aturan Indikator Risiko dan Aplikasi *Streamlit*

Fungsi penentuan risiko ditulis sesederhana mungkin agar mudah ditelusuri dan mudah diaudit. Setiap ambang yang terlampaui dihitung sebagai satu suara, lalu jumlah suara diterjemahkan menjadi warna indikator.

```
# ======================================
# ATURAN INDIKATOR RISIKO
# ======================================
def indikator_risiko(proporsi):
    suara = 0
    if proporsi.get("TARGETSEL", 0)   > 10: suara += 1
    if proporsi.get("hypochromic", 0) > 15: suara += 1
    if proporsi.get("teardrop", 0)    >  5: suara += 1
```

```
    if suara >= 2: return "MERAH",  "Risiko tinggi"
    if suara == 1: return "KUNING", "Risiko sedang"
    return "HIJAU", "Risiko rendah"

# ====================================
# APLIKASI STREAMLIT (app.py)
# ====================================
import streamlit as st

@st.cache_resource
def muat_model():
    return
tf.keras.models.load_model("tms_model_finetuned.keras")

model = muat_model()

st.title("Thalassemia Morphology Screener (TMS)")
st.caption("Prototipe proof-of-concept - bukan alat diagnosis
tervalidasi")

berkas = st.file_uploader("Unggah citra apusan darah tepi",
                          type=["png", "jpg", "jpeg"])

if berkas is not None:
    sel      = segmentasi_paper(berkas)
    hasil    = [kelas[np.argmax(model.predict(s, verbose=0))] for
s in sel]
    proporsi = hitung_proporsi(hasil)
    warna, keterangan = indikator_risiko(proporsi)

    st.subheader(f"Indikator risiko: {warna} - {keterangan}")
    st.bar_chart(proporsi)
    st.dataframe(tabel_proporsi(proporsi))
```

Alur kerja berkas aplikasi mengulang persis fungsi-fungsi yang sudah divalidasi pada tahap evaluasi dan analisis. Model dimuat sekali di awal dan disimpan dalam *cache* menggunakan dekorator khusus agar tidak perlu dimuat ulang setiap kali pengguna mengunggah citra baru. Fungsi segmentasi yang dipakai identik dengan fungsi yang digunakan pada notebook Metodologi, sehingga konsistensi terhadap apa yang telah diverifikasi tetap terjaga. Aplikasi juga memuat keterangan eksplisit bahwa hasil yang ditampilkan merupakan demonstrasi, bukan diagnosis tervalidasi.

➢ Contoh Studi Kasus Pelengkap: Pengenalan Pola pada Data Hematologi Tabular

Subbab ini menyajikan studi kasus kedua yang bekerja pada data hematologi berbentuk tabular, sebagai pelengkap dari sistem TMS yang bekerja pada data citra. Tujuan studi kasus ini adalah memprediksi status rawat inap pasien berdasarkan profil parameter hematologi hasil pemeriksaan laboratorium. Meskipun berbeda jenis data, alur berpikir metodologisnya serupa: memahami data terlebih dahulu, merekayasa fitur berdasarkan pemahaman tersebut, membangun model, lalu menginterpretasikan keputusannya.

Karakteristik Dataset

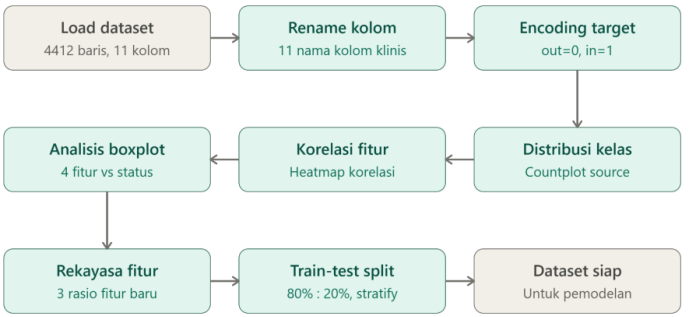
Dataset yang digunakan berupa rekam medis elektronik pasien dengan sebelas kolom. Sembilan kolom di antaranya merupakan parameter hematologi numerik, yaitu hematokrit, hemoglobin, *eritrosit*, leukosit, trombosit, serta tiga indeks *eritrosit* berupa MCH, MCHC, dan MCV. Dua kolom lainnya berupa data demografis, yaitu usia dan jenis kelamin. Kolom target bernama *source* merepresentasikan status rawat inap dengan nilai nol untuk rawat jalan dan satu untuk rawat inap.

Tabel 9.8 Ringkasan Struktur dan Distribusi Target Dataset Hematologi

Atribut	**Nilai**
Jumlah baris	4.412
Jumlah kolom awal	11
Kelas 0 (rawat jalan)	2.628 (59,6 %)
Kelas 1 (rawat inap)	1.784 (40,4 %)

Pembagian data	80 % latih (3.529) : 20 % uji (883), *stratified*

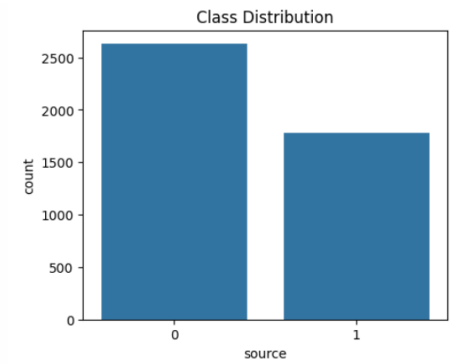
Alur kerja keseluruhan studi kasus ini ditunjukkan pada Gambar 9.9, dimulai dari pemuatan dan pembersihan data, eksplorasi melalui tiga sudut pandang, rekayasa fitur, hingga pembagian data untuk pemodelan.



Gambar 9.9 Alur Kerja Eksplorasi dan Penyiapan Data Hematologi Tabular

Eksplorasi Data

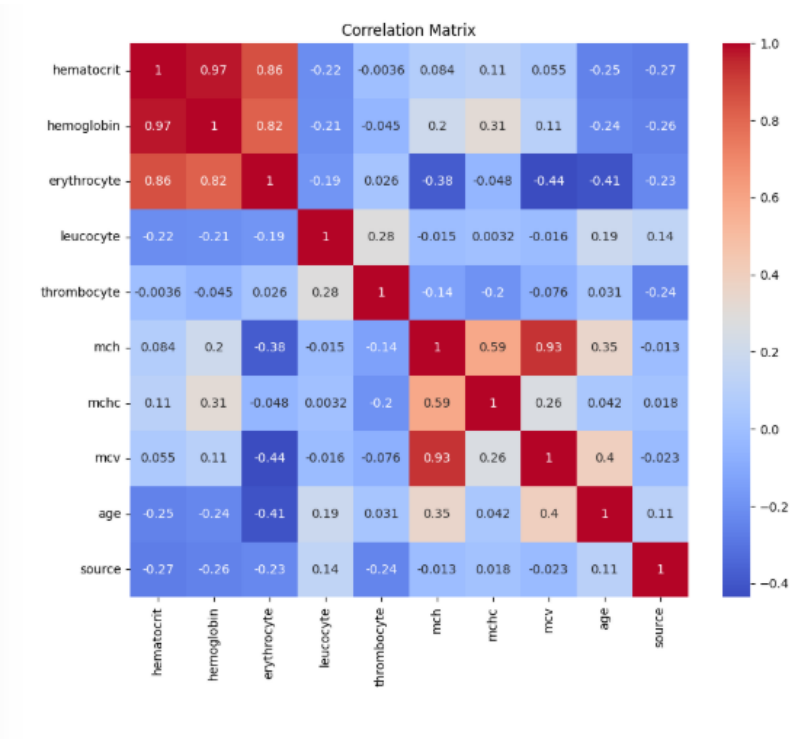
Eksplorasi data dilakukan melalui tiga sudut pandang yang saling melengkapi. Sudut pandang pertama adalah distribusi kelas target, yang ditunjukkan pada Gambar 9.10.



Gambar 9.10 Distribusi Kelas Status Rawat Inap pada Dataset Hematologi

Terlihat adanya ketidakseimbangan kelas yang tergolong moderat, di mana pasien rawat jalan sekitar satu setengah kali lebih banyak dibanding pasien rawat inap. Ketidakseimbangan sebesar ini belum tergolong ekstrem, namun tetap perlu diantisipasi pada tahap pemodelan agar model tidak terlalu condong pada kelas mayoritas.

Sudut pandang kedua adalah hubungan antar variabel laboratorium yang dianalisis menggunakan matriks korelasi sebagaimana ditunjukkan pada Gambar 9.11.

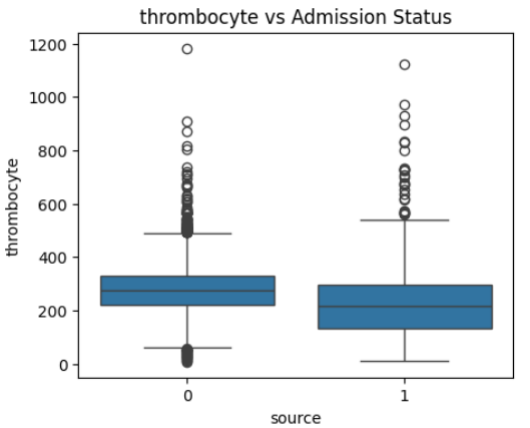


Gambar 9.11 Matriks Korelasi Antar Parameter Hematologi

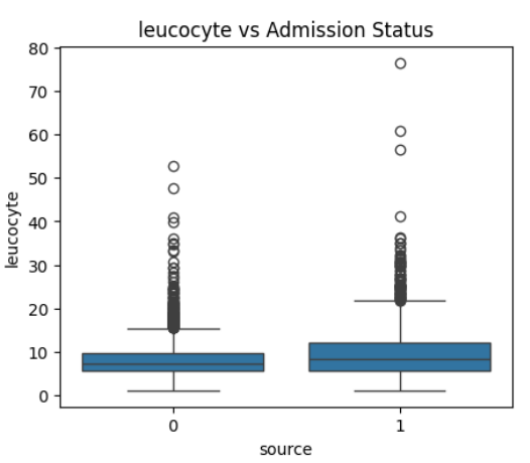
Berdasarkan Gambar 9.11, terlihat korelasi yang sangat kuat antara hematokrit dan hemoglobin dengan koefisien sebesar 0,97, serta antara MCH dan MCV dengan koefisien 0,93. Kedua pasangan tersebut menunjukkan adanya multikolinearitas yang sebenarnya wajar secara fisiologis, karena variabel-variabel tersebut sama-sama menggambarkan profil sel darah merah dari sudut yang berbeda. Terhadap variabel target, korelasi tertinggi meskipun tetap tergolong lemah ditunjukkan oleh hematokrit dengan koefisien negatif 0,27

dan hemoglobin dengan koefisien negatif 0,26. Arah negatif tersebut mengindikasikan bahwa pasien dengan hematokrit dan hemoglobin lebih rendah cenderung lebih banyak menjalani rawat inap, sebuah temuan yang masuk akal secara klinis.

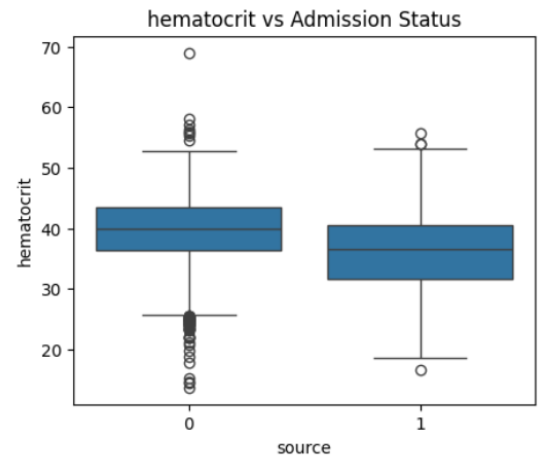
Sudut pandang ketiga adalah perbandingan distribusi empat fitur penting terhadap status rawat inap menggunakan diagram kotak, sebagaimana ditunjukkan pada Gambar 9.12 sampai Gambar 9.15.



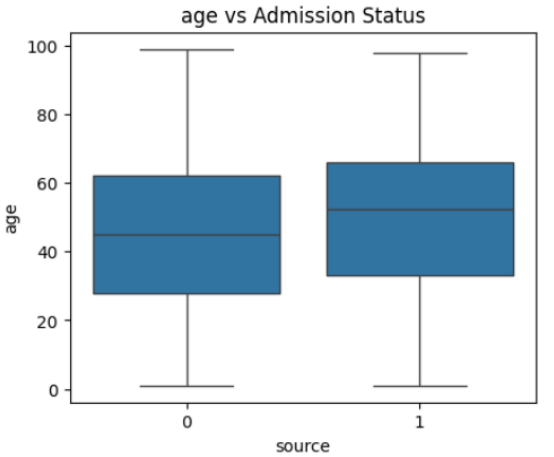
Gambar 9.12 Diagram Kotak Trombosit Terhadap Status Rawat Inap



Gambar 9.13 Diagram Kotak Leukosit Terhadap Status Rawat Inap



Gambar 9.14 Diagram Kotak Hematokrit Terhadap Status Rawat Inap



Gambar 9.15 Diagram Kotak Usia Terhadap Status Rawat Inap

Keempat diagram kotak tersebut menunjukkan pola yang konsisten dengan korelasi yang ditemukan sebelumnya. Median hematokrit pasien rawat inap sedikit lebih rendah dibanding pasien rawat jalan, median usia pasien rawat inap sedikit lebih tinggi, sementara leukosit pada kelompok rawat inap menunjukkan sebaran yang lebih lebar ke arah nilai tinggi. Sebaran leukosit yang melebar ke atas ini konsisten dengan kondisi klinis seperti infeksi atau peradangan yang umum mendasari keputusan rawat inap.

Rekayasa Fitur

Berdasarkan wawasan dari tahap eksplorasi, dilakukan rekayasa fitur untuk menciptakan variabel turunan yang menangkap interaksi antar parameter. Tiga fitur baru yang dihasilkan dirangkum pada Tabel 9.9.

Tabel 9.9 Fitur Hasil Rekayasa Fitur pada Data Hematologi

Fitur Baru	Formula	Wawasan yang Ditangkap
leu_throm_ratio	leukosit / (trombosit + 1)	Rasio sel darah putih terhadap trombosit sebagai indikator potensial peradangan
hb_hct_ratio	hemoglobin / (hematokrit + 1)	Rasio konsentrasi hemoglobin terhadap volume sel darah merah
platelet_age	trombosit × usia	Interaksi antara jumlah trombosit dan usia pasien

```
# ======================================
# REKAYASA FITUR (FEATURE ENGINEERING)
#======================================
df["leu_throm_ratio"] = df["leucocyte"]  / (df["thrombocyte"] + 1)
df["hb_hct_ratio"]    = df["hemoglobin"] / (df["hematocrit"]  + 1)
df["platelet_age"]    = df["thrombocyte"] * df["age"]

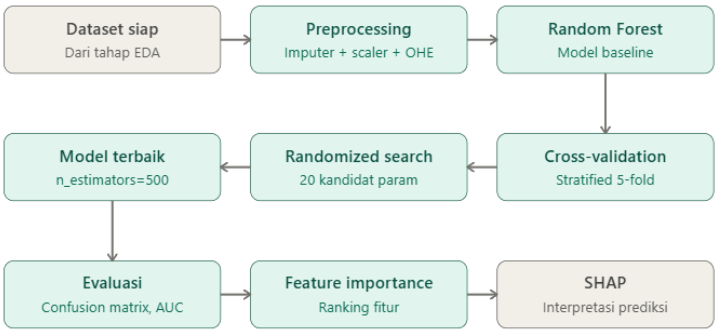
# ======================================
# TRAIN-TEST SPLIT
#
======================================
from sklearn.model_selection import train_test_split

X = df.drop("source", axis=1)
y = df["source"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, stratify=y, random_state=42
)
```

Pemodelan dengan *Random Forest*

Alur pemodelan pada studi kasus ini ditunjukkan pada Gambar 9.16, dimulai dari *pipeline* pra-pemrosesan, pelatihan model dasar, validasi silang, optimasi hiperparameter, evaluasi, hingga interpretasi.



Gambar 9.16 Alur Kerja Pemodelan Pengenalan Pola Status Rawat Inap

Algoritma *Random Forest* dipilih karena mampu menangkap hubungan non-linear antar parameter hematologi tanpa memerlukan asumsi distribusi tertentu, serta relatif tangguh terhadap keberadaan pencilan yang banyak dijumpai pada data laboratorium. *Pipeline* pra-pemrosesan menggabungkan imputasi nilai hilang, standardisasi variabel numerik, dan *one-hot encoding* untuk variabel kategorikal.

```
# ====================================
# PREPROCESSING PIPELINE + RANDOM FOREST
# ====================================
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import StratifiedKFold, cross_validate
```

```
numeric_transformer = Pipeline([
    ("imputer", SimpleImputer(strategy="median")),
    ("scaler",  StandardScaler())
])
categorical_transformer = Pipeline([
    ("imputer", SimpleImputer(strategy="most_frequent")),
    ("encoder", OneHotEncoder(drop="first"))
])

preprocessor = ColumnTransformer([
    ("num", numeric_transformer, numeric_features),
    ("cat", categorical_transformer, categorical_features)
])

rf = RandomForestClassifier(
    n_estimators=500, max_depth=20, min_samples_leaf=2,
    class_weight="balanced", random_state=42, n_jobs=-1
)

pipeline = Pipeline([("preprocessor", preprocessor),
                     ("classifier", rf)])

skf = StratifiedKFold(n_splits=5, shuffle=True,
random_state=42)
cv_results = cross_validate(pipeline, X_train, y_train,
                            cv=skf, scoring=scoring, n_jobs=-1)
```

Optimasi hiperparameter dilakukan menggunakan pencarian acak dengan dua puluh kandidat kombinasi parameter. Kombinasi terbaik yang ditemukan adalah jumlah pohon sebanyak 500, minimum sampel untuk melakukan pemisahan sebesar 5, minimum sampel pada setiap daun sebesar 1, skema jumlah fitur maksimum menggunakan log2, serta kedalaman maksimum pohon dibatasi hingga 10. Kombinasi ini menghasilkan skor F1 terbaik pada validasi silang sebesar 0,6723.

Pembatasan kedalaman pohon hingga sepuluh, meskipun tergolong dangkal untuk hutan dengan lima ratus estimator, justru menjadi indikasi bahwa pembatasan kompleksitas pohon individual membantu mencegah *overfitting* pada data dengan hubungan antar fitur yang kompleks namun jumlah sampel terbatas. Perbandingan kinerja model dasar dan model hasil optimasi disajikan pada Tabel 9.10.

Tabel 9.10 Perbandingan Model Baseline dan Model Hasil Tuning pada Validasi Silang Lima Lipatan

Metrik	**Baseline**	**Hasil Tuning**
Accuracy	0,7506 (± 0,0102)	0,7498 (± 0,0103)
Precision	0,7232 (± 0,0187)	0,7188 (± 0,0208)
Recall	0,6223 (± 0,0279)	0,6279 (± 0,0236)
F1-score	0,6684 (± 0,0164)	0,6698 (± 0,0135)
ROC AUC	0,7996 (± 0,0055)	0,8012 (± 0,0067)

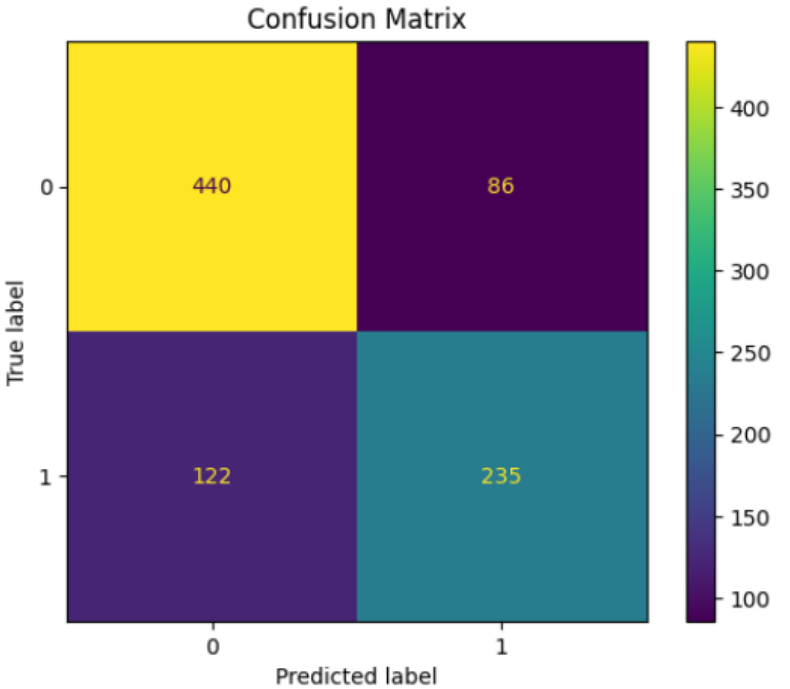
Terlihat bahwa peningkatan kinerja hasil optimasi tergolong sangat tipis. Hal ini merupakan informasi yang berharga: model dasar ternyata sudah mendekati batas optimal yang dapat dicapai pada konfigurasi fitur saat ini, sehingga perbaikan lebih lanjut kemungkinan besar harus datang dari sisi data dan fitur, bukan dari penyetelan hiperparameter.

Evaluasi pada Data Uji

Model hasil optimasi kemudian dievaluasi pada 883 sampel data uji yang belum pernah dilihat selama pelatihan, dengan hasil sebagaimana disajikan pada Tabel 9.11.

Tabel 9.11 Hasil Evaluasi Model pada Data Uji Sebanyak 883 Sampel

Kelas	***Precision***	***Recall***	**F1-score**	***Support***
0 (rawat jalan)	0,78	0,84	0,81	526
1 (rawat inap)	0,73	0,66	0,69	357
Akurasi keseluruhan	—	—	0,76	883
ROC AUC	—	—	0,8131	—



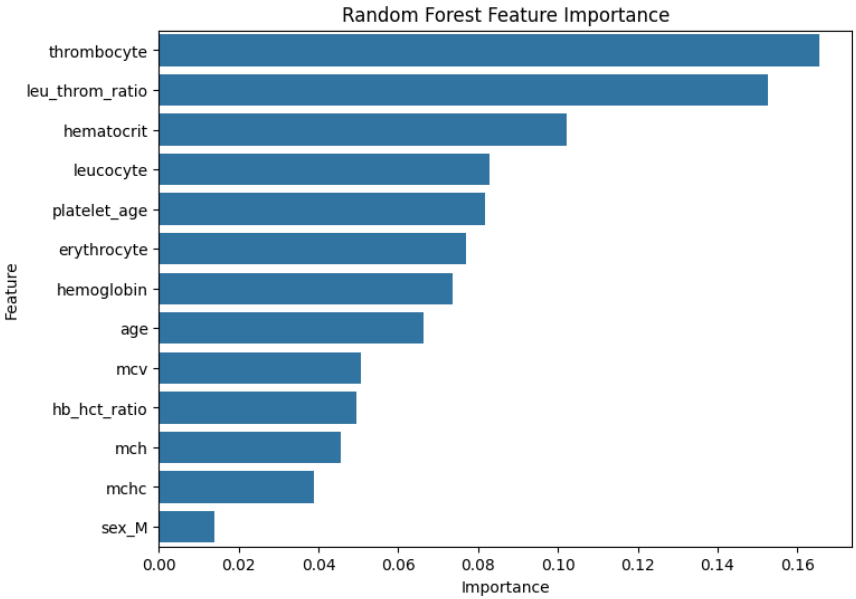
Gambar 9.17 *Confusion Matrix* Model Hasil Tuning pada Data Uji

Berdasarkan Gambar 9.17, model berhasil mengklasifikasikan 440 dari 526 pasien rawat jalan secara benar sebagai *true negative*, dan 235 dari 357 pasien rawat inap secara benar sebagai *true positive*. Kesalahan klasifikasi terbesar terjadi pada 122 pasien rawat inap yang justru diprediksi sebagai rawat jalan, yaitu kesalahan berjenis *false negative*.

Pola kesalahan tersebut menunjukkan bahwa model masih cenderung konservatif dalam mendeteksi kebutuhan rawat inap. Pada konteks klinis, hal ini merupakan area yang perlu diperbaiki, karena kesalahan *false negative* berisiko jauh lebih tinggi dibanding *false positive*. Pasien yang sebenarnya memerlukan rawat inap namun diprediksi cukup rawat jalan berpotensi mengalami perburukan kondisi, sedangkan pasien yang sebaliknya paling banter hanya mengalami pemeriksaan tambahan yang tidak perlu. Salah satu solusi yang dapat ditempuh adalah menurunkan ambang keputusan model agar lebih sensitif terhadap kelas rawat inap, dengan konsekuensi menurunnya *precision*.

Interpretasi Model

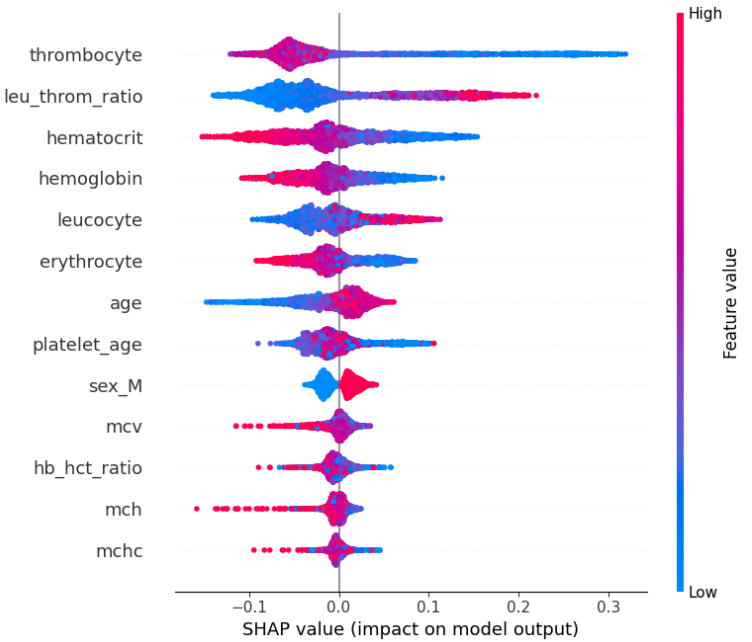
Untuk memahami kontribusi setiap fitur terhadap prediksi, dianalisis tingkat kepentingan fitur dari model *Random Forest* sebagaimana ditunjukkan pada Gambar 9.18.



Gambar 9.18 Tingkat Kepentingan Fitur pada Model *Random Forest*

Berdasarkan Gambar 9.18, fitur trombosit tercatat sebagai fitur paling penting, diikuti oleh *leu_throm_ratio* yang menempati peringkat kedua. *Ieu_throm_ratio* merupakan salah satu fitur hasil rekayasa fitur, bukan fitur asli dari dataset, namun tingkat kepentingannya lebih tinggi dibanding sebagian besar fitur aslinya. Hal ini membuktikan secara empiris bahwa rekayasa fitur yang dilakukan berdasarkan wawasan dari tahap eksplorasi berhasil menciptakan variabel yang secara nyata meningkatkan kemampuan model dalam mengenali pola.

Sebagai pelengkap, diterapkan pula metode SHAP untuk menjelaskan arah dan besar kontribusi tiap fitur terhadap prediksi individual, dengan hasil sebagaimana ditunjukkan pada Gambar 9.19.



Gambar 9.19 SHAP *Summary Plot* untuk Model Prediksi Status Rawat Inap

Berdasarkan Gambar 9.19, nilai trombosit yang rendah, yang ditandai dengan titik berwarna biru, cenderung mendorong prediksi ke arah rawat inap dengan nilai SHAP positif. Sebaliknya, nilai trombosit yang tinggi yang ditandai titik merah mendorong prediksi ke arah rawat jalan. Pola tersebut konsisten dengan korelasi negatif yang ditemukan pada tahap eksplorasi data. Pola serupa juga terlihat pada *leu_throm_ratio*, yang memperkuat bukti bahwa fitur hasil rekayasa tersebut memberikan sinyal yang jelas dan searah secara klinis, bukan sekadar kebetulan statistik.

Perbedaan penting antara tingkat kepentingan fitur dan SHAP perlu digarisbawahi. Tingkat kepentingan fitur hanya menunjukkan seberapa besar sebuah fitur berkontribusi terhadap keputusan model secara agregat, tanpa memberi tahu ke arah mana kontribusi tersebut mendorong. SHAP melengkapi kekurangan tersebut dengan menunjukkan arah kontribusi untuk setiap sampel individual, sehingga interpretasi menjadi jauh lebih kaya dan dapat dikaitkan dengan penalaran klinis.

➢ Keterbatasan Metodologi

Kejujuran dalam menyatakan keterbatasan merupakan bagian tak terpisahkan dari etos kerja ilmiah, terlebih pada sistem yang menyentuh ranah kesehatan manusia. Berikut ini diuraikan keterbatasan yang melekat pada sistem yang dibahas dalam bab ini.

a. Jumlah subjek apusan sangat sedikit. Hanya lima subjek yang digunakan, terdiri atas empat pasien talasemia dan satu individu sehat. Jumlah sekecil ini tidak memungkinkan penarikan kesimpulan statistik mengenai sensitivitas maupun spesifisitas sistem pada tingkat populasi.
b. Tidak tersedia label kebenaran per sel pada citra apusan utuh. Akibatnya, proporsi yang dilaporkan sepenuhnya bersifat estimasi berdasarkan prediksi model, dan tidak dapat diverifikasi terhadap anotasi pakar.
c. Aturan ambang risiko belum tervalidasi secara klinis. Nilai sepuluh persen, lima belas persen, dan lima persen bersifat demonstratif dan memerlukan kalibrasi bersama pakar hematologi menggunakan data yang jauh lebih besar.
d. Jumlah sampel sangat kecil pada kelas pencil. Dengan hanya dua sampel pada data uji, metrik untuk kelas tersebut tidak dapat diandalkan dan tidak memberikan informasi statistik yang bermakna.
e. Peta Grad-CAM relatif kasar. Keterbatasan resolusi masukan sebesar 96 kali 96 piksel membuat peta panas yang dihasilkan memiliki granularitas yang terbatas, sehingga hanya mampu menunjukkan area perhatian secara umum, bukan detail struktur halus.
f. Proses segmentasi masih dapat melewatkan sebagian sel. Sel yang terlalu banyak bertumpuk atau berada di tepi citra sengaja disingkirkan, sehingga proporsi yang dihitung didasarkan pada subset sel, bukan seluruh sel pada apusan.
g. Variasi pewarnaan dan peralatan belum diuji. Seluruh citra berasal dari satu protokol akuisisi dengan satu jenis mikroskop dan kamera. Kinerja sistem pada citra dari laboratorium lain dengan protokol berbeda belum diketahui.

h. Sistem tidak menggantikan pemeriksaan baku emas. Diagnosis pasti talasemia tetap memerlukan HPLC dan analisis DNA. Sistem ini hanya berfungsi sebagai penyaring awal untuk memprioritaskan kasus yang perlu ditindaklanjuti.

➢ Arah Pengembangan Lanjutan

Berdasarkan keterbatasan di atas, sejumlah arah pengembangan dapat ditempuh pada Metodologi selanjutnya.

a. Memperluas jumlah subjek apusan serta melibatkan lebih dari satu laboratorium agar variasi protokol akuisisi dan pewarnaan ikut terwakili dalam data latih.
b. Melakukan anotasi tingkat sel pada citra apusan utuh oleh tenaga laboratorium klinis, sehingga proporsi yang dihasilkan sistem dapat divalidasi secara langsung.
c. Mengkalibrasi ambang batas risiko menggunakan data berlabel diagnosis pasti, misalnya dengan mencari kombinasi ambang yang memaksimalkan sensitivitas pada tingkat spesifisitas yang dapat diterima secara klinis.
d. Mengganti pendekatan segmentasi klasik dengan segmentasi berbasis pembelajaran mendalam seperti U-Net atau Mask R-CNN, yang berpotensi lebih tangguh terhadap sel yang saling bertumpuk.
e. Menaikkan resolusi masukan model agar peta Grad-CAM menjadi lebih detail dan kelas yang saling mirip seperti stomatosit dan sel target dapat dibedakan dengan lebih baik.
f. Mengeksplorasi arsitektur *ensemble* lain seperti XGBoost atau LightGBM pada studi kasus tabular, serta menyesuaikan ambang keputusan agar model lebih sensitif terhadap kelas rawat inap.

➢ Rangkuman

Bab ini telah memaparkan perancangan dan pembangunan *Thalassemia Morphology Screener*, sebuah sistem peringatan dini talasemia berbasis analisis proporsi morfologi sel darah merah pada citra apusan darah tepi. Ringkasan komponen teknis sistem dari hulu ke hilir disajikan pada Tabel 9.12 sebagai referensi cepat.

Tabel 9.12 Ringkasan Komponen Teknis Sistem TMS dari Hulu ke Hilir

Komponen	**Konfigurasi / Pilihan**
Sumber data	RBCdataset (7.108 citra, 9 kelas) + ThalassemiaPBS (108 citra, 5 subjek)
Ukuran masukan model	96 × 96 piksel, tiga kanal, resize_with_pad
Augmentasi	RandomFlip horizontal dan vertikal, RandomRotation 0,3
Pembagian data	Group-aware split 70 % / 15 % / 15 %, kebocoran grup = 0
Arsitektur	*MobileNetV2* pra-latih *ImageNet* + GlobalAveragePooling2D + *Dropout* 0,3 + Dense 9 softmax
Tahap 1 (*head warm-up*)	Backbone dibekukan, Adam 1×10^{-3}, maks 20 epoch, patience 6, cw_balanced
Tahap 2 (*fine-tuning*)	30 lapisan akhir dicairkan, Adam 1×10^{-5}, maks 30 epoch, patience 8, cw_lembut
Hasil akhir	Akurasi uji 94,8 %, *recall* kelas normal ± 94 %
Interpretabilitas	Grad-CAM per kelas, fokus pada kontur dan tubuh sel
Segmentasi apusan	Resize 800 × 600 → green channel → Otsu → morfologi → *watershed*
Indikator risiko	Ambang demonstratif: target > 10 %, hipokromik > 15 %, teardrop > 5 %; 2 penanda = MERAH, 1 = KUNING, 0 = HIJAU
Deployment	Aplikasi *Streamlit* (app.py), model dan *pipeline* dipakai ulang dari notebook

Tiga pelajaran utama dapat ditarik dari keseluruhan pembahasan bab ini. Pertama, pemahaman terhadap asal-usul data jauh lebih menentukan daripada kecanggihan arsitektur model. Kegagalan pada percobaan awal penerapan apusan bukan disebabkan oleh model yang buruk, melainkan oleh perbedaan *pipeline* pra-pemrosesan yang tampak sepele namun berakibat fatal.

Kedua, kesabaran dalam pelatihan sering kali lebih berharga daripada menambah kompleksitas. Peningkatan akurasi dari 91 persen menjadi 95 persen dicapai bukan dengan mengganti arsitektur atau menambah lapisan, melainkan hanya dengan memperbesar nilai *patience* dan menjaga laju pembelajaran tetap stabil.

Ketiga, akurasi tinggi tidak cukup untuk menjadikan sebuah sistem layak dipercaya pada ranah kesehatan. Verifikasi penalaran model melalui *Grad*-CAM, pelaporan metrik per kelas alih-alih hanya akurasi agregat, serta pernyataan keterbatasan yang jujur, sama pentingnya dengan angka kinerja itu sendiri. Sebuah sistem kecerdasan buatan di bidang kesehatan pada akhirnya dinilai bukan hanya dari seberapa sering ia benar, tetapi juga dari seberapa jelas ia dapat menjelaskan mengapa ia berpendapat demikian dan seberapa jujur ia menyatakan batas kemampuannya.

➢ Soal Latihan

a. Jelaskan mengapa sel target dan sel hipokromik dianggap sebagai penanda khas pada apusan darah penderita talasemia. Kaitkan jawaban Anda dengan mekanisme gangguan sintesis rantai globin.
b. Sebuah citra apusan menghasilkan 120 sel tersegmentasi dengan komposisi 14 sel target, 20 sel hipokromik, 5 sel teardrop, dan sisanya kelas lain. Hitung proporsi ketiga penanda tersebut dan tentukan indikator risiko yang dihasilkan sistem beserta alasannya.
c. Jelaskan apa yang dimaksud dengan kebocoran data pada konteks pembagian dataset citra sel, dan mengapa teknik *group-aware split* dapat mencegahnya. Berikan contoh skenario konkret di mana pembagian acak biasa akan menghasilkan akurasi yang menyesatkan.
d. Uraikan alasan teknis mengapa pada tahap *fine-tuning* hanya tiga puluh lapisan terakhir tulang punggung yang dicairkan, bukan seluruhnya. Apa risiko yang timbul apabila seluruh lapisan dicairkan sekaligus?
e. Dengan menggunakan formula bobot kelas seimbang, hitunglah bobot untuk sebuah kelas yang memiliki 40 sampel pada dataset berisi 5.000 sampel yang terbagi dalam 10 kelas. Hitung pula bobot lembutnya.

f. Jelaskan perbedaan tujuan antara evaluasi menggunakan *confusion matrix* dan verifikasi menggunakan Grad-CAM. Mengapa keduanya tidak dapat saling menggantikan?
g. Sebuah model klasifikasi memperoleh akurasi keseluruhan 96 persen pada dataset dengan sembilan kelas, namun *recall* pada dua kelas minoritas hanya 30 persen. Jelaskan mengapa laporan akurasi tersebut berpotensi menyesatkan dan metrik apa yang sebaiknya dilaporkan sebagai gantinya.
h. Rancanglah secara garis besar sebuah eksperimen untuk menguji ketahanan sistem TMS terhadap variasi pewarnaan antar laboratorium. Sebutkan data yang diperlukan, variabel yang dikendalikan, serta metrik yang digunakan untuk menilai ketahanan tersebut.
i. Pada studi kasus prediksi status rawat inap, jelaskan mengapa kesalahan *false negative* dianggap lebih berisiko dibanding *false positive*, dan sebutkan satu strategi teknis untuk menekan angka *false negative*.
j. Diskusikan secara kritis: apakah sebuah sistem dengan akurasi 94,8 persen sudah layak digunakan untuk skrining massal talasemia di puskesmas? Sertakan pertimbangan teknis, klinis, dan etis dalam jawaban Anda.

➢ Daftar Pustaka

Bain, B. J. (2021). Blood cells: A practical guide (5th ed.). Wiley-Blackwell.

Barbalato, L., & Pillarisetty, L. S. (2022). Histology, red blood cell. In StatPearls. StatPearls Publishing.

Goh, L. P. W., Chong, E. T. J., & Lee, P.-C. (2020). Prevalence of alpha(α)-thalassemia in Southeast Asia (2010–2020): A meta-analysis involving 83,674 subjects. International Journal of Environmental Research and Public Health, 17(20), 7354.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263–1284.

Kattamis, A., Kwiatkowski, J. L., & Aydinok, Y. (2022). Thalassaemia. The Lancet, 399(10343), 2310–2324.

Kementerian Kesehatan Republik Indonesia. (2022). Pedoman pencegahan dan pengelolaan thalassemia di Indonesia. Kemenkes RI.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.

Needs, T., Gonzalez-Mosquera, L. F., & Lynch, D. T. (2023). Beta thalassemia. StatPearls Publishing.

Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1), 62–66.

Palmer, L., Briggs, C., McFadden, S., Zini, G., Burthem, J., Rozenberg, G., Proytcheva, M., & Machin, S. J. (2015). ICSH recommendations for the standardization of nomenclature and grading of peripheral blood cell morphological features. International Journal of Laboratory Hematology, 37(3), 287–303.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345–1359.

Powers, D. M. W. (2011). Evaluation: From *precision*, *recall* and F-measure to ROC, informedness, markedness and correlation. Journal of Machine Learning Technologies, 2(1), 37–63.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 4510–4520). IEEE.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) (pp. 618–626). IEEE.

Streamlit Inc. (2023). *Streamlit* documentation. https://docs.*streamlit*.io

Taher, A. T., Musallam, K. M., & Cappellini, M. D. (2021). β-Thalassemias. New England Journal of Medicine, 384(8), 727–743.

Tyas, D. A., Ratnaningsih, T., Harjoko, A., & Hartati, S. (2022). Erythrocyte (red blood cell) dataset in thalassemia case. Data in Brief, 41, 107886.

Wang, F., Wang, G., Yang, Z., Wang, X., Liu, D., Wan, N., & Wu, W. (2021). Differential diagnosis of thalassemia and iron deficiency anemia using the CellaVision Advanced Red Blood Cell software. International Journal of Laboratory Hematology, 43(4), 853–858.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. Journal of Big Data, 3(1), 1–40.

BAB 10
KLASIFIKASI PENYAKIT PARU-PARU PADA CITRA *CHEST X-RAY* MENGGUNAKAN *FINE TUNING VGG16* DAN *EXPLAINABLE ARTIFICIAL INTELLIGENCE*

Setelah mempelajari bab ini, pembaca diharapkan mampu:

a. Menjelaskan penanda radiologis lima kondisi paru-paru pada citra *chest X-ray* serta memperkirakan kelas mana yang paling sulit dikenali beserta alasannya.
b. Menguraikan cara kerja *Convolutional Neural Network* dan arsitektur VGG16, mulai dari operasi konvolusi hingga perhitungan jumlah parameter setiap blok.
c. Menjelaskan prinsip *transfer learning* serta alasan pembekuan *convolutional base* pada perbandingan dua rancangan lapisan klasifikasi.
d. Menguraikan peran *dropout*, *batch normalization*, dan *Global Average Pooling* sebagai penyeimbang terhadap penambahan kapasitas model.
e. Membaca *confusion matrix* multikelas sampai ke arah kesalahannya, bukan sekadar jumlah totalnya.
f. Mengenali keterbatasan metrik akurasi dalam menyembunyikan kegagalan pada satu kelas tertentu.
g. Menafsirkan hasil Grad-CAM, LIME, dan *Occlusion Sensitivity* secara kritis, termasuk mengenali pertanda *shortcut learning*.
h. Menjelaskan gejala *overconfidence* pada model klasifikasi serta dampaknya terhadap rancangan sistem yang menggunakan ambang keyakinan.

➢ Pendahuluan

Penyakit infeksi saluran pernapasan bagian bawah masih menempati posisi teratas sebagai penyebab kematian akibat penyakit menular di dunia. *Pneumonia* bakteri, *pneumonia* virus, tuberkulosis, dan *Corona Virus Disease* merupakan empat kondisi yang menyerang parenkim paru dengan tingkat penyebaran tinggi serta berpotensi menimbulkan komplikasi berat apabila penanganannya terlambat. Di Indonesia, sebaran fasilitas kesehatan yang belum merata

memperbesar risiko keterlambatan tersebut, terutama karena jumlah dokter spesialis radiologi masih terbatas dibandingkan jumlah penduduk yang harus dilayani.

Diagnosis keempat kondisi tersebut umumnya diawali dengan pemeriksaan radiografi toraks atau *chest X-ray*. Pemeriksaan ini dipilih karena murah, cepat, tersedia luas, dan memiliki dosis radiasi jauh lebih rendah dibandingkan *Computed Tomography*. Melalui citra *chest X-ray*, tenaga medis dapat mengamati konsolidasi, infiltrat, kavitas, maupun pola opasitas tertentu yang menjadi penanda suatu penyakit. Persoalannya, pembacaan citra semacam ini menuntut keahlian khusus, memerlukan waktu, dan hasilnya dipengaruhi perbedaan persepsi antar-pembaca maupun kelelahan tenaga medis.

Bab ini membahas pembangunan model klasifikasi lima kelas kondisi paru-paru pada citra *chest X-ray* menggunakan arsitektur *VGG16* dengan pendekatan transfer learning. Dua model dibandingkan secara berdampingan: model baseline dengan lapisan klasifikasi minimal, dan model modifikasi yang lapisan klasifikasinya dirancang ulang melalui penambahan dense layer, *batch normalization*, dan *dropout*. Seluruh faktor lain dataset, pembagian data, prapemrosesan, augmentasi, dan *hyperparameter* ditetapkan identik agar perbedaan yang teramati dapat diatribusikan pada satu variabel saja.

Perbedaan mendasar bab ini dengan pembahasan yang lazim ditemukan pada laporan sejenis terletak pada apa yang dianggap sebagai hasil. Modifikasi lapisan klasifikasi memang menaikkan akurasi dari 0,808 menjadi 0,858, dan pada kebanyakan laporan angka tersebut sudah dianggap cukup sebagai kesimpulan. Bab ini tidak berhenti di situ. Angka 85,75 persen akan dibongkar menjadi bagian-bagian penyusunnya, dan pembongkaran itu memunculkan tiga temuan yang tidak terlihat pada metrik agregat.

Pertama, kenaikan akurasi tersebut tidak terdistribusi merata. Satu kelas Viral *Pneumonia* tetap gagal dikenali pada hampir separuh kasusnya, dengan *recall* hanya 0,5556 setelah modifikasi. Kedua, model menghasilkan tingkat keyakinan antara 99,5 hingga 100 persen pada prediksi yang salah, yaitu setara dengan tingkat keyakinannya pada prediksi yang benar. Ketiga, visualisasi *Grad-*

CAM memperlihatkan aktivasi pada area di luar struktur anatomi paru pada sebagian sampel.

Ketiga hal tersebut tidak muncul dari metrik akurasi dan tidak akan pernah muncul apabila evaluasi berhenti pada satu angka tunggal. Menelusuri ketiganya jauh lebih bermanfaat daripada sekadar melaporkan bahwa modifikasi arsitektur berhasil menaikkan performa. Penelusuran itulah yang menjadi inti pelajaran pada bab ini.

➢ Pengantar Teori

Bagian ini menguraikan landasan teori yang diperlukan pada bab ini, mencakup karakteristik radiologis kelima kelas yang diklasifikasikan, mekanisme kerja *Convolutional Neural Network* beserta arsitektur VGG16, prinsip transfer learning, teknik regularisasi, metrik evaluasi, serta metode interpretasi model.

Lima Kondisi Paru-Paru dan Penandanya pada Citra

Citra *chest X-ray* dihasilkan melalui pemaparan sinar-X yang menembus rongga dada. Jaringan berdensitas tinggi seperti tulang menyerap lebih banyak radiasi sehingga tampak terang, sedangkan udara di dalam alveolus menyerap sedikit radiasi sehingga tampak gelap. Perbedaan derajat penyerapan inilah yang membentuk kontras pada citra dan memungkinkan identifikasi kelainan pada parenkim paru.

Kelima kondisi yang diklasifikasikan pada bab ini beserta penanda radiologisnya diuraikan pada Tabel 10.1.

Tabel 10.1 Lima Kondisi Paru-Paru dan Penanda Radiologisnya

Kondisi	**Penyebab**	**Pola Radiologis Khas**	**Distribusi**
Bacterial *Pneumonia*	Bakteri, tersering *Streptococcus pneumonia*e	Konsolidasi lobaris padat disertai air bronchogram	Fokal, batas relatif tegas
Viral *Pneumonia*	Influenza, RSV, *adenovirus*	Infiltrat interstisial halus	Difus bilateral, batas tidak tegas
Tuberculosis	*Mycobacterium tuberculosis*	Infiltrat, kavitas, nodul, fibrosis, kalsifikasi	Dominan lobus atas
Corona Virus Disease	SARS-CoV-2	Opasitas *ground-glass*, dapat berlanjut *crazy paving*	Bilateral perifer, zona bawah
Normal	—	Lapangan paru radiolusen, sudut kostofrenikus tajam	Simetris tanpa temuan

Catatan Klinis: Mengapa Viral *Pneumonia* Paling Sulit Dikenali

Kelima kondisi tersebut tidak setara tingkat kesulitannya, dan hal ini akan menjadi sangat penting ketika membaca hasil pada Subbab 10.4.6 dan 10.4.7.

Tuberculosis memiliki penanda spasial yang konsisten berupa kecenderungan lesi berada pada apeks paru. Bacterial *Pneumonia* menghasilkan konsolidasi padat dengan batas yang cukup tegas. *Corona Virus Disease* menampilkan opasitas ground-glass dengan distribusi perifer yang khas. Ketiganya menyediakan pola yang relatif mudah dipelajari oleh model.

Viral *Pneumonia* berbeda. Perubahan yang ditimbulkannya berupa infiltrat interstisial yang halus, menyebar, dan tidak memiliki batas tegas. Pada tahap awal, pola tersebut dapat menyerupai paru normal; pada tahap lanjut, dapat menyerupai *pneumonia* bakteri. Dengan kata lain, kelas ini berada di antara dua kelas lain yang keduanya memiliki penanda lebih kuat.

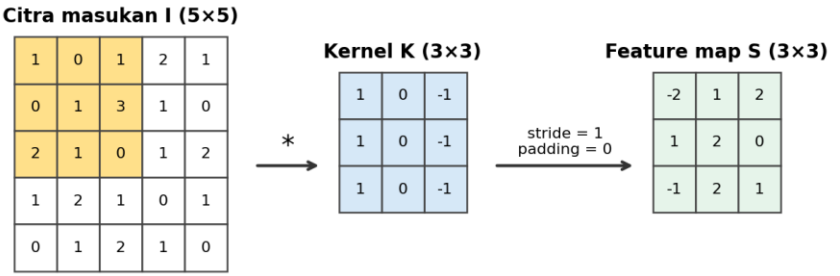
Konsekuensinya dapat diperkirakan sejak awal: apabila model kesulitan, kesulitan itu akan muncul pada Viral *Pneumonia*, dan kesalahannya akan mengarah pada Normal atau Bacterial *Pneumonia*. Perkiraan ini nanti akan terkonfirmasi oleh *confusion matrix*.

Convolutional Neural Network (CNN) merupakan arsitektur deep learning yang dirancang untuk memproses data berstruktur grid seperti citra digital. CNN memanfaatkan tiga gagasan utama, yaitu konektivitas lokal, pembagian bobot, dan invariansi translasi melalui operasi pooling. Ketiganya menyebabkan jumlah parameter CNN jauh lebih sedikit dibandingkan jaringan *fully connected* dengan ukuran masukan yang sama.

Operasi inti CNN adalah konvolusi diskrit dua dimensi antara citra masukan dan sekumpulan kernel yang bobotnya dipelajari selama pelatihan.

$$S(i,j) = (I * K)(i,j) = \sum_{m=0}^{F-1} \sum_{n=0}^{F-1} I(i+m, j+n) \cdot K(m,n) \quad (10.1)$$

Kernel digeser secara sistematis di atas citra masukan; pada setiap posisi dilakukan perkalian elemen demi elemen antara kernel dan area yang tertutup kernel, kemudian hasilnya dijumlahkan menjadi satu nilai pada *feature* map keluaran. Karena kernel yang sama digunakan untuk seluruh posisi, model hanya perlu mempelajari satu himpunan bobot untuk mendeteksi satu jenis pola pada seluruh bagian citra. Ilustrasi operasi tersebut disajikan pada Gambar 10.1.



Gambar 10.1 Ilustrasi Operasi Konvolusi pada Citra Digital

Ukuran *feature* map keluaran ditentukan oleh ukuran masukan, ukuran kernel, padding, dan stride.

$$W_{out} = \left\lfloor \frac{W_{in} - F + 2P}{S} \right\rfloor + 1 \quad (10.2)$$

Pada *VGG16* digunakan kernel 3×3 dengan padding 1 dan stride 1, sehingga substitusi ke dalam persamaan tersebut menghasilkan ukuran keluaran yang sama dengan ukuran masukan. Reduksi dimensi spasial pada *VGG16* karena itu sepenuhnya dikerjakan oleh lapisan pooling, bukan oleh lapisan konvolusi.

Setiap lapisan konvolusi diikuti fungsi aktivasi yang memberikan sifat non-linear pada model. Tanpa fungsi aktivasi non-linear, susunan lapisan yang dalam akan runtuh menjadi satu transformasi linear tunggal. Fungsi yang digunakan pada *VGG16* adalah Rectified Linear Unit.

$$f(x) = \mathrm{ReLU}(x) = \max(0, x) \quad \Rightarrow \quad f(x) = x \text{ jika } x > 0;\ f(x) = 0 \text{ jika } x \le 0 \tag{10.3}$$

Lapisan pooling mereduksi dimensi spasial *feature* map sehingga komputasi berkurang dan model menjadi lebih tahan terhadap pergeseran kecil pada citra. *VGG16* menggunakan max pooling berukuran 2×2 dengan stride 2.

$$y_{i,j} = \max_{0 \le m,n < F} x_{i \cdot S+m,\ j \cdot S+n} \tag{10.4}$$

Pada tahap akhir, lapisan keluaran menggunakan fungsi softmax yang mengubah skor logit menjadi distribusi probabilitas atas seluruh kelas, sehingga keluaran model dapat ditafsirkan sebagai probabilitas keanggotaan kelas.

$$\hat{y}_k = P(y = k \mid x) = \frac{\exp(z_k)}{\sum_{j=1}^{K} \exp(z_j)}, \qquad \sum_{k=1}^{K} \hat{y}_k = 1 \tag{10.5}$$

Fungsi kerugian yang bersesuaian dengan softmax pada klasifikasi multikelas adalah *categorical crossentropy*. Karena label berbentuk one-hot, nilai kerugiannya setara dengan negatif logaritma probabilitas yang diberikan model terhadap kelas yang benar.

$$\mathcal{L}_{CCE} = -\sum_{k=1}^{K} y_k \log(\hat{y}_k) \tag{10.6}$$

Catatan: *Categorical Crossentropy* Mendorong Keyakinan Ekstrem

Bentuk negatif logaritma pada Persamaan 10.6 memiliki konsekuensi yang jarang disadari dan baru akan terasa pada Subbab 10.4.9.

Fungsi kerugian tersebut tidak pernah mencapai nol kecuali probabilitas terhadap kelas yang benar bernilai tepat satu. Selama pelatihan berlangsung, model karena itu terus memperoleh dorongan untuk menaikkan probabilitas kelas yang benar mendekati satu dan menekan probabilitas kelas lain mendekati nol, bahkan ketika kelas tersebut sudah diprediksi dengan benar.

Akibatnya, keluaran softmax pada model yang dilatih dengan fungsi kerugian ini cenderung berbentuk ekstrem mendekati nol atau mendekati satu dan tidak lagi mencerminkan tingkat keandalan prediksi yang sesungguhnya. Model yang benar pada 86 persen kasus dapat menghasilkan keyakinan rata-rata jauh di atas 86 persen.

Pembaruan parameter dilakukan melalui algoritma *backpropagation* menggunakan *optimizer* Adam, yang memelihara estimasi rata-rata bergerak dari momen pertama dan momen kedua gradien.

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{10.7}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{10.8}$$

Karena kedua estimasi tersebut diinisialisasi bernilai nol, terjadi bias menuju nol pada iterasi awal sehingga diperlukan koreksi sebelum parameter diperbarui.

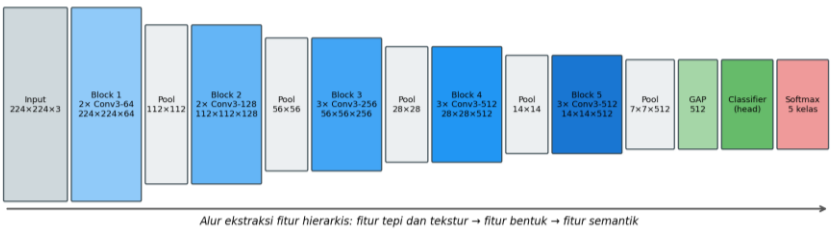
$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \qquad \hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{10.9}$$

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} \tag{10.10}$$

Arsitektur *VGG16*

VGG16 dikembangkan oleh Visual Geometry Group dari University of Oxford dan diperkenalkan oleh Simonyan dan Zisserman pada kompetisi *ImageNet Large Scale Visual Recognition Challenge* tahun 2014. Namanya mengacu pada jumlah lapisan berbobot yang dimilikinya, yaitu 13 lapisan konvolusi dan 3 lapisan *fully connected*.

Gagasan utamanya adalah penggunaan kernel berukuran kecil 3×3 yang disusun bertumpuk sebagai pengganti kernel besar. Dua lapisan konvolusi 3×3 berurutan memiliki medan reseptif efektif setara satu lapisan 5×5, sedangkan tiga lapisan berurutan setara kernel 7×7. Susunan bertumpuk tersebut menghasilkan jumlah parameter yang lebih sedikit sekaligus menambah fungsi non-linear pada setiap lapisan. Struktur lengkapnya disajikan pada Gambar 10.2.



Gambar 10.2 Struktur *Convolutional Base VGG16*

Berdasarkan Gambar 10.2, dimensi spasial berkurang setengahnya setelah setiap lapisan *pooling* sementara kedalaman kanal meningkat, sehingga citra masukan 224×224 tereduksi berturut-turut menjadi 112×112, 56×56, 28×28, 14×14, dan akhirnya 7×7 dengan 512 kanal. Pola ini mencerminkan pertukaran antara resolusi spasial dan kekayaan representasi semantik.

Jumlah parameter setiap lapisan konvolusi dihitung dari perkalian ukuran kernel, jumlah kanal masukan, dan jumlah filter, ditambah satu bias untuk setiap filter. Pada lapisan pertama dengan kernel 3×3, tiga kanal masukan, dan 64 filter, diperoleh 1.792 parameter. Perhitungan serupa pada seluruh lapisan menghasilkan total 14.714.688 parameter untuk keseluruhan *convolutional base*, sebagaimana dirinci pada Tabel 10.2.

Tabel 10.2 Struktur dan Jumlah Parameter *Convolutional Base VGG16*

Blok	**Susunan Lapisan**	**Dimensi Keluaran**	**Jumlah Parameter**
Input	—	224×224×3	0
Blok 1	2 × Conv 3×3, 64 filter	224×224×64	38.720
Blok 2	2 × Conv 3×3, 128 filter	112×112×128	221.440
Blok 3	3 × Conv 3×3, 256 filter	56×56×256	1.475.328
Blok 4	3 × Conv 3×3, 512 filter	28×28×512	5.899.776
Blok 5	3 × Conv 3×3, 512 filter	14×14×512	7.079.424
Keluaran	Max *pooling* 2×2	7×7×512	0
Total	13 lapisan konvolusi	—	14.714.688

Perlu dicatat bahwa angka tersebut hanya mencakup *convolutional base*. Apabila lapisan *fully connected* bawaan turut disertakan, total parameter mencapai sekitar 138 juta, dengan sebagian besar berasal dari lapisan dense pertama yang menghubungkan 25.088 neuron dengan 4.096 neuron. Alasan efisiensi inilah yang mendasari penggunaan konfigurasi tanpa lapisan *fully connected* bawaan pada bab ini.

Transfer Learning dan *Fine Tuning*

Transfer learning memanfaatkan pengetahuan yang telah dipelajari model pada suatu domain sumber untuk diterapkan pada domain target yang berbeda namun berkaitan. Pada bab ini, domain sumber adalah *ImageNet* yang berisi lebih dari satu juta citra objek sehari-hari, sedangkan domain targetnya adalah citra *chest X-ray*.

Dasar pemikirannya adalah bahwa lapisan awal CNN mempelajari fitur yang sangat umum berupa tepi, sudut, gradien intensitas, dan tekstur dasar. Fitur semacam itu relevan pada hampir semua jenis citra, termasuk citra medis. Lapisan yang lebih dalam mempelajari fitur yang lebih spesifik terhadap domain sumbernya.

Dengan mempertahankan lapisan awal dan menyesuaikan lapisan akhir, model dapat mencapai performa baik meskipun data pada domain target terbatas.

Terdapat tiga strategi yang lazim digunakan. *Feature extraction* membekukan seluruh *convolutional base* dan hanya melatih classifier baru. *Fine tuning* parsial membekukan sebagian blok awal dan melatih blok akhir bersama classifier. Full *fine tuning* melatih ulang seluruh lapisan dengan learning rate yang sangat kecil. Bab ini menggunakan strategi pertama pada *convolutional base*, dikombinasikan dengan perancangan ulang lapisan klasifikasi.

Catatan: Mengapa *Convolutional Base* Dibekukan Seluruhnya

Keputusan membekukan seluruh *convolutional base* memiliki tiga alasan yang saling menguatkan.

a. Jumlah data pelatihan sebanyak 6.455 citra tergolong sedang. Melatih ulang 14,7 juta parameter pada data sebanyak itu berisiko besar menimbulkan *overfitting*.
b. Pembekuan mengurangi kebutuhan komputasi secara drastis, karena hanya sekitar 2,6 persen dari total parameter yang perlu diperbarui pada model modifikasi.
c. Yang terpenting untuk keperluan bab ini: pembekuan memastikan kedua model bekerja pada representasi fitur yang persis sama, sehingga perbedaan performa yang teramati murni berasal dari rancangan classifier.

Alasan ketiga inilah yang menjadikan perbandingan pada bab ini bermakna. Apabila *convolutional base* ikut dilatih, perbedaan hasil akan bercampur antara pengaruh classifier dan pengaruh adaptasi fitur, dan tidak lagi mungkin memisahkan keduanya.

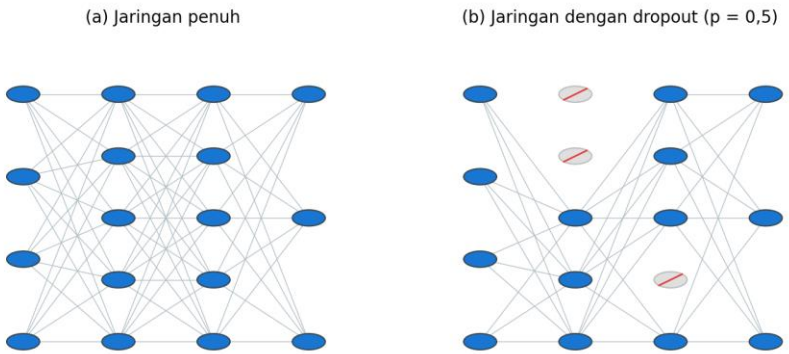
Teknik Regularisasi pada Lapisan Klasifikasi

Regularisasi merupakan sekumpulan teknik yang menekan risiko *overfitting*, yaitu kondisi ketika model mempelajari data pelatihan secara berlebihan termasuk derau yang terkandung di dalamnya. Tiga teknik digunakan pada rancangan classifier di bab ini.

Dropout menonaktifkan sebagian neuron secara acak pada setiap iterasi pelatihan, dengan setiap neuron dipertahankan menurut distribusi Bernoulli.

$$r_j \sim \text{Bernoulli}(1-p), \qquad \tilde{a}_j = r_j \cdot a_j \tag{10.11}$$

Dengan mekanisme tersebut model tidak dapat bergantung berlebihan pada neuron tertentu sehingga terdorong mempelajari representasi yang lebih redundan. *Dropout* dapat pula ditafsirkan sebagai bentuk ensemble implisit dari sejumlah besar sub-jaringan yang berbagi bobot, sebagaimana diilustrasikan pada Gambar 10.3.



Gambar 10.3 Ilustrasi Mekanisme *Dropout* pada Jaringan Saraf

Batch normalization menormalkan distribusi aktivasi pada setiap mini-batch untuk mengatasi *internal covariate shift*, yaitu perubahan distribusi masukan suatu lapisan akibat pembaruan parameter pada lapisan sebelumnya. Prosesnya diawali dengan perhitungan rata-rata dan varians batch, dilanjutkan normalisasi, kemudian transformasi skala dan pergeseran menggunakan dua parameter yang dipelajari.

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \tag{10.12}$$

$$y_i = \gamma \hat{x}_i + \beta \tag{10.13}$$

Global Average Pooling menghitung rata-rata seluruh nilai pada setiap *feature* map sebagai alternatif lapisan flatten.

$$\mathrm{GAP}(F_k) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} F_k(i,j) \tag{10.14}$$

Pada *VGG16* dengan keluaran 7×7×512, lapisan *flatten* akan menghasilkan vektor sepanjang 25.088 elemen sedangkan *Global Average Pooling* hanya menghasilkan 512 elemen. Reduksi ini menurunkan jumlah parameter pada lapisan dense berikutnya secara drastis sekaligus memaksa setiap *feature* map berkorespondensi langsung dengan tingkat kepercayaan terhadap suatu konsep visual.

Metrik Evaluasi dan *Confusion Matrix*

Confusion matrix membandingkan label sebenarnya dengan label hasil prediksi. Untuk permasalahan dengan lima kelas, matriks berukuran 5×5, dengan elemen diagonal utama menyatakan prediksi yang benar dan elemen di luar diagonal menyatakan kesalahan beserta arahnya. Pada kasus multikelas, besaran TP, FP, dan FN dihitung untuk setiap kelas menggunakan pendekatan one-versus-rest.

Precision menyatakan proporsi prediksi positif yang benar-benar positif, dan berkaitan dengan seberapa dapat dipercaya sebuah hasil positif.

$$Presisi = \frac{TP}{TP + FP} \tag{10.15}$$

Recall atau sensitivitas menyatakan proporsi sampel positif yang berhasil dideteksi. Pada penapisan penyakit menular, *recall* umumnya lebih kritis daripada *precision* karena kasus yang terlewat berpotensi menimbulkan keterlambatan penanganan sekaligus penularan lebih lanjut.

$$Recall = \frac{TP}{TP + FN} \tag{10.16}$$

F1-score merupakan rata-rata harmonik keduanya. Penggunaan rata-rata harmonik menyebabkan nilainya hanya tinggi apabila *precision* dan *recall* sama-sama tinggi, sehingga metrik ini efektif mendeteksi ketidakseimbangan di antara keduanya.

$$F1\text{-}score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \tag{10.17}$$

Untuk meringkas metrik per kelas menjadi satu nilai, digunakan rata-rata terbobot berdasarkan jumlah sampel setiap kelas.

$$M_{weighted} = \frac{\sum_{k=1}^{K} n_k M_k}{\sum_{k=1}^{K} n_k} \tag{10.18}$$

Perhatian: Akurasi Menyembunyikan Kegagalan pada Satu Kelas

Pada dataset seimbang, akurasi sering dianggap sebagai metrik yang memadai karena tidak dapat dimanipulasi dengan menebak kelas mayoritas. Anggapan itu benar sebagian, tetapi menyisakan satu celah penting.

Akurasi merupakan rata-rata. Sebuah model dengan lima kelas dapat mencapai akurasi 86 persen dengan cara mengenali empat kelas hampir sempurna sementara satu kelas hanya dikenali pada separuh kasusnya. Rata-ratanya tetap tinggi, dan angka tunggal itu tidak menyimpan jejak apa pun mengenai kelas yang gagal.

Persoalannya menjadi serius apabila kelas yang gagal kebetulan merupakan kelas yang penting secara klinis. Karena itu, evaluasi pada bab ini tidak berhenti pada metrik agregat melainkan diturunkan sampai tingkat kelas. Subbab 10.4.6 akan memperlihatkan bahwa celah tersebut bukan kemungkinan teoretis, melainkan kondisi yang benar-benar terjadi.

Explainable Artificial Intelligence

Model deep learning umumnya bersifat *black box*, yaitu menghasilkan keluaran tanpa penjelasan mengenai alasan di balik keputusannya. Pada domain medis, ketiadaan penjelasan menjadi hambatan serius bagi penerimaan klinis karena tenaga medis bertanggung jawab secara profesional atas setiap keputusan diagnosis. *Explainable Artificial Intelligence* berupaya menutup celah tersebut. Tiga metode digunakan pada bab ini.

Gradient-weighted Class Activation Mapping bekerja dengan memanfaatkan gradien skor kelas terhadap *feature* map pada lapisan konvolusi terakhir. Bobot kepentingan setiap *feature* map dihitung

melalui rata-rata gradien global, kemudian peta aktivasi disusun sebagai kombinasi linear terbobot yang dilewatkan fungsi ReLU agar hanya kontribusi positif yang ditampilkan.

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{10.19}$$

$$L_{Grad-CAM}^c = \mathrm{ReLU}\left(\sum_k \alpha_k^c A^k\right) \tag{10.20}$$

Local Interpretable Model-agnostic Explanations menjelaskan prediksi model kompleks dengan membangun model sederhana yang dapat ditafsirkan pada area lokal di sekitar sampel yang diamati. Pada data citra, LIME membagi citra menjadi sejumlah superpixel, membangkitkan sampel perturbasi dengan mengaktifkan dan menonaktifkan superpixel secara acak, lalu melatih regresi linear terbobot untuk memperkirakan kontribusi setiap segmen.

$$\xi(x) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{10.21}$$

Occlusion Sensitivity merupakan metode berbasis perturbasi yang paling langsung secara konseptual. Metode ini menutup sebagian area citra menggunakan jendela berukuran tetap, kemudian mengamati perubahan *confidence* score terhadap kelas yang diprediksi.

$$OS(i,j) = P_c(x) - P_c(x_{\odot(i,j)}) \tag{10.22}$$

Karakteristik ketiga metode tersebut berbeda dan saling melengkapi, sebagaimana dirangkum pada Tabel 10.3.

Tabel 10.3 Perbandingan Karakteristik Tiga Metode Interpretasi

Aspek	**Grad-CAM**	**LIME**	***Occlusion Sensitivity***
Pendekatan	Berbasis gradien	Model pengganti lokal	Perturbasi langsung
Akses yang dibutuhkan	Gradien internal model	Hanya keluaran model	Hanya keluaran model
Resolusi penjelasan	Kasar, 14×14	Segmen superpixel	Sesuai ukuran jendela

Biaya komputasi	Rendah	Sedang hingga tinggi	Tinggi
Stabilitas hasil	Deterministik	Stokastik, dapat bervariasi	Deterministik

Kalibrasi dan *Overconfidence*

Sebuah model dikatakan terkalibrasi dengan baik apabila tingkat keyakinan yang dihasilkannya mendekati tingkat ketepatan yang sesungguhnya. Pada model terkalibrasi, sekumpulan prediksi dengan keyakinan 90 persen seharusnya benar pada sekitar 90 persen kasus. Ukuran yang lazim digunakan untuk menilai kalibrasi adalah *Expected Calibration Error*, yaitu rata-rata terbobot selisih antara ketepatan dan keyakinan pada sejumlah kelompok nilai keyakinan.

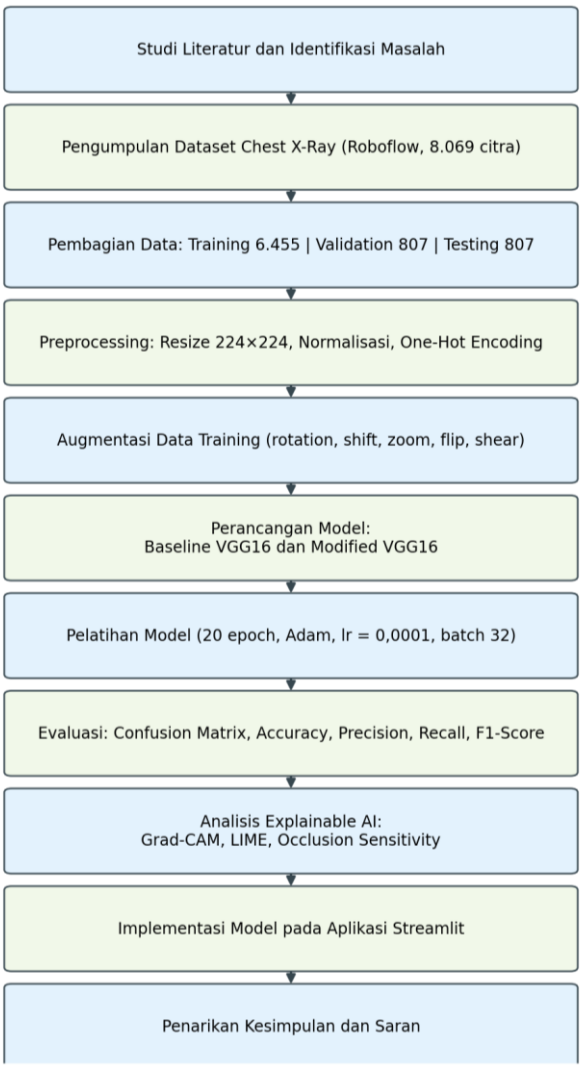
$$ECE = \sum_{b=1}^{B} \frac{|S_b|}{N} |\mathrm{acc}(S_b) - \mathrm{conf}(S_b)| \qquad (10.23)$$

Kalibrasi merupakan sifat yang berbeda dari ketepatan, dan keduanya tidak selalu berjalan seiring. Sebuah model dapat memiliki akurasi tinggi namun kalibrasi buruk, dan sebaliknya. Perbedaan ini penting karena pada penerapan praktis, nilai keyakinan sering digunakan sebagai dasar penyaringan: prediksi berkeyakinan tinggi diterima, prediksi berkeyakinan rendah diteruskan kepada manusia. Penyaringan semacam itu hanya bermakna apabila nilai keyakinannya dapat dipercaya.

➢ Metodologi

Alur Metodologi

Rangkaian tahapan yang ditempuh pada bab ini disajikan pada Gambar 10.4. Prinsip yang dipegang sepanjang rangkaian tersebut adalah pengendalian variabel: seluruh faktor selain rancangan lapisan klasifikasi ditetapkan identik pada kedua model.



Gambar 10.4 Alur Metodologi Klasifikasi Penyakit Paru-Paru

Sumber dan Karakteristik Data

Data yang digunakan berasal dari platform Roboflow dan terdiri atas 8.069 citra *chest X-ray* yang telah dilabeli ke dalam lima kelas. Seluruh citra berformat gambar digital standar dengan ukuran bervariasi sehingga memerlukan penyeragaman pada tahap prapemrosesan. Distribusi jumlah citra pada setiap kelas beserta pembagiannya ke dalam tiga subset disajikan pada Tabel 10.4.

Tabel 10.4 Distribusi dan Pembagian Data per Kelas

Kelas	***Training***	***Validation***	***Testing***	**Total**	**Proporsi (%)**
Bacterial *Pneumonia*	1.281	171	154	1.606	19,90
Corona Virus Disease	1.302	149	172	1.623	20,11
Normal	1.290	163	156	1.609	19,94
Tuberculosis	1.300	163	163	1.626	20,15
Viral *Pneumonia*	1.282	161	162	1.605	19,89
Total	6.455	807	807	8.069	100,00

Selisih antara kelas terbanyak dan kelas paling sedikit hanya 21 citra, atau kurang dari 1,3 persen dari rata-rata jumlah citra per kelas. Data karena itu dapat dikategorikan seimbang, sehingga tidak diperlukan teknik penanganan ketimpangan kelas seperti oversampling maupun pembobotan kelas pada fungsi kerugian.

Catatan: Keseimbangan Data Menyederhanakan Sebagian Persoalan, Bukan Seluruhnya

Keseimbangan data pada bab ini menghilangkan beberapa persoalan sekaligus. Tidak diperlukan SMOTE maupun teknik penyeimbangan lain, sehingga risiko kebocoran data akibat urutan penerapan yang keliru tidak muncul. Akurasi juga menjadi metrik yang lebih dapat dipercaya, karena model yang selalu menebak satu kelas hanya akan memperoleh akurasi sekitar 20 persen.

Namun keseimbangan jumlah data tidak berarti keseimbangan tingkat kesulitan. Kelima kelas memiliki jumlah sampel yang hampir sama, tetapi tidak sama-sama mudah dipelajari. Kelas dengan penanda visual yang lemah akan tetap sulit dikenali meskipun jumlah sampelnya memadai.

Perbedaan inilah yang sering terlewat. Ketimpangan jumlah data terlihat pada tabel distribusi dan mudah dideteksi sejak awal; ketimpangan tingkat kesulitan hanya terlihat pada metrik per kelas setelah model dilatih.

Pembagian data dilakukan secara stratified dengan proporsi 80 persen untuk pelatihan, 10 persen untuk validasi, dan 10 persen untuk pengujian. Data validasi digunakan untuk memantau performa selama pelatihan tanpa ikut memperbarui bobot, sedangkan data pengujian hanya digunakan satu kali pada tahap akhir agar hasil evaluasi tidak bias akibat penyesuaian model terhadap data uji.

Prapemrosesan

Seluruh citra diubah ukurannya menjadi 224 × 224 piksel menggunakan interpolasi bilinear, yaitu ukuran masukan standar arsitektur VGG16. Penyeragaman ini diperlukan agar proses *batching* dapat dilakukan dan bobot pada lapisan dense dapat didefinisikan secara tetap.

Normalisasi dilakukan menggunakan fungsi *preprocessing* bawaan VGG16, yang mengonversi urutan kanal warna dari RGB menjadi BGR dan mengurangkan nilai rata-rata setiap kanal yang dihitung dari dataset *ImageNet*. Penggunaan fungsi normalisasi yang sama dengan tahap pelatihan bobot *pretrained* merupakan hal yang krusial; apabila normalisasi dilakukan dengan cara berbeda, distribusi masukan tidak akan sesuai dengan asumsi bobot *pretrained* sehingga performa dapat menurun tajam.

Label diubah ke dalam bentuk *one-hot encoding*, yaitu vektor biner sepanjang lima elemen dengan nilai satu pada indeks kelas yang bersesuaian. Representasi ini bersesuaian langsung dengan keluaran softmax dan konsisten dengan fungsi kerugian *categorical crossentropy*.

Augmentasi Data

Augmentasi memperbesar variasi data pelatihan melalui transformasi geometris terhadap citra asli, tanpa menambah jumlah citra. Teknik ini penting pada domain citra medis karena pengumpulan data berlabel menuntut biaya dan keahlian yang tinggi. Augmentasi hanya diterapkan pada data pelatihan; data validasi dan data pengujian hanya melalui penyeragaman ukuran dan normalisasi agar hasil evaluasi mencerminkan kondisi data sebenarnya.

Transformasi rotasi, sebagai contoh, memutar citra pada sudut tertentu di sekitar titik pusatnya.

$$x' = x\cos\theta - y\sin\theta, \qquad y' = x\sin\theta + y\cos\theta, \qquad \theta \in [-20^\circ, 20^\circ] \tag{10.24}$$

Karena parameter setiap transformasi dibangkitkan secara acak pada setiap *epoch*, model praktis tidak pernah melihat citra yang persis sama dua kali sepanjang pelatihan. Sepanjang 20 *epoch*, model secara efektif menerima 129.100 sampel yang bervariasi meskipun jumlah citra aslinya tetap 6.455. Konfigurasi lengkap parameter augmentasi disajikan pada Tabel 10.5.

Tabel 10.5 Konfigurasi Parameter Augmentasi Data Pelatihan

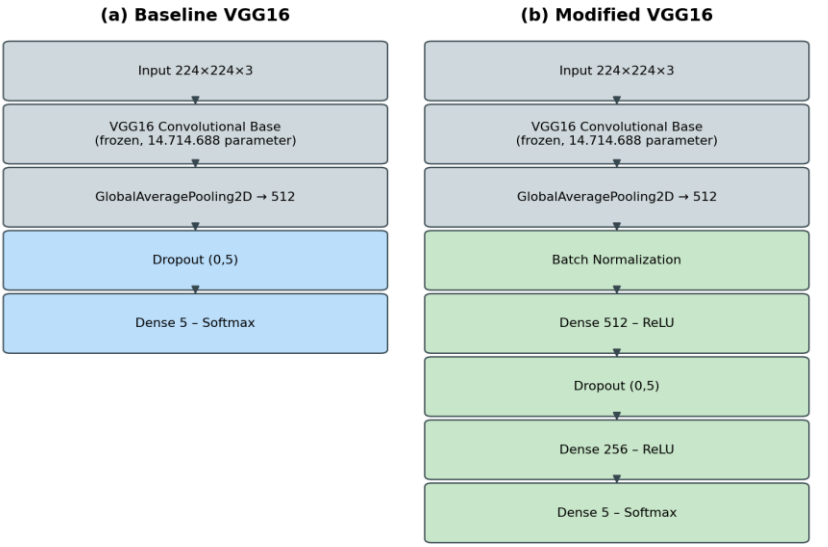
Parameter	**Nilai**	**Efek terhadap Citra**
rotation_range	20	Rotasi acak pada rentang −20° hingga +20°
width_shift_range	0,15	Pergeseran horizontal hingga 15% lebar citra
height_shift_range	0,15	Pergeseran vertikal hingga 15% tinggi citra
zoom_range	0,15	Penskalaan acak pada rentang 0,85 hingga 1,15
shear_range	0,1	Kemiringan citra hingga 0,1 radian
horizontal_flip	True	Pembalikan citra terhadap sumbu vertikal
fill_mode	nearest	Pengisian area kosong dengan piksel terdekat

Peringatan: Horizontal Flip Membalik Posisi Jantung
Satu parameter pada Tabel 10.5 memerlukan pertimbangan yang tidak diperlukan pada citra non-medis.
Pembalikan horizontal pada citra sehari-hari tidak menimbulkan persoalan: seekor kucing yang menghadap ke kiri tetap seekor kucing ketika menghadap ke kanan. Pada citra *chest X-ray*, pembalikan horizontal menghasilkan citra dengan bayangan jantung berada di sisi kanan, yaitu kondisi yang secara anatomis merupakan kelainan langka bernama dekstrokardia.
Dengan kata lain, augmentasi ini membangkitkan citra yang secara anatomis tidak realistis dan mengajarkan kepada model bahwa posisi jantung tidak membawa informasi. Konsekuensinya bersifat dua arah. Di satu sisi, model menjadi lebih tahan terhadap variasi orientasi dan tidak mempelajari sisi tubuh sebagai

penanda kelas hal yang justru diinginkan pada bab ini, karena fokus klasifikasi berada pada pola parenkim paru. Di sisi lain, apabila permasalahan yang dihadapi melibatkan kelainan yang bergantung pada sisi tubuh, augmentasi ini akan merusak informasi yang justru paling menentukan.
Keputusan menggunakannya pada bab ini dapat dibenarkan, tetapi keputusan tersebut perlu diambil secara sadar dan bukan karena parameter itu tersedia secara bawaan pada pustaka yang digunakan.

Rancangan Dua Model

Kedua model yang dibandingkan menggunakan *convolutional base* yang identik dan seluruhnya dibekukan, serta hanya berbeda pada rancangan lapisan klasifikasi. Perbandingan keduanya disajikan pada Gambar 10.5.



Gambar 10.5 Perbandingan Rancangan Lapisan Klasifikasi Kedua Model

Model baseline menyusun lapisan klasifikasi seminimal mungkin: keluaran *convolutional base* berukuran 7×7×512 direduksi menggunakan *Global Average Pooling* menjadi vektor 512 elemen, dilewatkan *dropout*, lalu langsung dipetakan ke lima neuron keluaran dengan aktivasi softmax. Rancangan ini merupakan bentuk paling sederhana dari *feature extraction* dan berfungsi sebagai titik acuan.

Model modifikasi menyisipkan tiga komponen tambahan di antara *Global Average Pooling* dan lapisan keluaran, yaitu *batch normalization*, dense *layer* 512 neuron dengan aktivasi ReLU, *dropout*, dan dense *layer* 256 neuron dengan aktivasi ReLU. Penambahan dua dense *layer* memberikan kapasitas untuk mempelajari kombinasi fitur non-linear yang lebih kompleks, dengan struktur menyempit bertahap dari 512 menuju 256 dan akhirnya 5 neuron. *Batch normalization* ditempatkan tepat setelah *Global Average Pooling* agar distribusi fitur masukan classifier menjadi stabil, sedangkan *dropout* ditempatkan di antara kedua dense *layer* sebagai penyeimbang terhadap peningkatan kapasitas.

Jumlah parameter pada sebuah dense *layer* dihitung dari perkalian jumlah neuron masukan dan keluaran ditambah satu bias untuk setiap neuron keluaran.

$$P_{dense} = (n_{in} \times n_{out}) + n_{out} \tag{10.25}$$

Penerapan perhitungan tersebut pada kedua rancangan menghasilkan perbedaan kapasitas yang sangat besar, sebagaimana dirinci pada Tabel 10.6.

Tabel 10.6 Perbandingan Jumlah Parameter Kedua Model

Komponen	**Baseline**	**Modified**
Convolutional base (dibekukan)	14.714.688	14.714.688
Global Average Pooling	0	0
Batch Normalization	—	2.048
Dense 512 (ReLU)	—	262.656
Dense 256 (ReLU)	—	131.328
Dense 5 (*Softmax*)	2.565	1.285
Total parameter	14.717.253	15.112.005
Parameter yang dilatih	2.565	396.293
Proporsi yang dilatih (%)	0,02	2,62

Model modifikasi memiliki parameter yang dapat dilatih sekitar 154 kali lebih banyak dibandingkan model baseline. Perbedaan inilah yang menjadi variabel tunggal pada perbandingan di bab ini. Selisih pada lapisan keluaran, yaitu 2.565 berbanding 1.285, terjadi karena masukan lapisan keluaran pada model baseline berasal langsung dari 512 fitur hasil *Global Average Pooling* sedangkan pada model modifikasi berasal dari 256 neuron dense *layer* sebelumnya.

Konfigurasi Pelatihan dan Skenario Pengujian

Konfigurasi *hyperparameter* yang digunakan pada kedua model disajikan pada Tabel 10.7 dan diterapkan secara identik agar perbandingan dapat dilakukan secara objektif.

Tabel 10.7 Konfigurasi *Hyperparameter* Pelatihan

Hyperparameter	**Nilai**	**Alasan Pemilihan**
Ukuran citra masukan	224 × 224	Ukuran masukan standar arsitektur VGG16
Batch size	32	Menyeimbangkan kestabilan gradien dan kebutuhan memori
Epoch	20	Memadai untuk konvergensi classifier pada base yang dibekukan
Learning rate	0,0001	Pembaruan bobot berlangsung bertahap dan stabil
Optimizer	Adam	Adaptif terhadap setiap parameter dan cepat konvergen
Fungsi kerugian	*Categorical crossentropy*	Sesuai klasifikasi multikelas dengan label one-hot
Jumlah kelas	5	Jumlah kategori kondisi pada data

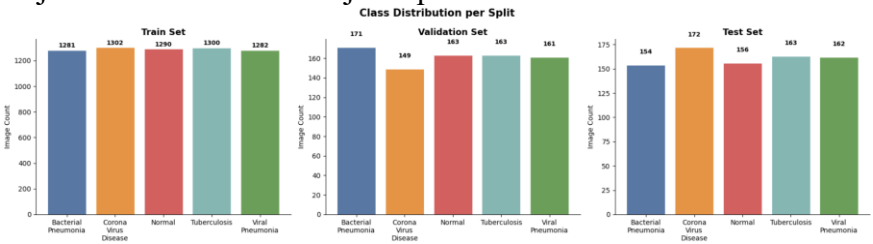
Dengan 6.455 citra pelatihan dan batch size 32, setiap *epoch* memerlukan 202 langkah pembaruan parameter, sehingga sepanjang 20 *epoch* terjadi 4.040 kali pembaruan. Evaluasi dilakukan terhadap 807 citra data pengujian yang belum pernah dilihat model, mencakup penyusunan *confusion matrix*, perhitungan metrik per kelas menggunakan pendekatan *one-versus-rest*, serta perhitungan metrik agregat menggunakan rata-rata terbobot. Analisis interpretabilitas

menggunakan Grad-CAM, LIME, dan *Occlusion Sensitivity* diterapkan pada model terpilih.

➢ Hasil dan Ujicoba

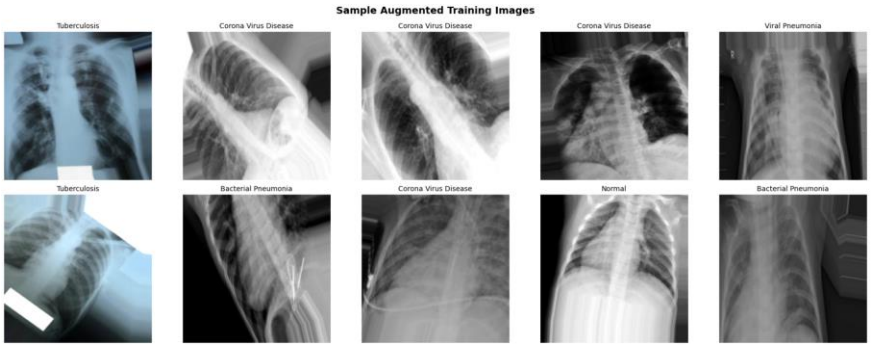
Distribusi Kelas dan Hasil Augmentasi

Sebelum hasil pelatihan dibahas, dua hal perlu diperiksa terlebih dahulu, yaitu apakah distribusi kelas benar-benar seimbang pada ketiga subset dan apakah augmentasi menghasilkan variasi yang wajar. Distribusi kelas disajikan pada Gambar 10.6.



Gambar 10.6 Distribusi Kelas pada Data Pelatihan, Validasi, dan Pengujian

Distribusi pada ketiga subset relatif seimbang sehingga setiap kelas memperoleh representasi yang memadai. Kondisi ini menjadikan akurasi sebagai metrik yang dapat dipercaya sebagai gambaran umum, sekalipun sebagaimana telah diperingatkan pada Subbab 10.2.6 tidak memadai sebagai gambaran menyeluruh.
Contoh citra hasil augmentasi disajikan pada Gambar 10.7.



Gambar 10.7 Contoh Citra Hasil Augmentasi pada Data Pelatihan

Pemeriksaan terhadap hasil augmentasi memperlihatkan bahwa struktur anatomi utama berupa lapangan paru, bayangan jantung, dan kerangka tulang iga tetap dapat dikenali pada seluruh citra hasil transformasi. Hal ini menandakan parameter augmentasi berada pada rentang yang wajar. Apabila parameternya ditetapkan terlalu agresif rotasi melebihi 45 derajat atau zoom melebihi 50 persen struktur anatomi dapat terpotong sehingga justru menurunkan kualitas pembelajaran.

Perbandingan Konfigurasi Kedua Model

Perbedaan konfigurasi kedua model dirangkum kembali pada Tabel 10.8 sebelum hasilnya dibandingkan.

Tabel 10.8 Perbandingan Konfigurasi Kedua Model

Konfigurasi	***VGG16* Baseline**	***VGG16* Modified**
Model dasar	*VGG16 pretrained ImageNet*	*VGG16 pretrained ImageNet*
Status *convolutional base*	Dibekukan	Dibekukan
Global Average Pooling	Digunakan	Digunakan
Batch Normalization	Tidak digunakan	Digunakan
Dense *layer* tambahan	Tidak ada	512 neuron + 256 neuron
Dropout	0,5	0,5
Lapisan keluaran	Softmax 5 kelas	Softmax 5 kelas
Parameter yang dilatih	2.565	396.293

Seluruh komponen bersifat identik kecuali pada bagian *classifier*. Kesamaan ini memastikan perbandingan bersifat adil dan perbedaan performa yang teramati murni disebabkan oleh perbedaan rancangan lapisan klasifikasi, bukan oleh perbedaan data, prapemrosesan, maupun *hyperparameter*.

Hasil Model Baseline

Model baseline berhasil mengklasifikasikan 652 dari 807 citra data pengujian dengan benar, sehingga menghasilkan 155 kesalahan prediksi dan akurasi sebesar 0,808. Rincian *confusion matrix*-nya disajikan pada Tabel 10.9.

Tabel 10.9 *Confusion Matrix* Model Baseline pada Data Pengujian

Aktual \ Prediksi	BP	CVD	N	TB	VP	Total
Bacterial *Pneumonia* (BP)	113	3	11	1	26	154
Corona Virus Disease (CVD)	1	165	2	4	0	172
Normal (N)	5	1	148	1	1	156
Tuberculosis (TB)	0	13	0	149	1	163
Viral *Pneumonia* (VP)	42	3	39	1	77	162
Total prediksi	161	185	200	156	105	807

Pola kesalahan yang paling menonjol terjadi pada baris terakhir. Dari 162 citra Viral *Pneumonia*, hanya 77 yang diprediksi benar; sebanyak 42 citra keliru diprediksi sebagai Bacterial *Pneumonia* dan 39 citra sebagai Normal. Dengan kata lain, lebih dari separuh citra Viral *Pneumonia* gagal dikenali oleh model baseline. Pola kedua yang menonjol adalah kebingungan pada arah sebaliknya, yaitu 26 citra Bacterial *Pneumonia* yang keliru diprediksi sebagai Viral *Pneumonia*. Selain itu terdapat 13 citra Tuberculosis yang keliru diprediksi sebagai Corona Virus Disease. Ketiga pola tersebut bersesuaian dengan perkiraan yang telah diuraikan pada Subbab 10.2.1: kesulitan muncul pada kelas dengan penanda visual paling lemah, dan kesalahannya mengarah pada kelas yang pola visualnya berdekatan.

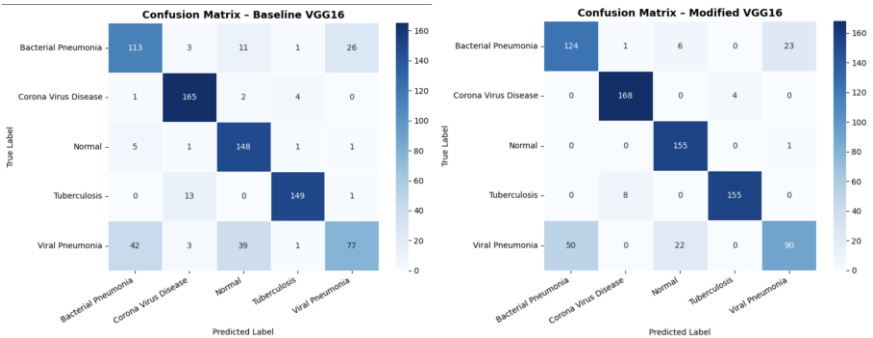
Hasil Model Modifikasi

Model modifikasi berhasil mengklasifikasikan 692 dari 807 citra dengan benar, sehingga menghasilkan 115 kesalahan dan akurasi sebesar 0,858. Rincian *confusion matrix*-nya disajikan pada Tabel 10.10.

Tabel 10.10 *Confusion Matrix* Model Modifikasi pada Data Pengujian

Aktual \ Prediksi	BP	CVD	N	TB	VP	Total
Bacterial *Pneumonia* (BP)	124	1	6	0	23	154
Corona Virus Disease (CVD)	0	168	0	4	0	172
Normal (N)	0	0	155	0	1	156
Tuberculosis (TB)	0	8	0	155	0	163
Viral *Pneumonia* (VP)	50	0	22	0	90	162
Total prediksi	174	177	183	159	114	807

Perbandingan visual kedua *confusion matrix* disajikan pada Gambar 10.8.

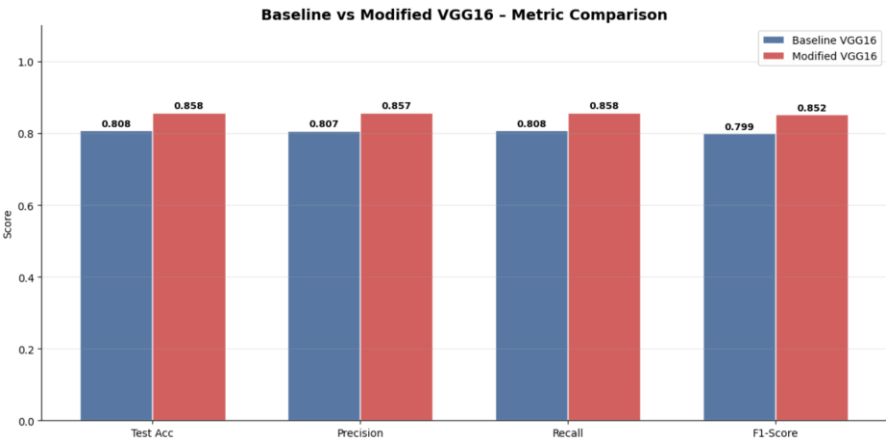


Gambar 10.8 Perbandingan *Confusion Matrix* Kedua Model

Peningkatan jumlah prediksi benar terjadi pada seluruh kelas tanpa terkecuali. Perbaikan paling mencolok terjadi pada kelas Normal, yang kesalahannya turun dari delapan citra menjadi satu citra saja. Hal ini penting secara klinis, sebab kesalahan mengklasifikasikan citra dengan kelainan sebagai normal merupakan jenis kesalahan yang paling berisiko.

Peningkatan yang Diberikan Modifikasi

Perbandingan metrik agregat kedua model disajikan pada Gambar 10.9 dan dirinci pada Tabel 10.11.



Gambar 10.9 Perbandingan Metrik Evaluasi Kedua Model

Tabel 10.11 Perbandingan Metrik Evaluasi Agregat

Metrik	**Baseline**	**Modified**	**Selisih**	**Peningkatan Relatif**
Accuracy	0,808	0,858	+0,050	6,14%
Precision (terbobot)	0,807	0,857	+0,050	6,22%
Recall (terbobot)	0,808	0,858	+0,050	6,14%
F1-Score (terbobot)	0,799	0,852	+0,053	6,58%
Jumlah prediksi benar	652	692	+40	6,13%
Jumlah kesalahan	155	115	−40	−25,81%

Baris terakhir pada Tabel 10.11 menyampaikan informasi yang paling bermakna. Modifikasi lapisan klasifikasi menghilangkan 40 dari 155 kesalahan yang sebelumnya dilakukan model baseline, atau setara dengan reduksi kesalahan sebesar 25,81 persen. Ketika akurasi telah berada pada tingkat yang cukup tinggi, reduksi kesalahan relatif merupakan indikator yang lebih informatif dibandingkan peningkatan akurasi absolut, sebab semakin tinggi akurasi semakin sulit memperoleh peningkatan tambahan.

Peningkatan tersebut dapat dijelaskan melalui dua mekanisme. Model baseline hanya mampu membentuk batas keputusan linear pada ruang fitur berdimensi 512, sedangkan penambahan dua dense *layer* dengan aktivasi ReLU memungkinkan model modifikasi membentuk batas keputusan non-linear yang jauh lebih fleksibel. Pada permasalahan dengan kelas yang saling tumpang tindih, fleksibilitas ini menentukan. Mekanisme kedua adalah kehadiran *batch normalization*, yang menempatkan seluruh fitur keluaran *Global Average Pooling* pada skala yang sebanding sehingga proses optimasi tidak didominasi oleh sebagian kecil fitur bermagnitudo besar.

Perlu ditegaskan bahwa kedua model bekerja pada representasi fitur yang persis sama, karena *convolutional base*-nya identik dan dibekukan. Peningkatan yang diperoleh karena itu sepenuhnya berasal dari kemampuan classifier memanfaatkan fitur yang tersedia secara lebih efektif, bukan dari perbaikan kualitas fitur itu sendiri.

Sampai titik ini, seluruh angka mengarah pada satu kesimpulan yang menyenangkan: modifikasi berhasil, dan Metodologi dapat dianggap selesai. Subbab berikutnya memperlihatkan mengapa kesimpulan itu terlalu dini.

Evaluasi Per Kelas: Angka yang Tidak Terlihat pada Akurasi

Metrik agregat pada Tabel 10.11 merupakan rata-rata terbobot atas lima kelas. Menurunkannya kembali menjadi metrik per kelas memperlihatkan sesuatu yang tidak tampak pada bentuk rata-ratanya. Hasil perhitungan untuk model baseline disajikan pada Tabel 10.12.

Tabel 10.12 Metrik Evaluasi Per Kelas Model Baseline

Kelas	**TP**	**FP**	**FN**	***Precision***	***Recall***	**F1-Score**	***Support***
Bacterial Pneumonia	113	48	41	0,7019	0,7338	0,7175	154
Corona Virus Disease	165	20	7	0,8919	0,9593	0,9244	172
Normal	148	52	8	0,7400	0,9487	0,8315	156
Tuberculosis	149	7	14	0,9551	0,9141	0,9342	163
Viral *Pneumonia*	77	28	85	0,7333	0,4753	0,5768	162
Rerata makro	—	—	—	0,8044	0,8062	0,7968	807
Rerata terbobot	—	—	—	0,8072	0,8079	0,7991	807

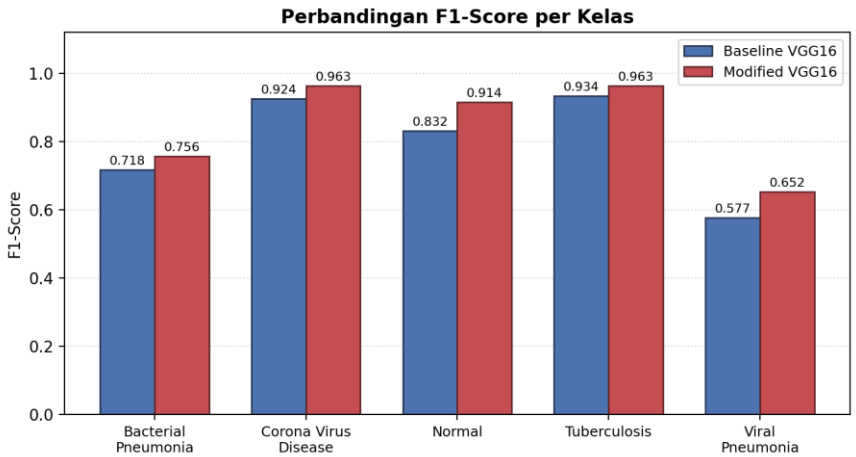
Hasil yang bersesuaian untuk model modifikasi disajikan pada Tabel 10.13.

Tabel 10.13 Metrik Evaluasi Per Kelas Model Modifikasi

Kelas	**TP**	**FP**	**FN**	***Precision***	***Recall***	**F1-Score**	***Support***
Bacterial Pneumonia	124	50	30	0,7126	0,8052	0,7561	154
Corona Virus Disease	168	9	4	0,9492	0,9767	0,9628	172
Normal	155	28	1	0,8470	0,9936	0,9145	156
Tuberculosis	155	4	8	0,9748	0,9509	0,9627	163

Viral Pneumonia	90	24	72	0,7895	0,5556	0,6522	162
Rerata makro	—	—	—	0,8546	0,8564	0,8496	807
Rerata terbobot	—	—	—	0,8574	0,8575	0,8516	807

Perbandingan *F1-score* per kelas kedua model divisualisasikan pada Gambar 10.10.



Gambar 10.10 Perbandingan *F1-Score* Per Kelas Kedua Model

Tiga hal dapat dibaca dari ketiga sajian tersebut. Pertama, peningkatan terjadi pada seluruh kelas, dengan kenaikan *F1-score* terbesar pada kelas Normal sebesar 0,0830 dan kelas Viral *Pneumonia* sebesar 0,0754. Kedua, urutan tingkat kesulitan kelas tetap sama pada kedua model. Ketiga dan inilah yang paling penting sebaran nilai antarkelas sangat lebar.

Pada model modifikasi, *F1-score* tertinggi mencapai 0,9628 pada kelas *Corona Virus Disease* sementara yang terendah hanya 0,6522 pada kelas Viral *Pneumonia*. Rentang sebesar 0,31 tersebut sepenuhnya lenyap ketika kelima nilai dirata-ratakan menjadi 0,8516. Angka akurasi 85,75 persen yang dilaporkan sebagai capaian utama sesungguhnya merupakan rata-rata dari empat kelas yang dikenali dengan sangat baik dan satu kelas yang dikenali hanya pada separuh kasusnya.

Perhatian: *Recall* 0,5556 Berarti Hampir Separuh Kasus Terlewat

Angka yang paling perlu digarisbawahi pada Tabel 10.13 adalah *recall* kelas Viral *Pneumonia* sebesar 0,5556.

Recall menyatakan proporsi kasus yang benar-benar merupakan Viral *Pneumonia* dan berhasil dikenali oleh model. Nilai 0,5556 berarti dari setiap seratus pasien dengan *pneumonia* virus, sekitar 44 orang tidak akan dikenali sebagai *pneumonia* virus oleh model. Pada penapisan penyakit menular, kasus yang terlewat berpotensi menimbulkan keterlambatan penanganan sekaligus penularan lebih lanjut.

Perbandingannya menjadi tajam apabila diletakkan berdampingan: pada kelas Normal, *recall* model mencapai 0,9936 hampir seluruh citra normal dikenali dengan benar. Model bekerja sangat baik pada kelas yang paling tidak memerlukan bantuan, dan bekerja paling buruk pada salah satu kelas yang justru memerlukannya.

Perlu dicatat bahwa nilai 0,5556 ini sudah merupakan hasil setelah modifikasi. Model baseline hanya mencapai 0,4753, yaitu kurang dari separuh. Modifikasi memperbaiki keadaan, tetapi tidak menyelesaikannya.

Pergeseran Arah Kesalahan: Sebuah Penelusuran

Subbab ini menelusuri satu keganjilan yang muncul apabila kedua *confusion matrix* dibandingkan baris demi baris pada kelas Viral *Pneumonia*. Sekilas, model modifikasi jelas lebih baik: prediksi benarnya naik dari 77 menjadi 90. Namun kenaikan tersebut tidak terjadi secara merata pada kedua arah kesalahan, sebagaimana dirinci pada Tabel 10.14.

Tabel 10.14 Pergeseran Arah Kesalahan pada Kelas Viral *Pneumonia*

Arah Prediksi	Baseline	Modified	Perubahan
Dikenali benar sebagai Viral *Pneumonia*	77	90	+13
Keliru diprediksi sebagai Bacterial *Pneumonia*	42	50	+8

Keliru diprediksi sebagai Normal	39	22	−17
Keliru diprediksi sebagai kelas lain	4	0	−4
Total citra Viral *Pneumonia*	162	162	—

Terdapat satu angka yang berlawanan arah dengan seluruh angka lainnya. Kesalahan menuju kelas Normal turun tajam dari 39 menjadi 22 citra, sedangkan kesalahan menuju kelas Bacterial *Pneumonia* justru naik dari 42 menjadi 50 citra. Padahal, keseluruhan performa model meningkat. Pertanyaannya adalah mengapa satu jenis kesalahan bertambah pada model yang secara umum lebih baik.

Analisis: Mengapa Kesalahan Bergeser dari Normal ke Bacterial *Pneumonia*

Terdapat tiga pengamatan yang secara bersama-sama mengarah pada satu penjelasan.

Pengamatan pertama. Pada model modifikasi, seluruh kesalahan pada baris kelas Normal praktis lenyap: dari 156 citra normal, 155 diprediksi benar. Model menjadi jauh lebih tegas dalam menyatakan bahwa suatu citra bukan citra normal.

Pengamatan kedua. Ketegasan tersebut memiliki konsekuensi. Sebuah citra Viral *Pneumonia* dengan infiltrat yang sangat halus sebelumnya diprediksi sebagai Normal oleh model baseline, karena model tersebut tidak cukup peka menangkap kelainan ringan. Pada model modifikasi, kelainan ringan itu tertangkap citranya tidak lagi dianggap normal. Namun setelah dinyatakan tidak normal, model tetap harus memilih salah satu dari empat kelas patologis, dan pilihannya jatuh pada Bacterial *Pneumonia*.

Pengamatan ketiga. Bacterial *Pneumonia* merupakan kelas dengan *precision* terendah pada kedua model, yaitu 0,7019 dan 0,7126. Rendahnya *precision* berarti kelas ini menampung banyak citra dari kelas lain. Dengan kata lain, Bacterial *Pneumonia* berperan sebagai kelas penampung bagi kasus patologis yang tidak berhasil dikenali secara spesifik.

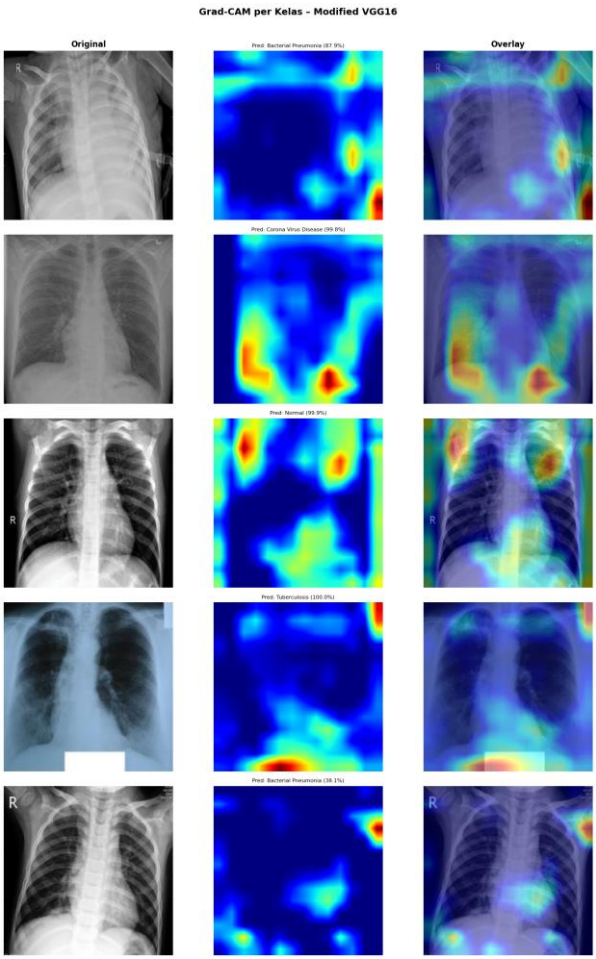
Kesimpulan yang wajar. Modifikasi tidak membuat model lebih mampu membedakan *pneumonia* virus dari *pneumonia* bakteri; yang meningkat adalah kemampuannya membedakan kondisi patologis dari kondisi normal. Kasus yang sebelumnya lolos sebagai normal kini tertangkap sebagai patologis, tetapi kemudian salah dialamatkan.
Mengapa hal ini penting. Dari sudut pandang klinis, pergeseran tersebut sesungguhnya menguntungkan meskipun tidak tercermin pada akurasi. Kesalahan mengklasifikasikan Viral *Pneumonia* sebagai Bacterial *Pneumonia* tetap mengarahkan pasien pada jalur penanganan *pneumonia*; kesalahan mengklasifikasikannya sebagai Normal berpotensi menyebabkan pasien tidak memperoleh penanganan sama sekali. Penurunan 17 kasus pada arah yang berbahaya merupakan perbaikan bermakna, sementara kenaikan 8 kasus pada arah yang relatif aman merupakan biaya yang wajar.
Pemeriksaan yang perlu dilakukan. Bandingkan biaya kekeliruan secara eksplisit dengan menyusun matriks biaya, kemudian evaluasi kembali kedua model menggunakan biaya tersebut dan bukan menggunakan akurasi. Apabila peringkat kedua model tetap sama, kesimpulan bab ini bertahan. Apabila berubah, berarti akurasi selama ini menyembunyikan perbedaan yang lebih penting daripada yang diperlihatkannya.

Penelusuran ini memperlihatkan bahwa dua model dengan selisih akurasi lima poin dapat memiliki perilaku yang berbeda secara kualitatif, dan perbedaan itu hanya terlihat apabila *confusion matrix* dibaca sampai ke arah kesalahannya, bukan sekadar ke jumlah totalnya.

Analisis *Explainable Artificial Intelligence*

Sampai titik ini seluruh analisis bertumpu pada angka. Bagian ini memeriksa hal yang berbeda: bukan seberapa sering model benar, melainkan atas dasar apa model mengambil keputusannya. Pemeriksaan ini diterapkan pada model modifikasi menggunakan tiga metode interpretasi.

Hasil visualisasi Grad-CAM pada kelima kelas disajikan pada Gambar 10.11.



Gambar 10.11Visualisasi Grad-CAM pada Lima Kelas Kondisi Paru-Paru

Pada sebagian besar citra, aktivasi tertinggi yang ditandai warna merah dan kuning berada pada area lapangan paru, yaitu lokasi yang secara medis memang relevan untuk membedakan kelima kondisi. Namun pada sebagian sampel terlihat aktivasi pada area di luar struktur paru, yaitu bagian tepi citra dan area sekitar tulang dada.

Perhatian: Aktivasi di Luar Paru Merupakan Pertanda yang Perlu Ditelusuri

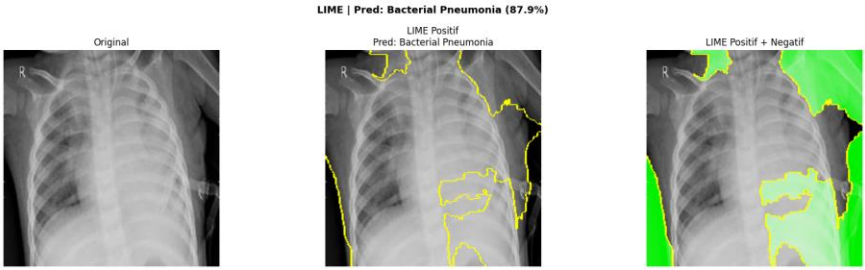
Aktivasi pada area non-anatomis dapat menandakan bahwa model memanfaatkan penanda tidak langsung, misalnya perbedaan pencahayaan, kontras latar, penanda huruf pada citra, atau artefak pemrosesan yang secara kebetulan berkorelasi dengan kelas tertentu pada data yang digunakan.

Fenomena ini dikenal sebagai *shortcut learning*, yaitu kondisi ketika model menemukan jalan pintas statistik yang efektif pada data pelatihan namun tidak memiliki dasar kausal yang benar. Kemungkinan ini bukan hal yang mengada-ada pada data citra *chest X-ray*, sebab citra dari sumber yang berbeda sering memiliki karakteristik teknis yang berbeda pula dan apabila satu kelas kebetulan lebih banyak berasal dari satu sumber, karakteristik teknis itu menjadi penanda yang jauh lebih mudah dipelajari daripada pola patologisnya.

Konsekuensinya bersifat tertunda. Model yang mengandalkan jalan pintas akan tampak berperforma baik pada data uji yang berasal dari distribusi yang sama, dan baru menurun tajam ketika diterapkan pada data dari rumah sakit atau perangkat pencitraan yang berbeda. Akurasi 85,75 persen pada bab ini karena itu belum dapat diperlakukan sebagai taksiran performa di lapangan.

Pemeriksaan yang perlu dilakukan. Terapkan segmentasi lapangan paru terlebih dahulu, sehingga area di luar paru tertutup sebelum citra dimasukkan ke model, kemudian latih ulang. Apabila akurasinya bertahan, aktivasi di luar paru hanyalah efek samping resolusi Grad-CAM yang kasar. Apabila akurasinya turun berarti sebagian kemampuan model memang bersandar pada informasi di luar anatomi.

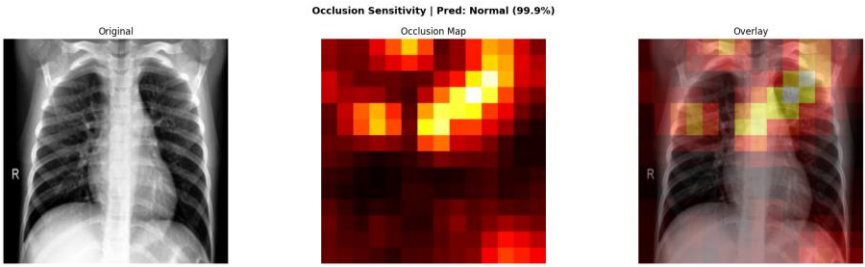
Interpretasi menggunakan LIME diterapkan pada sebuah citra yang diprediksi sebagai Bacterial *Pneumonia* dengan keyakinan 87,9 persen. Hasilnya disajikan pada Gambar 10.12.



Gambar 10.12 Hasil Interpretasi LIME pada Citra Bacterial *Pneumonia*

Segmen yang ditandai sebagai kontributor positif sebagian besar berada pada area lapangan paru, yaitu lokasi terjadinya konsolidasi pada *pneumonia* bakteri. Visualisasi kombinasi kontribusi positif dan negatif memperlihatkan bahwa tidak seluruh area citra mendukung keputusan model secara sama; beberapa bagian justru memberikan bukti yang berlawanan. Model memanfaatkan kombinasi beberapa fitur lokal, bukan bergantung pada satu area tunggal. Perlu dicatat bahwa hasil LIME bersifat stokastik sehingga penjelasannya dapat sedikit bervariasi apabila proses diulang.

Analisis *Occlusion Sensitivity* diterapkan pada citra kelas Normal dengan keyakinan prediksi 99,9 persen, dan hasilnya disajikan pada Gambar 10.13.



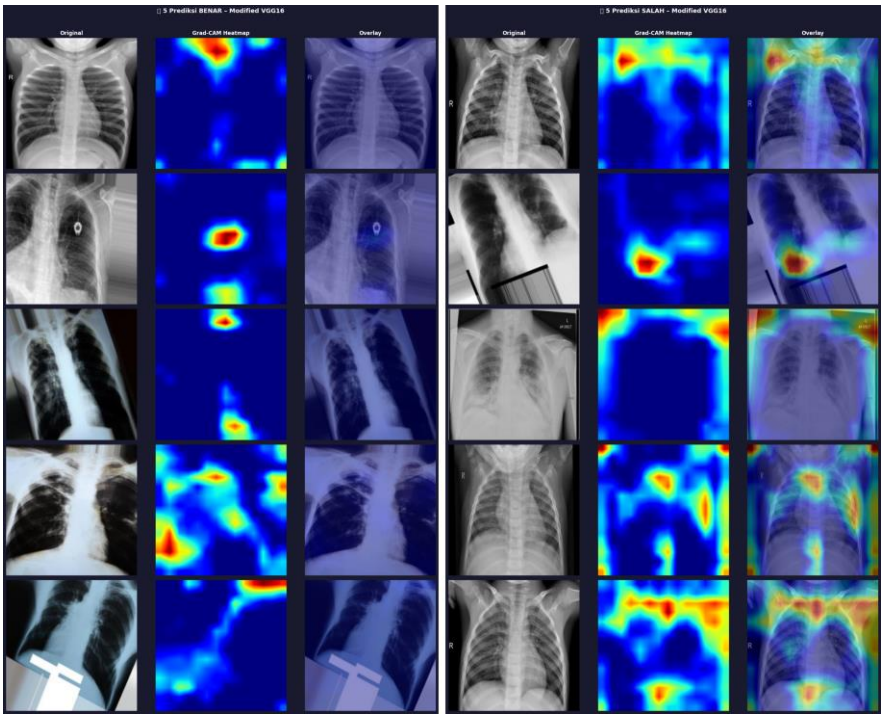
Gambar 10.13 Hasil Analisis *Occlusion Sensitivity* pada Citra Normal

Area dengan intensitas warna lebih terang menunjukkan bagian citra yang paling memengaruhi keputusan model. Hasilnya memperlihatkan bahwa area tengah paru memiliki pengaruh tertinggi: ketika bagian tersebut ditutup, keyakinan model berubah

secara berarti. Hal ini konsisten dengan penalaran medis, sebab penilaian kondisi normal memang didasarkan pada tampilan lapangan paru yang bersih tanpa opasitas abnormal. Konsistensi tersebut sejalan pula dengan performa kuantitatif kelas Normal yang memiliki *recall* tertinggi di antara seluruh kelas.

Keyakinan Model pada Prediksi yang Salah

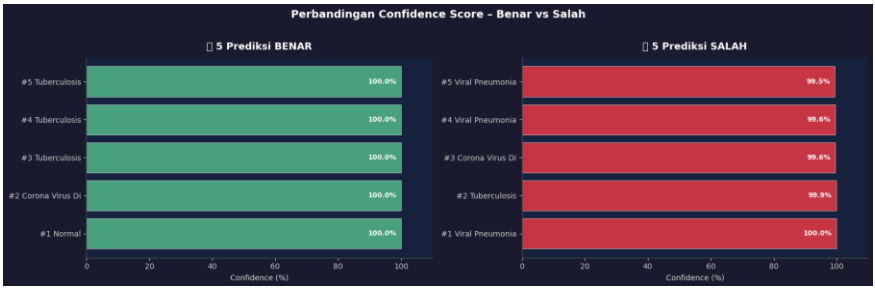
Pemeriksaan terakhir dilakukan terhadap lima prediksi dengan keyakinan tertinggi yang benar dan lima prediksi dengan keyakinan tertinggi yang salah, beserta *heatmap Grad*-CAM masing-masing, sebagaimana disajikan pada Gambar 10.14.



Gambar 10.14 Lima Prediksi Benar dan Lima Prediksi Salah dengan Keyakinan Tertinggi

Pada kelompok prediksi benar, model menghasilkan keyakinan 100 persen pada seluruh sampel, dengan pola aktivasi *Grad*-CAM yang relatif konsisten. Kelas yang mendominasi kelompok ini adalah Tuberculosis, kelas tersebut memiliki *precision* tertinggi sebesar 0,9748.

Pada kelompok prediksi salah, model tetap menghasilkan keyakinan antara 99,5 hingga 100 persen meskipun prediksinya tidak sesuai dengan label sebenarnya. Kesalahan yang paling dominan adalah Viral *Pneumonia* yang diprediksi sebagai Normal, disusul kekeliruan antara *Tuberculosis* dan *Corona Virus Disease*. Perbandingan keyakinan kedua kelompok disajikan pada Gambar 10.15.



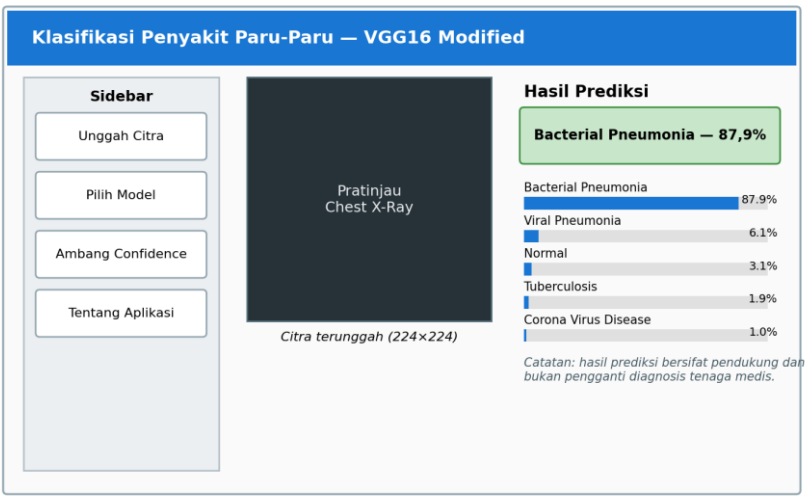
Gambar 10.15 Perbandingan Keyakinan pada Prediksi Benar dan Prediksi Salah

Perhatian: Keyakinan 99,9 Persen pada Prediksi yang Keliru
Gambar 10.15 memperlihatkan sesuatu yang lebih meresahkan daripada tingginya angka keyakinan itu sendiri, yaitu betapa tipisnya perbedaan antara kedua kelompok. Prediksi benar berkeyakinan 100 persen; prediksi salah berkeyakinan 99,5 sampai 100 persen. Selisih di antara keduanya berada pada kisaran setengah poin persentase.
Konsekuensi praktisnya langsung. Nilai keyakinan tidak dapat digunakan untuk memisahkan prediksi yang dapat dipercaya dari yang tidak. Rancangan sistem yang lazim diterapkan menerima prediksi berkeyakinan tinggi secara otomatis dan meneruskan prediksi berkeyakinan rendah kepada tenaga medis tidak akan bekerja pada model ini, sebab hampir seluruh prediksi berada pada kategori berkeyakinan tinggi, termasuk yang keliru.
Penyebabnya telah disinggung pada Subbab 10.2.2. Fungsi kerugian *categorical crossentropy* terus mendorong model menaikkan probabilitas kelas yang benar mendekati satu sepanjang pelatihan, dan tidak ada mekanisme apa pun pada rancangan ini yang mengendalikan kalibrasi. Model dengan

akurasi 85,75 persen yang menghasilkan keyakinan mendekati 100 persen memiliki Expected Calibration Error yang besar. Perbaikan yang tersedia relatif murah. Temperature scaling membagi nilai logit dengan sebuah parameter suhu yang dioptimasi pada data validasi sebelum softmax diterapkan; teknik ini tidak menuntut pelatihan ulang dan tidak mengubah peringkat kelas sehingga akurasi tidak terpengaruh sama sekali. Label smoothing dan Monte Carlo *Dropout* merupakan alternatif lain yang dapat dipertimbangkan.

Implementasi Sistem Berbasis *Streamlit*

Model modifikasi diterapkan pada sebuah aplikasi berbasis *Streamlit* agar dapat digunakan secara praktis. Rancangan antarmukanya disajikan pada Gambar 10.16.



Gambar 10.16 Rancangan Antarmuka Aplikasi Prediksi Berbasis *Streamlit*

Alur kerjanya sederhana: pengguna mengunggah citra *chest X-ray*, aplikasi menjalankan penyeragaman ukuran dan normalisasi yang identik dengan tahap pelatihan, model melakukan inferensi, dan hasilnya ditampilkan berupa kelas dengan probabilitas tertinggi beserta distribusi probabilitas seluruh kelas dalam bentuk diagram batang.

Keputusan menampilkan distribusi probabilitas kelima kelas, dan bukan hanya kelas terpilih, diambil sebagai tanggapan langsung pada Subbab 10.4.9. Penyajian semacam itu memungkinkan pengguna mengetahui apakah model benar-benar yakin pada satu kelas atau sesungguhnya ragu di antara beberapa kelas. Meskipun demikian, mengenai *overconfidence* memperlihatkan bahwa manfaat penyajian ini terbatas selama kalibrasi belum diperbaiki, sebab distribusi yang ditampilkan pun cenderung berbentuk ekstrem.

Pembahasan dan Keterbatasan

Hasil pada bab ini perlu dibaca bersama sejumlah keterbatasan berikut, yang dikelompokkan menurut tahapannya.

Keterbatasan pada data:

a. Data berasal dari satu sumber, sehingga kemampuan generalisasi terhadap citra dari rumah sakit maupun perangkat pencitraan lain belum diketahui.
b. Karakteristik teknis citra tidak diperiksa per kelas, padahal pemeriksaan tersebut diperlukan untuk menutup kemungkinan *shortcut learning* yang terindikasi pada Subbab 10.4.8.
c. Label diterima sebagaimana adanya tanpa verifikasi ulang oleh ahli radiologi, sehingga sebagian kesalahan model dapat berasal dari label yang keliru.

Keterbatasan pada prapemrosesan dan pemodelan:

a. Augmentasi horizontal flip membangkitkan citra yang secara anatomis tidak realistis, sebagaimana diuraikan pada Subbab 10.3.4.
b. *Convolutional base* dibekukan seluruhnya, sehingga potensi perbaikan melalui *fine tuning* parsial pada blok konvolusi akhir belum dieksplorasi.
c. *Hyperparameter* ditetapkan secara tetap tanpa proses penalaan sistematis.
d. Pelatihan dilakukan satu kali untuk setiap model tanpa pengulangan dengan seed acak yang berbeda, sehingga variabilitas hasil akibat inisialisasi acak belum terukur dan selisih lima poin belum dapat dipastikan berada di luar rentang keacakan.

Keterbatasan pada evaluasi dan sistem:

a. *Recall* kelas Viral *Pneumonia* hanya 0,5556, sehingga hampir separuh kasus *pneumonia* virus tidak dikenali.
b. Aturan keputusan menggunakan probabilitas tertinggi tanpa penalaan ambang maupun matriks biaya, padahal biaya kekeliruan antarkelas jelas tidak setara.
c. Model belum terkalibrasi, sehingga nilai keyakinan tidak dapat digunakan sebagai dasar penyaringan.
d. Aplikasi berupa prototipe yang belum diuji pada alur kerja nyata bersama tenaga medis.

Arah pengembangan berikut disusun menurut perbandingan antara dampak dan biayanya. Kelompok pertama tidak menuntut pelatihan ulang maupun data tambahan: menerapkan temperature scaling menggunakan data validasi untuk memperbaiki kalibrasi, menurunkan ambang keputusan bagi kelas Viral *Pneumonia* agar *recall*-nya meningkat, dan menandai prediksi yang berada di dekat batas keputusan sebagai kasus yang memerlukan verifikasi manusia.

Kelompok kedua menuntut pelatihan ulang namun tidak menuntut data tambahan: menerapkan segmentasi lapangan paru sebagai tahap prapemrosesan untuk menutup kemungkinan *shortcut learning*, mengganti fungsi kerugian menjadi focal loss atau menambahkan bobot kelas agar Viral *Pneumonia* memperoleh perhatian lebih besar, membuka blok konvolusi terakhir untuk *fine tuning* parsial dengan learning rate yang sangat kecil, serta mengulang seluruh percobaan dengan beberapa seed acak untuk mengukur variabilitas hasil.

Kelompok ketiga menuntut upaya yang lebih besar: memperluas data melalui kerja sama dengan beberapa fasilitas layanan kesehatan agar variasi perangkat pencitraan terwakili, melakukan verifikasi label bersama ahli radiologi, memvalidasi kesesuaian antara area aktivasi model dan anotasi ahli secara kuantitatif, serta menguji sistem pada alur kerja nyata untuk mengukur kegunaan praktisnya.

➢ Ringkasan

Bab ini telah menguraikan pembangunan model klasifikasi lima kondisi paru-paru pada citra *chest X-ray* menggunakan *VGG16* dengan dua rancangan lapisan klasifikasi yang dibandingkan secara terkendali, beserta analisis interpretabilitas dan penerapannya pada sistem berbasis *Streamlit*. Beberapa hal penting dapat dirangkum sebagai berikut.

a. Modifikasi lapisan klasifikasi melalui penambahan dense layer, *batch normalization*, dan *dropout* menaikkan akurasi dari 0,808 menjadi 0,858, atau setara dengan reduksi kesalahan sebesar 25,81 persen, sementara seluruh faktor lain dikendalikan agar identik.
b. Karena *convolutional base* kedua model identik dan dibekukan, peningkatan tersebut sepenuhnya berasal dari kemampuan classifier memanfaatkan fitur yang sama secara lebih efektif, bukan dari perbaikan kualitas fitur.
c. Akurasi 85,75 persen merupakan rata-rata yang menyembunyikan sebaran yang lebar: *F1-score* antarkelas berkisar dari 0,6522 pada Viral *Pneumonia* sampai 0,9628 pada Corona Virus Disease.
d. *Recall* kelas Viral *Pneumonia* hanya 0,5556, sehingga hampir separuh kasus *pneumonia* virus tidak dikenali, sementara pada kelas Normal *recall* mencapai 0,9936 model bekerja paling baik pada kelas yang paling tidak memerlukan bantuan.
e. Modifikasi menggeser arah kesalahan, bukan sekadar menguranginya: kesalahan menuju kelas Normal turun dari 39 menjadi 22 citra sedangkan kesalahan menuju Bacterial *Pneumonia* naik dari 42 menjadi 50 citra, yaitu pergeseran yang menguntungkan secara klinis namun tidak tercermin pada akurasi.
f. Model menghasilkan keyakinan 99,5 sampai 100 persen pada prediksi yang salah, yaitu setara dengan keyakinannya pada prediksi yang benar, sehingga nilai keyakinan tidak dapat digunakan sebagai dasar penyaringan.
g. Visualisasi *Grad*-CAM memperlihatkan aktivasi pada area di luar struktur anatomi paru pada sebagian sampel, yang mengindikasikan kemungkinan *shortcut learning* dan

menjadikan akurasi pada data uji belum dapat diperlakukan sebagai taksiran performa di lapangan.

Gagasan yang paling penting untuk dibawa dari bab ini adalah bahwa sebuah angka akurasi merupakan titik awal evaluasi, bukan titik akhirnya. Apabila Metodologi ini berhenti pada angka 85,75 persen, seluruh kesimpulannya akan berbunyi bahwa modifikasi arsitektur berhasil dan model siap digunakan. Justru karena angka itu diurai menjadi metrik per kelas, dibaca sampai ke arah kesalahannya, diperiksa tingkat kalibrasinya, dan ditelusuri dasar keputusannya melalui interpretasi visual, tiga persoalan yang sebelumnya tidak terlihat menjadi terlihat.

Ketiga persoalan tersebut tidak mengurangi nilai bahwa modifikasi memang meningkatkan performa. Yang dilakukannya adalah menempatkan pada konteks yang tepat: model ini lebih baik daripada pembandingnya, tetapi belum layak digunakan sebagai alat diagnosis mandiri. Kedua pernyataan tersebut dapat benar secara bersamaan, dan kemampuan memegang keduanya sekaligus merupakan keterampilan yang perlu dilatih.

➢ Daftar Pustaka

Chollet, F. (2021). Deep Learning with Python (2nd ed.). Shelter Island: Manning Publications.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). *ImageNet*: A Large-Scale Hierarchical Image Database. Dalam IEEE Conference on Computer Vision and Pattern Recognition, hlm. 248–255.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., & Wichmann, F. A. (2020). Shortcut Learning in Deep Neural Networks. Nature Machine Intelligence, 2(11), 665–673.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. Cambridge: MIT Press.

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. Dalam Proceedings of the 34th International Conference on Machine Learning, hlm. 1321–1330.

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network *Training* by Reducing *Internal Covariate Shift*. Dalam Proceedings of the 32nd International Conference on Machine Learning, hlm. 448–456.

Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C. S., Liang, H., Baxter, S. L., dkk. (2018). Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. Cell, 172(5), 1122–1131.

Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. Dalam International Conference on Learning Representations.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. Nature, 521(7553), 436–444.

Lin, M., Chen, Q., & Yan, S. (2014). Network in Network. Dalam International Conference on Learning Representations.

Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345–1359.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why Should I Trust You? Explaining the Predictions of Any Classifier. Dalam Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, hlm. 1135–1144.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. Dalam IEEE International Conference on Computer Vision, hlm. 618–626.

Shorten, C., & Khoshgoftaar, T. M. (2019). A Survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1), 1–48.

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. Dalam International Conference on Learning Representations.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from *Overfitting*. Journal of Machine Learning Research, 15(1), 1929–1958.

Varoquaux, G., & Colliot, O. (2023). Evaluating Machine Learning Models and Their Diagnostic Value. Dalam Machine Learning for Brain Disorders. New York: Springer.

World Health Organization. (2023). Global Tuberculosis Report. Jenewa: WHO.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How Transferable Are Features in Deep Neural Networks? Dalam Advances in Neural Information Processing Systems, 27, 3320–3328.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. Dalam European Conference on Computer Vision, hlm. 818–833.

BIODATA PENULIS



I Gede Susrama Mas Diyasa

Penulis merupakan akademisi dan peneliti di bidang Ilmu Komputer yang aktif mengembangkan penerapan kecerdasan buatan pada citra medis. Menyelesaikan pendidikan S1 di Institut Teknologi Adhi Tama Surabaya serta S2 dan S3 di Institut Teknologi Sepuluh Nopember. Saat ini mengajar pada Program Studi Sains Data dan Program Magister Teknologi Informasi, Fakultas Ilmu Komputer, UPN "Veteran" Jawa Timur, serta pernah menjabat sebagai Kepala Lembaga Sertifikasi Profesi UPN "Veteran" Jawa Timur. Bidang kepakarannya meliputi *image processing*, biomedik, dan *computer vision*, dengan berbagai penelitian penerapan *Convolutional Neural Network* untuk analisis citra kesehatan. Pengalaman akademik dan penelitiannya memperkaya pembahasan buku ini, khususnya pada aspek pengolahan citra medis, arsitektur *deep learning*, dan penerapannya dalam sistem pendukung diagnosis.



Adhi Dharma Wibawa

Penulis merupakan Guru Besar pada Institut Teknologi Sepuluh Nopember (ITS) yang saat ini menjabat sebagai Kepala Pusat Studi Kecerdasan Artifisial dan Teknologi Digital ITS. Menyelesaikan pendidikan S1 dan S2 Teknik Elektro di ITS, kemudian meraih gelar doktor bidang *Biomedical Engineering* di W.J. Kolff Institute, University of Groningen, Belanda, dengan disertasi mengenai pemodelan sistem muskuloskeletal manusia. Aktif mengajar pada Departemen Teknologi Kedokteran dan Teknik Biomedik, serta berkontribusi dalam organisasi profesi APTIKOM dan IEEE. Bidang kepakarannya meliputi pemrosesan sinyal biomedik, biomekanika, kecerdasan artifisial, dan pengolahan citra. Perspektif keilmuannya memberikan landasan yang kuat bagi pembahasan buku ini pada aspek pemrosesan sinyal dan citra biomedis serta pemodelan komputasional dalam bidang kesehatan.



Khairul Anam

Penulis merupakan Guru Besar pada Departemen Teknik Elektro, Universitas Jember. Menyelesaikan pendidikan S1 Teknik Elektro di Universitas Brawijaya, S2 Teknik Elektro bidang Sistem Pengaturan di Institut Teknologi Sepuluh Nopember, serta meraih gelar Ph.D. di University of Technology Sydney, Australia, pada bidang *rehabilitation robotics*. Saat ini menjabat sebagai Koordinator Program Studi Magister Teknik Elektro Universitas Jember, dan pernah menjadi Sekretatisl Bidang Penelitian LP2M Universitas Jember. Bidang kepakarannya mencakup kecerdasan artifisial, sistem kendali, robotika, serta pengenalan pola sinyal mioelektrik dan sinyal otak untuk pengembangan alat bantu bagi penyandang disabilitas. Keahliannya memperkaya buku ini pada aspek pembelajaran mesin dan pengenalan pola pada data biomedis.



Anindita Septiarini

Penulis merupakan Guru Besar pada Program Studi Informatika, Universitas Mulawarman, Samarinda. Menyelesaikan pendidikan S1 Teknik Informatika di Universitas Surabaya, serta S2 dan S3 Ilmu Komputer di Universitas Gadjah Mada. Bidang kepakarannya meliputi kecerdasan artifisial, pengolahan citra, *computer vision*, dan pengenalan pola, dengan lebih dari seratus publikasi ilmiah serta sejumlah Hak Kekayaan Intelektual yang telah diperoleh. Rekam jejak penelitiannya yang panjang pada segmentasi dan klasifikasi citra memberikan kontribusi penting dalam penyusunan buku ini, khususnya pada pembahasan metode ekstraksi fitur, segmentasi citra medis, serta evaluasi kinerja model klasifikasi.



Yessy Andriani Fauziah

Penulis merupakan dokter gigi sekaligus akademisi yang saat ini menjadi staf pengajar pada Fakultas Kedokteran Gigi Universitas Ciputra, Surabaya. Menyelesaikan pendidikan dokter gigi di Fakultas Kedokteran Gigi Universitas Airlangga dan Magister Ilmu Forensik di universitas yang sama, serta menyelesaikan pendidikan Sarjana Ilmu Hukum di Universitas 45 Surabaya. Saat ini sedang menempuh Program Doktor Ilmu Kedokteran Gigi Universitas Airlangga. Kajiannya banyak menyoroti pemanfaatan teknologi digital dalam praktik kedokteran gigi, termasuk potensi kecerdasan buatan untuk deteksi dini penyakit serta aspek etik dan legal penerapannya. Perspektif klinis yang dibawanya melengkapi buku ini dengan sudut pandang praktisi kesehatan mengenai kelayakan, keamanan, dan tanggung jawab profesional dalam penerapan sistem berbasis kecerdasan buatan di lingkungan medis.