

TUGAS AKHIR - ET234801

**DETEKSI DINI KANKER SERVIKS MENGGUNAKAN
MODEL BERBASIS CNN DENGAN *QUANTIZATION* DAN
*EXPLAINABLE AI***

Naufan Zaki Luqmanulhakim

NRP 5027221065

Dosen Pembimbing

Dr. Rizka Wakhidatus Sholikhah, S.Kom

NIP 1993202012054

Irzal Ahmad Sabilla, S. Kom., M.Kom.

NIP 2022199611028

Program Studi Strata 1 (S1) Teknologi Informasi

Departemen Teknologi Informasi

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2026



TUGAS AKHIR - ET234801

**DETEKSI DINI KANKER SERVIKS MENGGUNAKAN
MODEL BERBASIS CNN DENGAN *QUANTIZATION* DAN
*EXPLAINABLE AI***

NAUFAN ZAKI LUQMANULHAKIM
NRP 5027221065

Dosen Pembimbing

Dr. Rizka Wakhidatus Sholikhah, S.Kom
NIP 1993202012054

Irzal Ahmad Sabilla, S.Kom., M.Kom.
NIP 2022199611028

Program Studi Strata 1 (S1) Teknologi Informasi

Departemen Teknologi Informasi

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2026



FINAL PROJECT - ET234801

EARLY DETECTION OF CERVICAL CANCER USING A CNN-BASED MODEL WITH QUANTIZATION AND *EXPLAINABLE AI*

NAUFAN ZAKI LUQMANULHAKIM
NRP 5027221065

Advisor

Dr. Rizka Wakhidatus Sholikah, S.Kom
NIP 1993202012054

Irzal Ahmad Sabilla, S.Kom., M.Kom.
NIP 2022199611028

Undergraduate Study Program of Information Technology

Departemen of Information Technology

Faculty of Intelligent Electrical and Informatics Technology

Institut Teknologi Sepuluh Nopember

Surabaya

2026

LEMBAR PENGESAHAN

DETEKSI DINI KANKER SERVIKS MENGGUNAKAN MODEL BERBASIS CNN DENGAN *QUANTIZATION* DAN *EXPLAINABLE AI*

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Komputer pada
Program Studi S-1 Teknologi Informasi
Departemen Teknologi Informasi
Fakultas Teknik Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh : **NAUFAN ZAKI LUQMANULHAKIM**

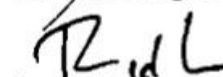
NRP. 5027221065

Disetujui oleh Tim Penguji Tugas Akhir:

1. Dr. Rizka Wakhidatus Sholikhah, S.Kom
NIP: 1993202012054
2. Irzal Ahmad Sabilla, S.Kom., M.Kom.
NIP: 2022199611028
3. Ridho Rahman Hariadi, S.Kom., M.Sc.
NIP: 198702132014041001
4. Hafara Firdausi, S.Kom., M.Kom.
NIP: 2022199812030


Pembimbing I


Pembimbing II


Penguji I


Penguji II

SURABAYA

Juli, 2026

[Halaman ini sengaja dikosongkan]

APPROVAL SHEET

EARLY DETECTION OF CERVICAL CANCER USING A CNN-BASED MODEL WITH QUANTIZATION AND EXPLAINABLE AI

FINAL PROJECT

Submitted to fulfill one of the requirements

For obtaining a Bachelor of Computer Science degree in the

Undergraduate Study Program of Information Technology

Department of Information Technology

Faculty of Intelligent Electrical and Informatics Technology

Institut Teknologi Sepuluh Nopember

By : **NAUFAN ZAKI LUQMANULHAKIM**

NRP. 5027221065

Approved by the Final Project Review Team :

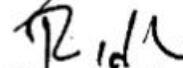
1. Dr. Rizka Wakhidatus Sholikhah, S.Kom
NIP: 1993202012054


Advisor I

2. Irzal Ahmad Sabilla, S.Kom., M.Kom.
NIP: 2022199611028


Advisor II

3. Ridho Rahman Hariadi, S.Kom., M.Sc.
NIP: 198702132014041001


Examiner I

4. Hafara Firdausi, S.Kom., M.Kom.
NIP: 2022199812030


Examiner II

SURABAYA

July, 2026

[Halaman ini sengaja dikosongkan]

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP : Naufan Zaki Luqmanulhakim / 5027221065
Program studi : S-1 Teknologi Informasi
Dosen Pembimbing / NIP : 1. Dr. Rizka Wakhidatus Sholikhah, S.Kom /
1993202012054
2. Irzal Ahmad Sabilla, S. Kom., M.Kom /
2022199611028

dengan ini menyatakan bahwa Tugas Akhir dengan judul “Deteksi Dini Kanker Serviks Menggunakan Model Berbasis CNN dengan *Quantization* dan *Explainable AI*” adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

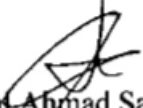
Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, Juli 2026

Mengetahui
Dosen Pembimbing


Dr. Rizka Wakhidatus
Sholikhah, S.Kom
NIP. 1993202012054

Mengetahui
Dosen Pembimbing


Irzal Ahmad Sabilla, S.
Kom., M.Kom
NIP. 2022199611028

Mahasiswa


Naufan Zaki Luqmanulhakim
NRP. 5027221065

[Halaman ini sengaja dikosongkan]

STATEMENT OF ORIGINALITY

The undersigned below:

Name of Student / NRP : Naufan Zaki Luqmanulhakim / 5027221065
Department : S-1 Teknologi Informasi
Advisor / NIP : 1. Dr. Rizka Wakhidatus Sholikah, S.Kom /
1993202012054
2. Irzal Ahmad Sabilla, S. Kom., M.Kom /
2022199611028

hereby declare that the Final Project with the title of "Early Detection of Cervical Cancer Using a CNN-Based Model with Quantization and Explainable AI" is the result of my work, is original, and is written by following the rules of scientific writing.

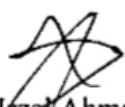
If in the future there is a discrepancy with this statement, then I am willing to accept sanctions in accordance with the provisions that apply at Institut Teknologi Sepuluh Nopember.

Surabaya, July 2026

Acknowledge
Advisor


Dr. Rizka Wakhidatus
Sholikah, S.Kom
NIP. 1993202012054

Acknowledge
Advisor


Irzal Ahmad Sabilla, S.
Kom., M.Kom
NIP. 2022199611028

Student


Naufan Zaki Luqmanulhakim
NRP. 5027221065

[Halaman ini sengaja dikosongkan]

ABSTRAK

Deteksi Dini Kanker Serviks Menggunakan Model Berbasis CNN dengan *Quantization* dan *Explainable AI*

Nama Mahasiswa / NRP : Naufan Zaki Luqmanulhakim / 5027221065
Departemen : Teknologi Informasi FTEIC - ITS
Dosen Pembimbing : Dr. Rizka Wakhidatus Sholikhah, S.Kom
Irzal Ahmad Sabilla, S.Kom., M.Kom.

Abstrak

Kanker serviks merupakan salah satu penyebab utama kematian pada perempuan, khususnya di wilayah dengan keterbatasan akses layanan kesehatan, sementara metode skrining Inspeksi Visual Asam Asetat (IVA) masih bergantung pada subjektivitas tenaga medis dalam proses interpretasi. Penelitian ini bertujuan mengembangkan sistem deteksi dini kanker serviks berbasis citra tes IVA menggunakan pendekatan *Convolutional Neural Network* (CNN) yang diimplementasikan secara *offline* melalui mekanisme *on-device inference* pada perangkat Android. Tiga arsitektur CNN berbasis *transfer learning*, yaitu ResNet-50, EfficientNet-B0, dan MobileNetV3-Small dievaluasi secara komparatif melintasi sembilan skenario *ablation study* strategi *preprocessing*. Guna menekan bahaya klinis *shortcut learning* akibat efek *Clever Hans* dari elemen non-biologis latar belakang, diterapkan teknik pemotongan otomatis berbasis kontur (*OpenCV Auto-Crop*). Model dioptimasi menggunakan *Post-Training Quantization* (PTQ) ke format TensorFlow Lite INT8 untuk meningkatkan efisiensi komputasi pada perangkat bergerak.

Hasil eksperimen menunjukkan bahwa kombinasi *OpenCV Auto-Crop* (skenario NB-6) dengan arsitektur ResNet-50 INT8 merupakan model final terbaik, berhasil mencapai keseimbangan performa optimal dengan nilai PR-AUC sebesar 0,759, sensitivitas (*recall*) 89%, spesifisitas 56%, serta akurasi keseluruhan sebesar 75,7% pada *hold-out test set*. Model ini secara signifikan mengungguli model *baseline machine learning* tradisional (*handcrafted features* + Naive Bayes) yang hanya meraih akurasi total 44,8%. Optimasi kuantisasi berhasil memangkas ukuran model menjadi 23,12 MB dengan rata-rata waktu inferensi lokal yang sangat responsif, yaitu 110,84 ms pada perangkat Android. Model terbaik ini diintegrasikan dengan metode *Explainable AI* (XAI) berbasis Grad-CAM untuk memvisualisasikan area patologis (*acetowhite*) secara transparan guna memvalidasi keputusan klinis. Hasil pengujian *User Acceptance Testing* (UAT) dan validasi klinis lapangan bersama 4 dokter spesialis dan residen Obstetri dan Ginekologi di RSUP Hasan Sadikin Bandung mengonfirmasi kelayakan prototipe aplikasi CervicalAI sebagai alat bantu skrining lapis pertama yang portabel dan independen dari konektivitas internet.

Kata Kunci: *Convolutional Neural Network, Explainable AI, Grad-CAM, Kanker Serviks, OpenCV Auto-Crop, Post-Training Quantization, Tes IVA.*

[Halaman ini sengaja dikosongkan]

ABSTRACT

Early Detection of Cervical Cancer Using a CNN-Based Model with Quantization and Explainable AI

Student Name / NRP : Naufan Zaki Luqmanulhakim / 5027221065
Department : Teknologi Informasi FTEIC - ITS
Advisor : Dr. Rizka Wakhidatus Sholikah, S.Kom
Irzal Ahmad Sabilla, S.Kom., M.Kom.

Abstract

Cervical cancer remains one of the leading causes of cancer-related mortality among women, particularly in regions with limited access to healthcare services, while Visual Inspection with Acetic Acid (VIA) screening still relies heavily on subjective interpretation by medical personnel. This research aims to develop an early detection system for cervical cancer based on VIA images using a Convolutional Neural Network (CNN) implemented through *offline* on-device inference on Android devices. Three *transfer learning*-based CNN architectures, namely ResNet-50, EfficientNet-B0, and MobileNetV3-Small, were comparatively evaluated across nine ablation study scenarios of preprocessing strategies. To mitigate the clinical risks of *shortcut learning* induced by the Clever Hans effect from non-biological background artifacts, an *automated* contour-based *cropping* technique (OpenCV *Auto-Crop*) was implemented. The models were optimized using Post-Training Quantization (PTQ) into TensorFlow Lite INT8 format to enhance computational efficiency on *mobile* devices.

Experimental results demonstrate that the combination of OpenCV *Auto-Crop* (scenario NB-6) and the ResNet-50 INT8 architecture emerged as the optimal final model, achieving a balanced peak performance with a PR-AUC of 0.759, sensitivity (*recall*) of 89%, specificity of 56%, and an overall accuracy of 75.7% on the *hold-out test* set. This model significantly outperformed the traditional *machine learning baseline* (handcrafted features + Naive Bayes), which only achieved an overall accuracy of 44.8%. The quantization optimization successfully compressed the model size to 23.12 MB while delivering a highly responsive local inference latency of 110.84 ms on the Android device. The selected model was further integrated with a Grad-CAM-based Explainable AI (XAI) method to transparently visualize pathological (*acetowhite*) regions for clinical decision validation. User Acceptance Testing (UAT) and field clinical validation conducted with 4 Obstetrics and Gynecology specialists and residents at RSUP Hasan Sadikin Bandung confirmed the feasibility of the CervicalAI prototype as a portable, *offline*-capable first-line screening decision support tool.

Keywords: Cervical Cancer, *Convolutional Neural Network*, *Explainable AI*, Grad-CAM, OpenCV *Auto-Crop*, *Post-Training Quantization*, VIA Test.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT, atas rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“Deteksi Dini Kanker Serviks Menggunakan Model Berbasis CNN dengan *Quantization* dan *Explainable AI*”**. Tugas Akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Departemen Teknologi Informasi. Penulis berharap karya ini dapat memberikan kontribusi nyata terhadap pengembangan teknologi informasi, khususnya dalam deteksi kanker serviks.

1. Dalam proses penyusunan dan penyelesaian Tugas Akhir ini, penulis telah mendapatkan banyak dukungan, bantuan, dan doa dari berbagai pihak, baik secara langsung maupun tidak langsung. Oleh karena itu, pada kesempatan ini, penulis ingin menyampaikan ucapan terima kasih kepada:
2. Ayah dan Ibu tercinta, yang selalu memberikan doa, dukungan moral, dan modal finansial tiada henti dalam setiap langkah perjalanan pendidikan penulis.
3. Ibu Dr. Rizka Wakhidatus Sholikah, S.Kom dan Bapak Irzal Ahmad Sabilla, S. Kom., M.Kom., selaku dosen pembimbing I dan II yang telah membimbing, memberikan arahan, masukan berharga, serta meluangkan waktu dan perhatian dalam proses penyusunan Tugas Akhir ini.
4. Seluruh dosen dan staf Departemen Teknologi Informasi, yang telah memberikan ilmu dan pengalaman selama masa studi yang sangat berarti bagi perkembangan akademik dan pribadi penulis
5. Rekan-rekan mahasiswa angkatan 2022 Departemen Teknologi Informasi, khususnya teman seperjuangan Excelsior, atas kebersamaan, semangat, dan dukungan selama menempuh pendidikan di kampus tercinta.
6. dr. Dodi Suardi, Sp. OG (K) Onk selaku narasumber dan mentor yang telah membantu banyak dalam penelitian ini.
7. Semua dokter Poli Onkologi Rumah Sakit Umum Daerah Hasan Sadikin.

Akhir kata, penulis harap Tugas Akhir ini dapat memberikan manfaat bagi pembaca serta menjadi salah satu bentuk sumbangan penulis terhadap perkembangan ilmu pengetahuan, khusus nya di bidang teknologi informasi. Penulis menyadari bahwa karya ini masih jauh dari kata sempurna, oleh karena itu saran dan kritik yang membangun sangat diharapkan untuk perbaikan di masa mendatang.

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

LEMBAR PENGESAHAN	i
APPROVAL SHEET	iii
PERNYATAAN ORISINALITAS	v
STATEMENT OF ORIGINALITY	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR GAMBAR	xviii
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER	xxiii
BAB 1 PENDAHULUAN	25
1.1 Latar Belakang	25
1.2 Rumusan Masalah	25
1.3 Batasan Masalah	26
1.4 Tujuan	26
1.5 Manfaat	26
BAB 2 TINJAUAN PUSTAKA	29
2.1 Hasil Penelitian Terdahulu	29
2.2 Dasar Teori	32
2.2.1 Kanker Serviks	32
2.2.2 Tes IVA	32
2.2.3 <i>Deep learning</i>	33
2.2.4 <i>Convolutional Neural Network</i>	33
2.2.5 Arsitektur MobileNet V3	34
2.2.6 Arsitektur ResNet	34
2.2.7 Arsitektur EfficientNet-B0	35
2.2.8 Model <i>Quantization</i>	36
2.2.9 <i>Explainable AI</i>	36
2.2.10 Metrik Evaluasi	37
BAB 3 METODOLOGI PENELITIAN	41
3.1 Diagram Metodologi	41

3.1.1	Pengumpulan Dataset	41
3.1.2	Pengolahan Dataset.....	43
3.1.3	Pelatihan Model CNN	46
3.1.4	Evaluasi Model.....	50
3.1.5	<i>Heatmap Explainable AI</i>	52
3.1.6	Pengembangan Aplikasi Android.....	54
3.1.7	Pengujian <i>User Acceptance Test</i> dan Validasi Klinis	55
3.2	Bahan dan Peralatan yang Digunakan	56
3.2.1	Perangkat Lunak	56
3.2.2	Perangkat Keras	57
3.3	Implementasi	57
3.3.1	Persiapan Dataset dan Augmentasi Data	57
3.3.2	<i>Model factory</i> dan <i>Transfer learning</i> Dua Fase.....	68
3.3.3	<i>Post-Training Quantization</i> (INT8)	72
3.3.4	Evaluasi Model	73
3.3.5	Implementasi Grad-CAM.....	74
3.3.6	Model <i>Baseline</i> Berbasis <i>Machine learning</i> Tradisional (NB-10).....	74
3.3.7	Implementasi Model <i>Baseline</i>	76
3.3.8	Implementasi Prapemrosesan Android (<i>CervixCropper</i>).....	78
3.3.9	Implementasi pada Aplikasi Android	80
BAB 4 HASIL DAN PEMBAHASAN		89
4.1	Skenario Uji Coba	89
4.1.1	Uji Performa Model.....	89
4.1.2	Uji Pengguna	89
4.2	Hasil Uji Coba	90
4.2.1	Evaluasi Strategi Pra-pemrosesan Citra.....	90
4.2.2	Hasil Kuantisasi (<i>Post-Training Quantization</i> INT8)	105
4.2.3	Evaluasi Model dan Pemilihan Model Final	106
4.2.4	Hasil <i>Explainable AI</i> (Grad-CAM)	117
4.2.5	Evaluasi Model Terbaik dengan Model <i>Baseline</i>	121
4.2.7	Hasil <i>User Acceptance Testing</i> (UAT) dan Validasi Klinis	122
4.2.8	Hasil Uji Validasi Klinis Diagnostik Lapangan	123
4.3	Pembahasan	126
4.3.1	Jawaban terhadap Rumusan Masalah	126

4.3.2 Posisi Temuan terhadap Penelitian Terdahulu.....	127
4.3.3 Keterkaitan Antar-Temuan	128
4.3.4 Implikasi Praktis dan Klinis.....	129
4.3.5 Keterbatasan Penelitian.....	130
BAB 5 KESIMPULAN.....	133
5.1 Kesimpulan.....	133
5.2 Saran	133
DAFTAR PUSTAKA	135
LAMPIRAN.....	139
BIOGRAFI PENULIS	151

DAFTAR GAMBAR

Gambar 2.1 (a) Serviks Sebelum IVA dan (b) Serviks Setelah IVA	33
Gambar 2.2 Arsitektur <i>Convolutional Neural Network</i>	34
Gambar 3.1 Diagram Alir Sistem	41
Gambar 3.2 (a) Serviks Normal dan (b) Terindikasi Kanker Serviks.....	42
Gambar 3.3 (a) Contoh Gambar Datasert IARC WHO dan (b) Contoh Gambar Dataset Kolaborasi Primer.....	43
Gambar 3.4 Skema Pelatihan dan Evaluasi Model	48
Gambar 3.5 Contoh Penerapan <i>Heatmap</i> pada Foto Tes IVA.....	53
Gambar 3.6 Proses Kerja <i>Heatmap</i> Grad-CAM	54
Gambar 3.7 Diagram Alir Sistem Aplikasi CervicalAI dari Pengambilan Gambar hingga Ekspor Laporan Diagnosis	81
Gambar 3.8 Tampilan Halaman Beranda Aplikasi CervicalAI	82
Gambar 3.9 Tampilan Halaman Kamera Kustom dengan <i>Overlay</i> Panduan SSK	83
Gambar 3.10 Tampilan Halaman Hasil Analisis: (a) Pratinjau Citra dengan <i>Overlay Heatmap</i> CAM, (b) Kartu Hasil Diagnosis	84
Gambar 3.11 Tampilan Halaman Pratinjau Gambar Layar Penuh dengan <i>Toggle Heatmap</i>	86
Gambar 4.1 Contoh citra mentah tanpa pra-pemrosesan (NB-1): serviks dikelilingi oleh spekulum logam, bayangan, dan teks klinis	91
Gambar 4.2 Kurva <i>Precision-Recall</i> dan ROC model NB-1 pada dataset uji	91
Gambar 4.3 <i>Dashboard</i> ringkasan performa model NB-7: metrik klinis dan komputasi..	92
Gambar 4.4 Kurva <i>Precision-Recall</i> dan ROC model NB-2 pada dataset uji	93
Gambar 4.5 <i>Dashboard</i> ringkasan performa model NB-2: metrik klinis dan komputasi..	93
Gambar 4.6 Perbandingan citra sebelum dan sesudah pemotongan otomatis OpenCV: (Kiri) citra asli dengan anotasi kontur dan kotak pembatas, (Kanan) hasil <i>auto-crop</i> 224x224	94
Gambar 4.7 Kurva <i>Precision-Recall</i> dan ROC model NB-4 pada dataset uji	94
Gambar 4.8 Kurva <i>Precision-Recall</i> dan ROC model NB-5 pada dataset uji	95
Gambar 4.9 <i>Dashboard</i> ringkasan performa model NB-5: metrik klinis dan komputasi..	95
Gambar 4.10 Contoh hasil masking ROI sirkular dinamis (NB-7): serviks bulat di tengah dengan latar belakang hitam	96
Gambar 4.11 Kurva <i>Precision-Recall</i> dan ROC model NB-7 pada dataset uji	96
Gambar 4.12 <i>Dashboard</i> ringkasan performa model NB-7: metrik klinis dan komputasi	97
Gambar 4.13 Contoh hasil masking ROI sirkular statis (NB-8).....	98
Gambar 4.14 Kurva <i>Precision-Recall</i> dan ROC model NB-8 pada dataset uji (kurva ROC nyaris menyentuh garis acak)	98
Gambar 4.15 <i>Dashboard</i> ringkasan performa model NB-8: metrik klinis dan komputasi	99
Gambar 4.16 Contoh deteksi OWL-ViT pada sampel citra serviks: kotak pembatas, skor kepercayaan, dan hasil <i>crop</i>	100
Gambar 4.17 Kurva <i>Precision-Recall</i> dan ROC model NB-8 pada dataset uji	101
Gambar 4.18 <i>Dashboard</i> ringkasan performa model NB-8: metrik klinis dan komputasi	101
Gambar 4.19 Contoh deteksi YOLOv11 SEVIA pada sampel citra serviks: kotak pembatas, skor kepercayaan, dan hasil <i>crop</i>	102

Gambar 4.20 Kurva <i>Precision-Recall</i> dan ROC model NB-9 pada dataset uji.....	103
Gambar 4.21 <i>Dashboard</i> ringkasan performa model NB-9: metrik klinis dan komputasi	103
Gambar 4.22 Proses cara kerja sistem CBAM.....	104
Gambar 4.23 Kurva <i>Precision-Recall</i> dan ROC model NB-3 pada dataset uji.....	105
Gambar 4.24 <i>Dashboard</i> ringkasan performa model NB-3: metrik klinis dan komputasi	105
Gambar 4.25 Hasil visualisasi Grad-CAM pada model tanpa pra-pemrosesan (NB-7) ..	117
Gambar 4.26 Hasil visualisasi Grad-CAM pada model <i>baseline</i> augmentasi tanpa pemotongan (NB-3).....	117
Gambar 4.27 Hasil visualisasi Grad-CAM pada model dengan mekanisme atensi CBAM (NB-4).....	118
Gambar 4.28 Perbandingan visualisasi Grad-CAM pada metode masking ROI sirkular: (a) masking dinamis (NB-6), (b) masking statis (NB-7)	119
Gambar 4.29 Perbandingan visualisasi Grad-CAM pada metode pemotongan otomatis OpenCV: (a) model NB-4, (b) model pemenang NB-5	119
Gambar 4.30 Perbandingan visualisasi Grad-CAM pada pemotongan berbasis model fondasi: (a) model NB-8 (OWL-ViT), (b) model NB-9 (YOLOv11 SEVIA)	120
Gambar 4.31 Perbandingan visualisasi Grad-CAM lintas arsitektur model pada eksperimen NB-5: (a) ResNet-50, (b) MobileNetV3-Small	121

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	29
Tabel 3.1 Sumber Dataset	42
Tabel 3.2 Parameter Augmentasi Dataset	44
Tabel 3.3 <i>Ablation study</i> Strategi <i>Preprocessing</i> yang Diuji.....	45
Tabel 3.4 Rancangan <i>Unified Classification head</i>	46
Tabel 3.5 Konfigurasi <i>Hyperparameter</i> Lengkap Per Model Per Fase	47
Tabel 3.6 Konfigurasi <i>Callback Training</i>	49
Tabel 3.7 Parameter Evaluasi	50
Tabel 3.8 Parameter User Acceptance Test dan Validasi Klinis	55
Tabel 3.9 Perangkat Lunak	56
Tabel 4.4 Komponen Aplikasi Android CervicalAI	80
Tabel 4.1 Perbandingan Rata-rata Ukuran Model	105
Tabel 4.2 Waktu Inferensi Sebelum dan Sesudah Kuantisasi Model	106
Tabel 4.3 Performa Diagnostik Utama Skenario NB-2	107
Tabel 4.4 Metrik Tambahan dan Efisiensi Skenario NB-2	107
Tabel 4.5 Performa Diagnostik Utama Skenario NB-3	108
Tabel 4.6 Metrik Tambahan dan Efisiensi Skenario NB-3	108
Tabel 4.7 Performa Diagnostik Utama Skenario NB-4	109
Tabel 4.8 Metrik Tambahan dan Efisiensi Skenario NB-4	110
Tabel 4.9 Performa Diagnostik Utama Skenario NB-5	110
Tabel 4.10 Metrik Tambahan dan Efisiensi Skenario NB-5	110
Tabel 4.11 Performa Diagnostik Utama Skenario NB-1	111
Tabel 4.12 Metrik Tambahan dan Efisiensi Skenario NB-1	111
Tabel 4.13 Performa Diagnostik Utama Skenario NB-6	112
Tabel 4.14 Metrik Tambahan dan Efisiensi Skenario NB-6	112
Tabel 4.15 Performa Diagnostik Utama Skenario NB-7	113
Tabel 4.16 Metrik Tambahan dan Efisiensi Skenario NB-7	113
Tabel 4.17 Performa Diagnostik Utama Skenario NB-8	113
Tabel 4.18 Metrik Tambahan dan Efisiensi Skenario NB-8	114
Tabel 4.19 Performa Diagnostik Utama Skenario NB-9	114
Tabel 4.20 Metrik Tambahan dan Efisiensi Skenario NB-9	115
Tabel 4.21 Performa Model pada Skenario NB-10	115
Tabel 4.22 <i>Confusion matrix</i> Model Final (ResNet-50 INT8, Skenario NB-5)	116
Tabel 4.23 Perbandingan Akhir melawan <i>Baseline</i> Tradisional (Target Evaluasi).....	121
Tabel 4.24 Skor Rerata Kuesioner Evaluasi UAT CervicalAI	122
Tabel 4.25 <i>Confusion matrix</i> Uji Validasi Klinis Lapangan (20 Kasus)	124
Tabel 4.26 Performa Diagnostik Lapangan Aplikasi CervicalAI	124
Tabel 4.27 Rata-rata <i>Confidence score</i> Hasil Pengujian Lapangan (Sampel 10 Pasien).	124

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 3.1 Konfigurasi augmentasi data dan fungsi pembentukan generator (NB-2 & NB-3).....	58
Kode Sumber 3.2 Fungsi pemotongan spasial otomatis berbasis kontur OpenCV (NB-4)	60
Kode Sumber 3.3 Generator pelatihan tanpa augmentasi (NB-5)	61
Kode Sumber 3.4 Generator pelatihan kontrol tanpa augmentasi dan <i>cropping</i> (NB-1)...	62
Kode Sumber 3.5 Fungsi masking ROI sirkular berbasis kontur (NB-6).....	62
Kode Sumber 3.6 Fungsi masking ROI statis berbasis pusat citra (NB-7).....	63
Kode Sumber 3.7 Deteksi lesi <i>zero-shot</i> berbasis OWL-ViT dan fungsi <i>crop</i> (NB-8)	65
Kode Sumber 3.8 Deteksi lesi berbasis SEVIA YOLOv11 dan fungsi <i>crop</i> (NB-9)	67
Kode Sumber 3.9 Fungsi <i>model factory</i> dengan <i>unified classification head</i>	69
Kode Sumber 3.10 Implementasi modul atensi CBAM (NB-3).....	71
Kode Sumber 3.9 Konversi model ke TFLite INT8 dengan konfigurasi <i>dual-output</i>	73
Kode Sumber 3.12 Fungsi evaluasi model TFLite dengan pengukuran latensi per sampel	74
Kode Sumber 3.13 Fungsi Grad-CAM berbasis GradientTape dan ekspor bobot <i>Dense layer</i>	74
Kode Sumber 3.14 Fungsi penghapusan pantulan spekular via Telea Inpainting (NB-10)	75
Kode Sumber 3.15 Deteksi ROI serviks menggunakan Gaussian Mixture Model (NB-10)	76
Kode Sumber 3.16 Fungsi ekstraksi fitur warna dan tekstur handcrafted (NB-10).....	76
Kode Sumber 3.17 Pra-pemrosesan dan Deteksi ROI Model <i>Baseline</i>	77
Kode Sumber 3.18 Ekstraksi, Reduksi, dan Normalisasi Fitur Manual	78
Kode Sumber 3.19 Klasifikasi Komparatif dan Analisis Kepentingan Fitur (SHAP).....	78
Kode Sumber 3.20 Prapemrosesan <i>CervixCropper</i>	79

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Kanker serviks merupakan salah satu penyebab utama kematian pada perempuan di dunia, dengan proporsi kasus dan kematian yang lebih tinggi terjadi di negara berpendapatan rendah dan menengah akibat keterbatasan akses skrining (Bray et al., 2024). Di Indonesia, kanker serviks masih menempati peringkat atas dalam angka insidensi dan mortalitas pada perempuan, sehingga deteksi dini menjadi strategi utama dalam menekan angka kematian. WHO merekomendasikan skrining berbasis HPV DNA sebagai metode utama, namun implementasinya memerlukan infrastruktur laboratorium dan tenaga ahli yang belum merata di seluruh wilayah (Bradford & Goodman, 2013). Oleh karena itu, Inspeksi Visual Asam Asetat (IVA) tetap menjadi metode skrining yang banyak digunakan di fasilitas kesehatan primer karena prosedurnya sederhana, berbiaya rendah, dan memungkinkan hasil diperoleh secara langsung (Harsono et al., 2022).

Akurasi tes IVA sangat bergantung pada kompetensi tenaga kesehatan, sehingga variabilitas interpretasi dapat memengaruhi ketepatan diagnosis lesi pra-kanker. Pemanfaatan citra digital serviks melalui smartphone memungkinkan dokumentasi dan evaluasi yang lebih terstandar, sementara perkembangan *Convolutional Neural Network* (CNN) membuka peluang analisis citra medis secara objektif melalui ekstraksi fitur visual. Namun, sistem berbasis *cloud* kurang sesuai untuk wilayah dengan keterbatasan konektivitas. Oleh karena itu, diperlukan implementasi model yang mampu berjalan secara *offline* melalui mekanisme *on-device inference*, sehingga tetap operasional pada fasilitas layanan kesehatan dengan infrastruktur terbatas (Asgary et al., 2020).

Tantangan utama dalam menjalankan model AI pada perangkat *mobile* adalah keterbatasan spesifikasi dari perangkatnya itu sendiri. Untuk menjalankan suatu model AI tentunya diperlukan kekuatan komputasi yang kuat. Untuk mengatasi hal ini, penelitian ini menerapkan teknik *Model Quantization*, yaitu optimasi model dengan mengubah presisi bobot jaringan saraf tanpa menghilangkan struktur aslinya. Teknik ini memungkinkan model AI tetap memiliki konteks pemahaman visual yang luas namun tetap ringan, cepat, dan hemat energi saat dijalankan pada *smartphone* spesifikasi menengah ke bawah (Bae et al., 2020).

Selain aspek perangkat, aspek kepercayaan medis (*medical trust*) menjadi faktor krusial dalam adopsi teknologi AI di bidang kesehatan. Model AI seringkali dianggap sebagai *black box* yang sulit dijelaskan dasarnya. Penelitian ini mengintegrasikan *Explainable AI* (XAI) menggunakan metode Grad-CAM. Fitur ini akan memvisualisasikan area pada serviks yang dianggap mencurigakan melalui *heatmap*, sehingga tenaga medis dapat memahami dasar keputusan AI dan melakukan validasi secara klinis (Sritharan et al., 2025).

1.2 Rumusan Masalah

latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana penerapan *transfer learning* pada model *Convolutional Neural Network* (CNN) untuk klasifikasi citra tes IVA secara akurat
2. Bagaimana pengaruh *Post-Training Quantization* pada model CNN terhadap efisiensi sumber daya (ukuran model dan waktu inferensi) di perangkat Android?
3. Bagaimana efektivitas Grad-CAM dalam memberikan visualisasi *explainability* terhadap keputusan model pada citra medis kanker serviks?
4. Bagaimana implementasi model CNN ke dalam aplikasi Android untuk melakukan klasifikasi citra tes IVA secara *offline* guna mendukung skrining dini kanker serviks?

1.3 Batasan Masalah

Agar penelitian tetap terarah dan terukur, ditetapkan beberapa batasan sebagai berikut:

1. Dataset yang digunakan terbatas pada citra tes IVA yang diperoleh melalui kolaborasi dengan Poli Onkologi Rumah Sakit Umum Pusat Hasan Sadikin serta dataset berlisensi dari IARC/WHO.
2. Teknik optimasi yang diterapkan dibatasi pada *Post-Training Quantization* (PTQ).
3. *Output* sistem berupa klasifikasi biner (*Normal/Positive*) yang dilengkapi dengan visualisasi *heatmap* Grad-CAM sebagai bentuk interpretabilitas model.
4. Ruang lingkup medis sistem berfungsi sebagai alat bantu skrining (*decision support*), bukan sebagai pengganti diagnosis klinis akhir oleh dokter spesialis maupun pemeriksaan *histopatologi*.

1.4 Tujuan

Adapun tujuan yang ingin dicapai dari penelitian ini adalah:

1. Menghasilkan model CNN berbasis *transfer learning* yang optimal untuk mengklasifikasikan citra tes IVA dengan tingkat akurasi, sensitivitas, dan spesifisitas yang memenuhi standar teknis.
2. Menganalisis pengaruh penerapan *Post-Training Quantization* pada model CNN terhadap efisiensi sumber daya (ukuran model dan waktu inferensi) pada perangkat Android.
3. Mengimplementasikan Grad-CAM sebagai metode *Explainable AI* (XAI) untuk meningkatkan transparansi dan kepercayaan tenaga medis terhadap hasil deteksi otomatis.
4. Menerapkan model CNN ke dalam sebuah aplikasi Android yang dapat melakukan klasifikasi citra tes IVA secara *offline* guna membantu tenaga kesehatan dalam skrining dini kanker serviks.

1.5 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi Tenaga Medis: Memberikan alat bantu objektif dalam menginterpretasi hasil tes IVA, sehingga mengurangi risiko kesalahan diagnosis akibat subjektivitas visual.
2. Bagi Pasien: Meningkatkan peluang deteksi dini kanker serviks, terutama bagi masyarakat di daerah pelosok yang jauh dari fasilitas laboratorium canggih.
3. Bagi Penelitian dan Masyarakat: Memberikan kontribusi pada bidang riset *Mobile-based deployment* dan optimasi model pembelajaran mendalam untuk kebutuhan medis di perangkat *mobile*.

[Halaman ini sengaja dikosongkan]

BAB 2 TINJAUAN PUSTAKA

2.1 Hasil Penelitian Terdahulu

Berikut adalah hasil pembahasan penelitian terdahulu yang dijadikan sebagai landasan pedoman untuk penelitian ini. Untuk informasi lengkapnya bisa dilihat pada Tabel 2.1 yang berisi tentang judul penelitian, penulis, tahun penelitian, dan pembahasan dari penelitian tersebut.

Tabel 2.1 Penelitian Terdahulu

No.	Penulis (Tahun)	Pembahasan
1	<i>Automated Cervical Cancer Detection through Visual Inspection with Acetic Acid in Resource-Poor Settings with Lightweight Deep learning Models Deployed on an Android Device</i>	
	(Maben et al., 2025)	Penelitian ini mengembangkan sistem deteksi kanker serviks berbasis Android yang dirancang untuk lingkungan dengan keterbatasan sumber daya. Metodologi yang digunakan mengintegrasikan arsitektur <i>lightweight</i> MobileNet-V2 yang dipangkas (<i>truncated</i>) bersama EfficientDet-Lite3 sebagai detektor <i>Region of Interest</i> (ROI). Model dikonversi ke format TensorFlow Lite (TFLite) untuk memungkinkan inferensi secara <i>offline</i> pada perangkat <i>mobile</i> . Hasil evaluasi menunjukkan akurasi sebesar 92,31% dan sensitivitas 98,24% pada perangkat kelas menengah, yang mengindikasikan bahwa optimasi melalui reduksi parameter tetap mampu mempertahankan performa diagnostik pada platform seluler.
2	<i>Integrating artificial intelligence with smartphone-based imaging for cancer detection in vivo</i>	
	(Song & Liang, 2025)	Studi literatur ini menguraikan landasan teoretis terkait integrasi kecerdasan buatan pada sistem pencitraan berbasis smartphone untuk deteksi kanker secara <i>in vivo</i> , dengan penekanan pada kebutuhan arsitektur yang efisien guna menyesuaikan keterbatasan komputasi dan memori pada perangkat <i>mobile</i> . Selain itu, dibahas pentingnya penerapan <i>Explainable AI</i> (XAI), seperti Grad-CAM, untuk meningkatkan interpretabilitas model dan mendukung kepercayaan dalam pengambilan keputusan klinis. Tinjauan tersebut menjadi dasar konseptual bagi pengembangan sistem yang tidak hanya berorientasi pada

No.	Penulis (Tahun)	Pembahasan
		performa klasifikasi, tetapi juga pada aspek transparansi visual dalam konteks aplikasi medis.
3	<i>Using Dynamic Features for Automatic Cervical Precancer Detection</i>	
	(Viñals et al., 2021)	Penelitian ini membedah penggunaan fitur dinamis melalui analisis kurva intensitas warna temporal yang diperoleh dari rekaman video tes IVA menggunakan <i>smartphone</i> . Algoritma yang dikembangkan melibatkan teknik reduksi dimensi <i>Principal Component Analysis</i> (PCA) dan klasifikasi berbasis <i>Artificial Neural Network</i> (ANN). Hasil eksperimen menunjukkan sensitivitas sebesar 90% dan spesifisitas 87%, yang secara signifikan melampaui performa rata-rata pakar kolposkopi senior. Temuan ini memperkuat landasan teoretis bahwa sensor kamera <i>smartphone</i> memiliki kapabilitas teknis yang memadai untuk ekstraksi biomarker digital serviks.
4	<i>Cervical Lesion Classification Method Based on Cross-Validation Decision Fusion Method of Vision Transformer and DenseNet</i>	
	(Li et al., 2022)	Penelitian ini mengusulkan skema fusi keputusan (<i>decision fusion</i>) yang menggabungkan karakteristik <i>Vision Transformer</i> (ViT) dan <i>DenseNet161</i> melalui metode <i>fivefold cross-validation</i> . Inovasi teoretis terletak pada kemampuan sistem untuk menangkap korelasi fitur global dan lokal secara bersamaan guna melakukan klasifikasi empat kategori lesi (Normal, LSIL, HSIL, <i>Cancer</i>).
5	<i>Dual-attention EfficientNet based on Multi-view Feature Fusion for Cervical Squamous Intraepithelial Lesions Diagnosis</i>	
	(Guo et al., 2022)	Paper ini memperkenalkan model MCFNet yang mengadopsi struktur EfficientNet-B0 dengan integrasi mekanisme <i>Dual-Attention</i> (CBAM dan <i>Coordinate Attention</i>). Strategi penelitian difokuskan pada fusi fitur <i>multi-view</i> yang mengombinasikan citra serviks <i>original</i> , asam asetat, dan iodin untuk meningkatkan presisi diagnostik hingga 90%.
6	<i>Diagnosis of Cervical Precancerous Lesions Based on Multimodal Feature Changes</i>	

No.	Penulis (Tahun)	Pembahasan
	(Peng et al., 2021)	Fokus studi ini adalah pemanfaatan citra <i>multimodal</i> melalui registrasi spasial antara citra <i>pre-acetic</i> dan <i>post-acetic acid</i> . Ekstraksi fitur dilakukan menggunakan jaringan <i>double channel</i> VGG16 untuk menangkap perubahan karakteristik jaringan serviks secara simultan. Penelitian ini mendemonstrasikan akurasi sebesar 86,3% dan memberikan argumen teoretis bahwa analisis komparatif antara kondisi sebelum dan sesudah aplikasi asam asetat sangat krusial dalam meminimalisir subjektivitas inspeksi visual.
7	<i>Cervical Pre-cancerous Lesion Detection: Development of Smartphone-Based VIA Application Using Artificial Intelligence</i>	
	(Harsono et al., 2022)	Penelitian oleh Harsono dkk. mengembangkan aplikasi Android berbasis kecerdasan buatan (AI) untuk mendeteksi hasil Inspeksi Visual Asam Asetat (IVA) secara <i>real-time</i> di lingkungan dengan keterbatasan sumber daya. Metodologi yang digunakan mencakup tiga tahapan utama, yaitu prapemrosesan citra serviks yang diambil menggunakan <i>smartphone</i> (sebelum dan sesudah tes IVA), ekstraksi fitur, serta pengembangan klasifikasi melalui perbandingan sembilan algoritma berbeda. Berdasarkan metrik evaluasi <i>Precision at Recall</i> (0,8), algoritma <i>Gradient Boosting Classifier</i> menghasilkan performa terbaik sebesar 0,6827, mengungguli <i>Logistic Regression</i> (0,6613) dan <i>Random Forest Classifier</i> (0,6467). Secara keseluruhan, pengujian dengan 65 data uji dari total 199 data menghasilkan tingkat akurasi model akhir sebesar 93,8%, sensitivitas 80%, spesifisitas 96,4%, presisi 80%, dan ROC/AUC sebesar 0,85.

Berdasarkan tinjauan penelitian terdahulu, terdapat pergeseran dari penggunaan arsitektur *deep learning* berukuran besar menuju model yang lebih efisien untuk implementasi pada perangkat dengan keterbatasan sumber daya. Meskipun model kompleks seperti *Vision Transformer* dan *DenseNet* mampu menghasilkan representasi fitur yang kaya, kebutuhan komputasi yang tinggi membatasi penerapannya pada perangkat *mobile* di fasilitas layanan kesehatan primer. Sebaliknya, arsitektur ringan seperti *MobileNet* menunjukkan kemampuan menjalankan inferensi *offline* dengan performa yang kompetitif. Namun, tantangan yang masih muncul adalah rendahnya transparansi keputusan model dalam konteks klinis.

Penelitian ini mengintegrasikan arsitektur *MobileNetV3*, teknik *Post-Training Quantization* (PTQ), dan metode Grad-CAM dalam satu sistem terpadu untuk menghasilkan model yang efisien, dapat dijalankan secara *on-device* pada perangkat Android, serta memiliki kemampuan visualisasi keputusan. Pendekatan ini menekankan keseimbangan

antara performa klasifikasi, efisiensi komputasi, dan interpretabilitas guna mendukung implementasi sistem skrining pada layanan kesehatan primer.

2.2 Dasar Teori

2.2.1 Kanker Serviks

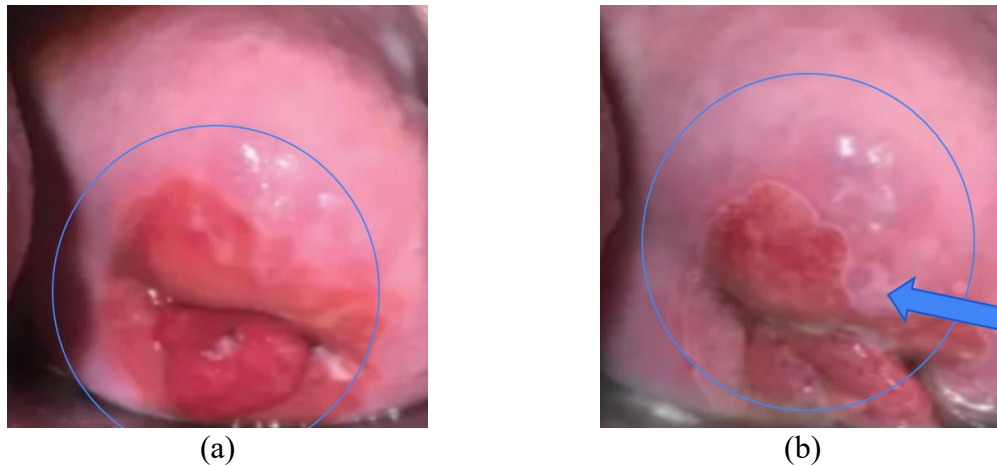
Kanker serviks merupakan jenis kanker keempat yang paling sering menyerang wanita di seluruh dunia, dengan perkiraan 604.000 kasus baru dan 342.000 kematian pada tahun 2020. Fenomena ini menjadi perhatian serius di negara-negara berpenghasilan rendah dan menengah, di mana sekitar 90% dari total kematian akibat kanker serviks global pada tahun tersebut terjadi di wilayah-wilayah tersebut.

Tingginya angka insidensi dan mortalitas di negara berkembang utamanya disebabkan oleh terbatasnya akses masyarakat terhadap program skrining yang memadai. Hal ini menciptakan kesenjangan kesehatan yang signifikan, di mana deteksi dini seringkali terlambat dilakukan sehingga menurunkan peluang keberhasilan pengobatan bagi pasien (Harsono et al., 2022).

2.2.2 Tes IVA

Metode Inspeksi Visual Asam Asetat (IVA) merupakan prosedur pemeriksaan serviks secara visual setelah aplikasi larutan asam asetat sebagai alternatif skrining di wilayah dengan keterbatasan sumber daya. *World Health Organization* (WHO) merekomendasikan IVA sebagai pilihan ketika tes HPV tidak tersedia, mengingat biayanya relatif rendah, prosedurnya sederhana, serta memungkinkan keputusan tindak lanjut dilakukan pada kunjungan yang sama. Secara klinis, pemeriksaan dilakukan melalui penyisipan spekulum untuk mengidentifikasi serviks dan area Sambungan *Skuamosa Kolumnar* (SSK) atau *Squamocolumnar Junction* (SCJ), pengolesan asam asetat 3–5%, ditunggu sekitar 60 detik, kemudian inspeksi visual untuk mendeteksi epitel *acetowhite* yang mengindikasikan kemungkinan lesi prakanker.

Secara morfologis, epitel *acetowhite* ditandai dengan munculnya bercak atau plak berwarna putih pekat dengan batas yang tegas pada permukaan serviks. Karakteristik visual ini memiliki signifikansi klinis yang tinggi apabila area putih tersebut berlokasi di dalam zona transformasi dan bersinggungan langsung dengan garis SSK. Representasi visual dari epitel *acetowhite* yang berada di sekitar area SSK pasca-pemberian asam asetat dapat dilihat pada Gambar 2.1. Meskipun prosedur ini efektif, akurasi tes IVA sangat bergantung pada kompetensi dan pengalaman pemeriksa, sehingga variabilitas subjektivitas manusia dalam interpretasi visual dapat memengaruhi ketepatan diagnosis (Harsono et al., 2022).



Gambar 2.1 (a) Serviks Sebelum IVA dan (b) Serviks Setelah IVA

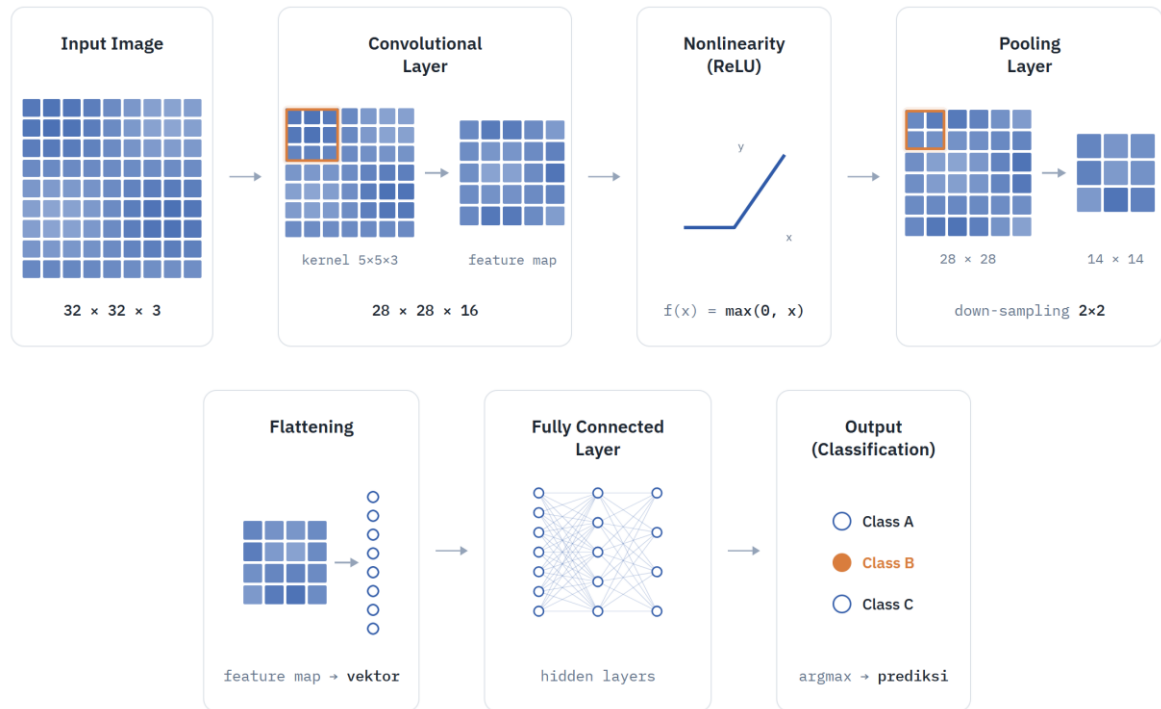
2.2.3 Deep learning

Deep learning merupakan cabang kecerdasan buatan yang memanfaatkan jaringan saraf tiruan berlapis (*deep neural networks*) untuk mempelajari representasi data secara hierarkis. Berbeda dengan *machine learning* tradisional yang bergantung pada ekstraksi fitur manual, *deep learning* mampu melakukan ekstraksi fitur secara otomatis dari data mentah melalui representasi non-linear pada berbagai lapisan tersembunyi. Keunggulan utamanya meliputi kemampuan pembelajaran berbasis data skala besar, representasi fungsi kompleks, serta pemodelan pola melalui struktur yang menyerupai jaringan saraf biologis. Dalam praktiknya, beberapa arsitektur utama *deep learning* antara lain *Convolutional Neural Networks* (CNN), *Recurrent Neural Networks* (RNN), dan *Deep Neural Networks* (DNN) (Purwono et al., 2022).

2.2.4 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan varian algoritma *deep learning* yang mengadopsi mekanisme fungsional korteks visual manusia untuk mengeksekusi tugas-tugas visi komputer secara komputasional. Operasi internal model melibatkan pemrosesan citra digital dalam format matriks numerik guna merepresentasikan intensitas cahaya pada tiap piksel sehingga memungkinkan ekstraksi fitur dilakukan secara otomatis melalui representasi hirarkis. Mekanisme ini mengamplifikasi pola signifikan pada *input* data melalui lapisan-lapisan jaringan yang tersembunyi untuk menghasilkan identifikasi objek yang terukur.

Struktur arsitektur CNN terbagi ke dalam empat lapisan fungsional utama meliputi lapisan konvolusi, *pooling*, *fully connected*, dan non-linearitas. Lapisan konvolusi mengaplikasikan filter *kernel* untuk menghasilkan *activation map* melalui perhitungan produk titik antara bobot filter dan nilai piksel, sedangkan lapisan *pooling* mengeksekusi representasi *down-sampling* guna meminimalkan dimensi parameter dan beban komputasi. Klasifikasi akhir ditentukan melalui proses *flattening* yang mengonversi matriks fitur menjadi vektor tunggal pada lapisan *fully connected* dengan dukungan fungsi aktivasi non-linear guna mengoptimasi keputusan luaran. Konfigurasi lapisan yang sistematis ini memberikan efisiensi inferensi yang optimal sekaligus mempertahankan sensitivitas tinggi terhadap fitur-fitur visual yang relevan (Purwono et al., 2022).



Gambar 2.2 Arsitektur *Convolutional Neural Network*

2.2.5 Arsitektur MobileNet V3

MobileNetV3 merupakan arsitektur *Convolutional Neural Network* (CNN) versi ketiga yang dikembangkan oleh Google guna memfasilitasi tugas analisis gambar dan deteksi objek pada platform komputasi terbatas. Pengembangan model ini melibatkan kombinasi *Network Architecture Search* (NAS) yang dilengkapi dengan algoritma NetAdapt untuk melakukan kustomisasi struktur jaringan secara otomatis terhadap spesifikasi *Central Processing Unit* atau CPU ponsel. Rancang bangun arsitektur tersebut difokuskan pada pencapaian efisiensi komputasi tinggi melalui minimalisasi jumlah parameter sehingga menghasilkan performa inferensi yang lebih ringan dan cepat pada perangkat *mobile* (Zhao & Wang, 2022).

Operasi internal MobileNetV3 menerapkan teknik *depthwise separable convolution* guna mereduksi intensitas operasi komputasi serta mengintegrasikan modul *Squeeze-and-Excitation* (SE) untuk memperkuat kapasitas representasi fitur melalui mekanisme atensi saluran. Arsitektur ini mengadopsi fungsi aktivasi *hard-swish* sebagai komponen pengganti ReLU konvensional karena karakteristiknya yang lebih efisien dalam lingkungan dengan keterbatasan daya komputasi. Pemanfaatan komponen teknis tersebut memungkinkan model untuk menjaga akurasi diagnostik yang signifikan sekaligus mempertahankan profil penggunaan sumber daya yang rendah pada perangkat *on-device* (Howard et al., 2019).

2.2.6 Arsitektur ResNet

ResNet (Residual Network) merupakan arsitektur *Convolutional Neural Network* (CNN) yang diperkenalkan oleh He et al. guna mengatasi permasalahan degradasi performa pada jaringan saraf dalam yang terjadi akibat fenomena *vanishing gradient* dan *exploding gradient*. Arsitektur ini menjadi tonggak penting dalam perkembangan *deep learning* karena memungkinkan pelatihan jaringan dengan kedalaman yang jauh lebih besar dibandingkan model CNN konvensional. Keunggulan utama ResNet terletak pada kemampuannya

mempertahankan kinerja yang tinggi seiring bertambahnya kedalaman jaringan melalui desain yang efisien dan terstruktur.

Komponen inti dari arsitektur ResNet adalah *residual block* yang dilengkapi dengan mekanisme *shortcut connection*. Ketika matriks fitur memasuki sebuah *residual block*, aliran informasi dibagi menjadi dua cabang: satu cabang melewati lapisan konvolusional untuk ekstraksi fitur, sementara cabang lainnya melewati lapisan perantara secara langsung dan terhubung ke keluaran blok. Mekanisme ini memungkinkan jaringan untuk mempelajari pemetaan residual antara masukan dan keluaran yang direpresentasikan sebagai $f(x) = H(x) - x$, di mana $H(x)$ merupakan pemetaan yang diinginkan dan x adalah identitas masukan. Desain tersebut secara efektif meningkatkan kecepatan konvergensi jaringan sekaligus mempertahankan performa yang kuat seiring meningkatnya kedalaman model (Liao et al., 2025).

Dalam perkembangannya, ResNet hadir dalam beberapa varian yang disesuaikan dengan kompleksitas tugas. ResNet-34 menggunakan struktur *residual block* sederhana dengan dua lapisan konvolusional per blok dan cocok untuk dataset berskala kecil dengan keterbatasan sumber daya komputasi. ResNet-50 mengadopsi struktur *residual block* yang lebih kompleks dengan tiga lapisan konvolusional per blok serta mengintegrasikan *identity* dan *projection residual connection*, sehingga sesuai untuk tugas yang membutuhkan representasi fitur yang kaya. Adapun ResNet-101 memperdalam arsitektur ResNet-50 dengan menambah jumlah *residual block* yang mengandung empat lapisan konvolusional, menjadikannya pilihan tepat untuk permasalahan pengenalan gambar yang lebih kompleks dan dataset berskala besar (Liao et al., 2025).

2.2.7 Arsitektur EfficientNet-B0

EfficientNet B0 merupakan varian pertama dari keluarga arsitektur EfficientNet yang secara keseluruhan terdiri dari delapan model mulai dari B0 hingga B7. Arsitektur ini dirancang sebagai model *lightweight* yang bertujuan mengoptimalkan akurasi dan efisiensi komputasi dalam operasi *floating point* secara bersamaan. Keunggulan utama EfficientNet B0 terletak pada penerapan pendekatan *compound scaling*, yaitu suatu metode penskalaan yang secara simultan memperbesar resolusi, lebar (*width*), dan kedalaman (*depth*) jaringan secara proporsional. Metode *compound scaling* ini memastikan seluruh aspek jaringan dioptimalkan secara seimbang sehingga menghasilkan peningkatan akurasi dan efisiensi yang lebih baik dibandingkan pendekatan penskalaan konvensional yang hanya memperbesar satu dimensi jaringan.

Secara struktural, jaringan EfficientNet B0 tersusun atas blok-blok *mobile inverted bottleneck convolution* (MBConv) yang menerima masukan berupa citra tiga kanal dengan resolusi piksel 224×224 . Blok MBConv merupakan komponen inti yang menggabungkan konvolusi *depthwise separable* dengan struktur residual terinversi, sehingga memungkinkan ekstraksi fitur yang efisien dengan beban komputasi yang lebih rendah. *Feature map* yang dihasilkan oleh lapisan konvolusional terakhir EfficientNet B0 selanjutnya dapat dimanfaatkan sebagai representasi fitur untuk berbagai tugas klasifikasi maupun deteksi gambar. Desain arsitektur yang ringkas namun berperforma tinggi ini menjadikan EfficientNet B0 pilihan yang tepat untuk aplikasi berbasis perangkat dengan keterbatasan sumber daya komputasi (Nigam et al., 2024).

2.2.8 Model Quantization

Model *quantization* merupakan teknik optimasi *neural network* dengan mereduksi presisi bobot dan aktivasi dari *floating-point* 32-bit menjadi representasi *bit-depth* yang lebih rendah guna menurunkan konsumsi memori dan beban komputasi. Secara matematis, proses ini menggunakan pemetaan yang menghubungkan nilai integer terkuantisasi (q) dengan nilai real (r) melalui parameter skala (S) dan *zero-point* (Z), yang dirumuskan sebagai $r = S(q - Z)$. Parameter Z merepresentasikan nilai nol pada domain terkuantisasi sehingga operasi seperti *zero-padding* tetap konsisten tanpa perubahan struktur jaringan. Metode kuantisasi terbagi menjadi *Post-Training Quantization* (PTQ) yang diterapkan setelah pelatihan serta *Quantization-Aware Training* (QAT) yang mengintegrasikan simulasi presisi rendah selama pelatihan untuk mengurangi degradasi akurasi. Untuk meminimalkan kesalahan pada konfigurasi *bit-width* rendah, digunakan *Parameterized Clipping Activation* (PACT) dengan mekanisme $a_{clip} = \text{clip}(a, -\alpha, \alpha)$, di mana parameter kliping dipelajari secara adaptif terhadap distribusi data (Rezabeyk et al., 2024). Implementasi kuantisasi *integer-only* memungkinkan inferensi dijalankan sepenuhnya menggunakan operasi integer sehingga lebih efisien pada perangkat *mobile*, dengan reduksi beban komputasi yang signifikan dibandingkan model *floating-point* konvensional, sehingga sesuai untuk *deployment* aplikasi medis berbasis Android secara *offline* (Jacob et al., n.d.).

2.2.9 Explainable AI

Explainable AI (XAI) merupakan sub-bidang kecerdasan buatan yang bertujuan memberikan pemahaman mengenai mekanisme pengambilan keputusan model, khususnya pada sistem *deep learning* yang bersifat *black-box*. Pendekatan XAI diklasifikasikan menjadi teknik interpretabilitas *post-hoc*, seperti *saliency maps* dan metode atribusi fitur, serta arsitektur yang bersifat *intrinsically interpretable*. Dalam konteks kesehatan, efektivitas XAI ditentukan oleh kesesuaian antara penjelasan yang diberikan dengan proses penalaran aktual model serta kemampuannya untuk dipahami secara intuitif oleh tenaga medis. Teknik visualisasi atribusi fitur memungkinkan identifikasi area citra yang berkontribusi terhadap prediksi, sehingga mendukung validasi klinis dan meningkatkan kepercayaan terhadap sistem. Oleh karena itu, integrasi XAI dalam aplikasi skrining kesehatan diperlukan untuk memastikan transparansi, relevansi kontekstual, dan akuntabilitas keputusan model (Räz et al., 2025).

Salah satu teknik penjelasan *post-hoc* lokal yang paling populer dan banyak diimplementasikan pada klasifikasi berbasis citra medis adalah *Gradient-weighted Class Activation Mapping* (Grad-CAM), yang pertama kali diperkenalkan oleh Selvaraju et al. (2017). Grad-CAM bekerja dengan memanfaatkan informasi gradien yang mengalir ke lapisan konvolusi terakhir dari jaringan saraf untuk menghasilkan peta lokalisasi kasar (*coarse localization map*) yang menyoroti wilayah-wilayah penting dalam gambar guna memprediksi kelas tertentu.

Secara matematis, untuk memperoleh peta lokalisasi Grad-CAM $L_{\text{Grad-CAM}}^c$ bagi kelas target c , langkah pertama adalah menghitung gradien skor kelas y^c (skor sebelum fungsi aktivasi seperti *softmax* atau *sigmoid/logit*) terhadap peta fitur (*feature map*) A^k dari lapisan konvolusi terakhir jaringan. Gradien-gradien ini kemudian dirata-ratakan secara global (*global average pooling*) melalui dimensi spasial (lebar i dan tinggi j) untuk mendapatkan bobot kepentingan saluran (*channel importance weights*) α_k^c , yang diformulasikan sebagai:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \quad (2.1)$$

Di mana:

- α_k^c adalah bobot kepentingan saluran untuk kanal k pada kelas target c .
- Z merepresentasikan luas atau dimensi spasial dari peta fitur (perkalian lebar dan tinggi peta fitur).
- y^c adalah skor prediksi sebelum fungsi aktivasi (*logit*) untuk kelas target c .
- A_{ij}^k menyatakan nilai aktivasi pada lokasi spasial (i, j) dari peta fitur ke- k .

Setelah bobot kepentingan α_k^c diperoleh, dilakukan kombinasi linear berbobot dari seluruh peta fitur A^k . Kombinasi linear tersebut kemudian dilewatkan pada fungsi aktivasi *Rectified Linear Unit* (ReLU) untuk memperoleh peta lokalisasi final $L_{\text{Grad-CAM}}^c$ yang diformulasikan sebagai:

$$L_{\text{Grad-CAM}}^c = \text{ReLU} \left(\sum_k \alpha_k^c A^k \right) \quad (2.2)$$

Penerapan fungsi ReLU dalam persamaan di atas bertujuan untuk menapis fitur-fitur negatif yang berasosiasi dengan kelas selain kelas target c . Dengan kata lain, model hanya mempertahankan fitur-fitur yang berkontribusi positif (meningkatkan skor kelas c) dan mengabaikan fitur yang berkontribusi negatif (menurunkan skor kelas c) terhadap keputusan klasifikasi (Selvaraju et al., 2020). Peta aktivasi mentah yang dihasilkan ini kemudian di-*upsample* ke dimensi citra asli menggunakan interpolasi bilinear untuk disajikan bersama citra *input* asli sebagai peta atensi spasial (*overlay heatmap*).

2.2.10 Metrik Evaluasi

Evaluasi performa model klasifikasi dalam konteks medis memerlukan pemilihan metrik yang tepat agar hasil penilaian mencerminkan kualitas diagnostik secara klinis, bukan sekadar performa statistik umum. Hal ini menjadi krusial terutama pada dataset yang tidak seimbang (*imbalanced*), di mana jumlah sampel antar kelas berbeda secara signifikan. Sub-bab ini menjelaskan dasar teori dari seluruh metrik evaluasi yang digunakan dalam penelitian ini.

2.2.10.1 Sensitivitas (*Recall*)

Sensitivitas, yang juga dikenal sebagai *recall* atau *True positive Rate* (TPR), mengukur proporsi kasus Positif yang berhasil diidentifikasi dengan benar oleh model terhadap seluruh kasus yang sesungguhnya Positif. Dalam konteks skrining medis, sensitivitas menjadi metrik paling kritis karena nilai yang rendah mengindikasikan banyak kasus lesi prakanker yang tidak terdeteksi (Harsono et al., 2022). Secara matematis, sensitivitas dirumuskan sebagai:

$$\text{Sensitivitas} = TP / (TP + FN) \quad (2.3)$$

di mana TP adalah *True positive* dan FN adalah *False Negative*. Nilai sensitivitas berkisar antara 0 hingga 1, di mana nilai mendekati 1 menunjukkan kemampuan deteksi kasus positif yang sangat baik.

Spesifisitas, atau *True negative Rate* (TNR), mengukur proporsi kasus Negatif yang berhasil diidentifikasi dengan benar oleh model terhadap seluruh kasus yang sesungguhnya Negatif. Metrik ini mengukur kemampuan model dalam menekan laju *false alarm* atau diagnosis positif yang salah, sehingga menjaga efisiensi sistem skrining (Harsono et al., 2022). Rumus spesifisitas adalah sebagai berikut:

$$\text{Spesifisitas} = TN / (TN + FP) \quad (2.4)$$

di mana TN adalah *True negative* dan FP adalah *False Positive*. Dalam praktik klinis, terdapat *trade-off* antara sensitivitas dan spesifisitas yang harus diseimbangkan sesuai dengan kebutuhan aplikasi.

2.2.10.4 Presisi (*Precision*)

Presisi mengukur proporsi prediksi Positif yang benar-benar merupakan kasus Positif dari seluruh prediksi yang diklasifikasikan sebagai Positif oleh model. Metrik ini mengukur tingkat kepercayaan terhadap setiap prediksi positif yang dikeluarkan model (Viñals et al., 2021). Rumus presisi adalah:

$$\text{Presisi} = TP / (TP + FP) \quad (2.5)$$

Presisi menjadi sangat relevan dalam skenario di mana setiap prediksi positif yang keliru memiliki biaya tinggi, misalnya prosedur tindak lanjut klinis yang invasif. Dalam penelitian ini, metrik *Precision@Recall=80%* digunakan sebagai acuan komparatif dengan penelitian *baseline* Harsono et al. (2022) (Harsono et al., 2022), yakni nilai presisi yang diperoleh pada titik *threshold* di mana *recall* model mencapai tepat 80%.

2.2.10.5 PR-AUC (*Area Under the Precision-Recall Curve*)

Kurva *Precision-Recall* (PR) memvisualisasikan hubungan antara presisi dan *recall* pada berbagai nilai *threshold* klasifikasi. Area di bawah kurva ini (PR-AUC) merangkum performa model ke dalam satu nilai skalar yang merepresentasikan kemampuan model mempertahankan keseimbangan presisi-*recall* secara keseluruhan.

PR-AUC merupakan metrik yang lebih robust dibandingkan ROC-AUC pada dataset yang tidak seimbang (*imbalanced*), karena kurva PR berfokus pada performa model terhadap kelas minoritas (Sritharan et al., 2025). Secara formal, PR-AUC dihitung sebagai:

$$\text{PR-AUC} = \int_0^1 P(R) dR \quad (2.6)$$

di mana $P(R)$ adalah fungsi presisi terhadap *recall*. Dalam implementasi praktis, nilai ini diperkirakan menggunakan metode trapesoid atas titik-titik diskret pada kurva PR. Nilai

PR-AUC mendekati 1 mengindikasikan performa model yang sangat baik pada dataset tidak seimbang.

2.2.10.6 ROC-AUC (*Area Under the Receiver Operating Characteristic Curve*)

Kurva *Receiver Operating Characteristic* (ROC) memvisualisasikan hubungan antara *True positive Rate* (sensitivitas) dan *False Positive Rate* (1 – spesifisitas) pada berbagai nilai *threshold*. Area di bawah kurva ROC (ROC-AUC) mengukur kemampuan diskriminatif model secara keseluruhan terlepas dari *threshold* tertentu (Fawcett, 2006). Secara matematis, ROC-AUC didefinisikan sebagai:

$$ROC-AUC = \int_0^1 TPR(FPR^{-1}(t)) dt \quad (2.7)$$

Nilai ROC-AUC = 0,5 setara dengan klasifikasi acak, sedangkan ROC-AUC = 1,0 menunjukkan diskriminasi sempurna. Meskipun ROC-AUC bersifat *threshold-independent* dan berguna untuk perbandingan model secara umum, metrik ini dapat memberikan gambaran yang terlalu optimis pada dataset tidak seimbang, sehingga perlu dikombinasikan dengan PR-AUC (Li et al., 2022).

Dalam penelitian ini, kombinasi kelima metrik di atas sensitivitas, spesifisitas, presisi, PR-AUC, dan ROC-AUC digunakan secara komplementer untuk menilai performa model CNN dari perspektif klinis maupun statistik. Prioritas utama diberikan pada PR-AUC sebagai metrik seleksi model utama karena kestabilannya terhadap distribusi kelas yang tidak seimbang pada dataset, diikuti oleh nilai sensitivitas minimum yang ditetapkan sebagai syarat keamanan klinis alat skrining.

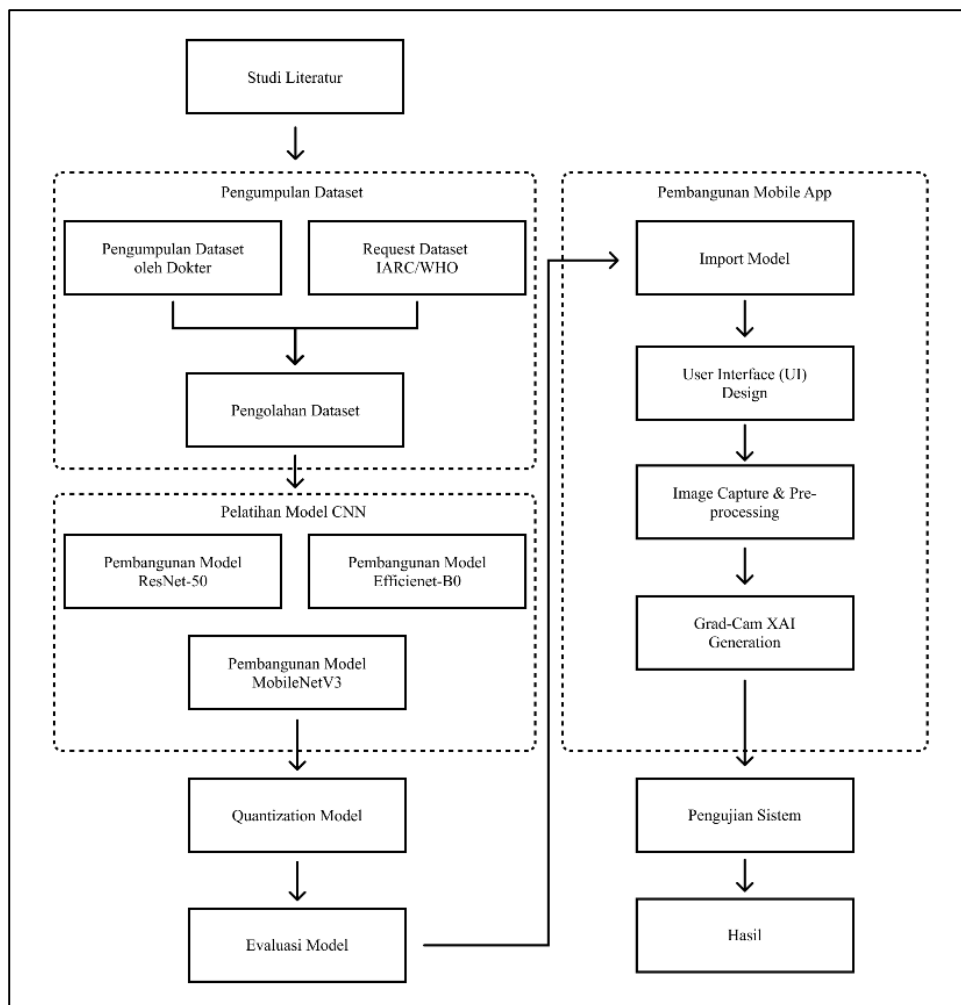
[Halaman ini sengaja dikosongkan]

BAB 3

METODOLOGI PENELITIAN

3.1 Diagram Metodologi

Metode yang digunakan dalam penelitian ini dibagi menjadi 3 tahapan yaitu pengumpulan dataset, pelatihan model CNN, dan pembangunan *mobile app*. Metode ini bertujuan untuk mendapatkan model terbaik untuk melakukan klasifikasi tes IVA sebelum diintegrasikan dengan *mobile app*. Berikut merupakan detail dari tahapan metodologi yang akan divisualisasikan pada Gambar 3.1 dan dijelaskan lebih rinci pada sub-bab selanjutnya.



Gambar 3.1 Diagram Alir Sistem

3.1.1 Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini bersumber dari dua institusi, yaitu citra tes IVA yang dikumpulkan melalui kolaborasi dengan dokter pada poli onkologi Rumah Sakit Umum Daerah Hasan Sadikin Bandung sebagai dataset primer dan dataset *International Agency for Research on Cancer (IARC) WHO* yang diperoleh dari repositori resmi WHO, mencakup *Image Bank Visual Inspection with Acetic Acid (VIA)* (International Agency for Research on Cancer (IARC), 2021b) dan *IARC Colposcopy Image* (International Agency for

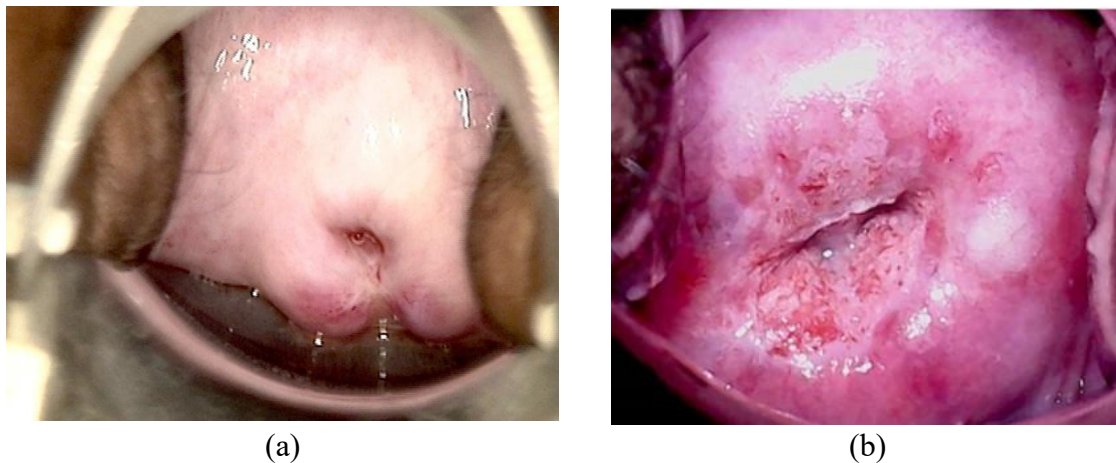
Research on Cancer (IARC), 2021a) . Kedua dataset telah melalui proses anotasi dan validasi oleh tenaga medis ahli sehingga dapat langsung digunakan sebagai data latih model CNN.

Secara klinis, citra tes IVA diklasifikasikan ke dalam tiga status, yaitu *Negative*, *Positive* dan, *Cancer*. Namun, dalam konteks penelitian ini, ketiga kelas tersebut dikonsolidasikan menjadi skema klasifikasi biner dengan pertimbangan klinis bahwa tujuan utama sistem skrining adalah membedakan kondisi normal dari kondisi yang membutuhkan tindak lanjut medis. Oleh karena itu, kelas *Positive* (lesi prakanker) dan kelas *Cancer* (lesi kanker invasif) digabungkan menjadi satu kelas *Positive* biner, sedangkan kelas *Negative* tetap dipertahankan sebagai kelas negatif. Distribusi dataset setelah penggabungan ditampilkan pada Tabel 3.1 di bawah ini.

Tabel 3.1 Sumber Dataset

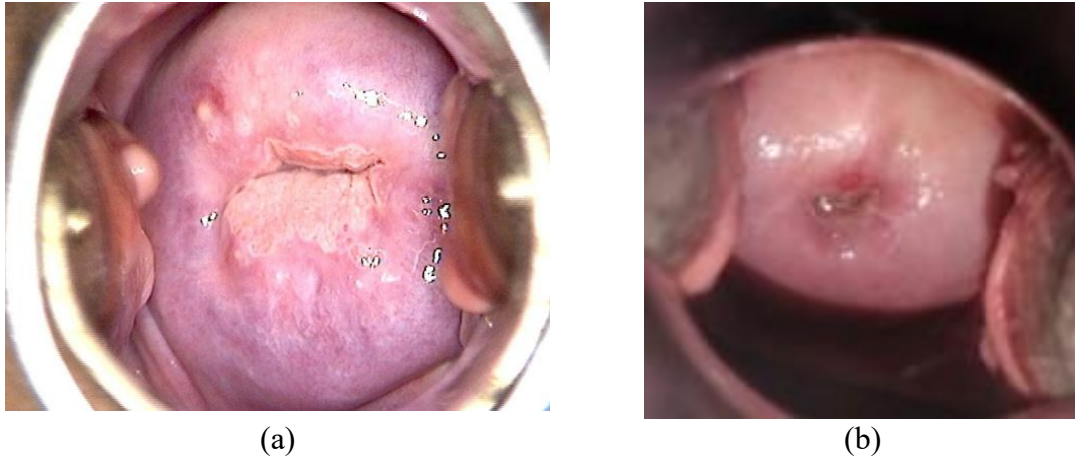
Status IVA	Dataset Primer	Dataset IARC
<i>Negative</i>	15	263
<i>Positive</i>	95	322

Sebagai representasi visual dari dataset yang digunakan pada penelitian ini, perbandingan karakteristik citra antara kondisi serviks normal (*Negative*) dan kondisi yang memiliki indikasi lesi (*Positive*) dapat dilihat pada Gambar 3.2.



Gambar 3.2 (a) Serviks Normal dan (b) Terindikasi Kanker Serviks

Penting untuk dicatat bahwa proses akuisisi citra pada kedua dataset ini dilakukan menggunakan berbagai jenis perangkat seluler dengan spesifikasi optik yang heterogen dan kondisi pencahayaan yang tidak terstandarisasi. Variabilitas kualitas citra ini dipertahankan secara sengaja untuk meningkatkan kemampuan generalisasi (*robustness*) model. Dengan melatih model pada data yang beragam secara visual, diharapkan sistem mampu beradaptasi dengan kondisi riil di lapangan baik di Puskesmas maupun fasilitas layanan kesehatan primer di daerah terpencil di mana spesifikasi kamera dan kondisi pencahayaan seringkali tidak ideal. Citra berwarna pada dataset IARC WHO memiliki resolusi yang telah terstandarisasi, yaitu 960x720 piksel, sedangkan dataset kolaborasi dengan primer memiliki resolusi yang bervariasi, sebagaimana ditunjukkan pada Gambar 3.3 dan Gambar 3.4.



Gambar 3.3 (a) Contoh Gambar Dataset IARC WHO dan (b) Contoh Gambar Dataset Kolaborasi Primer

Secara visual, representasi citra pada kelas *Negative* ditandai dengan penampakan epitel serviks yang normal tanpa adanya reaksi perubahan warna pasca-pemberian asam asetat. Sebaliknya, kelas *Positive* ditandai dengan kemunculan epitel *acetowhite* berupa plak atau bercak putih dengan batas yang tegas maupun keberadaan lesi kanker invasif. Perbedaan karakteristik morfologis yang kontras ini menjadi landasan kuat dalam penyatuan kelas menjadi klasifikasi biner, di mana fokus utama sistem adalah mendeteksi anomali yang memerlukan intervensi klinis.

Dalam proses pengenalanannya, arsitektur *Convolutional Neural Network* (CNN) tidak diinstruksikan secara manual untuk mencari warna tertentu, melainkan mendeteksi lesi tersebut melalui mekanisme ekstraksi fitur hierarkis (*hierarchical feature extraction*). Pada lapisan konvolusi awal (*low-level features*), filter *kernel* pada model akan menangkap fitur dasar seperti gradien warna dan batas tepi (*edges*) yang tajam, yang merepresentasikan perbatasan antara jaringan serviks normal yang kemerahan dengan epitel *acetowhite*. Seiring dengan kedalaman jaringan (*high-level features*), arsitektur model mengagregasi fitur-fitur spasial tersebut untuk memahami bentuk, tekstur, dan struktur kompleks dari lesi. Kemampuan model dalam menangkap dan mempertahankan konteks spasial dari lesi inilah yang nantinya memungkinkan sistem untuk tidak hanya mengklasifikasikan citra secara akurat, tetapi juga memvisualisasikan area patologis tersebut melalui *heatmap* Grad-CAM pada tahap akhir evaluasi.

3.1.2 Pengolahan Dataset

Tahap pengolahan dataset terdiri dari dua sub-proses utama, yaitu pembagian dataset dan penanganan ketidakseimbangan kelas.

3.1.2.1 Pembagian Dataset

Dataset yang telah dibagi menjadi dua kelas (*Negative* dan *Positive*) dibagi ke dalam tiga subset menggunakan metode *stratified k-fold splitting* untuk memastikan proporsi distribusi kelas yang representatif terjaga pada setiap subset. Pembagian dilakukan dengan rasio 70:15:15 untuk data latih, data validasi, dan data uji. Data uji bersifat *hold-out* dan tidak digunakan dalam proses pelatihan maupun pemilihan model untuk menghindari *data leakage*.

3.1.2.3 Pembagian Dataset

Dikarenakan dataset yang tidak seimbang (278 *Negative* berbanding 417 *Positive*). Diperlukan penanganan ketidakseimbangan tersebut dengan dua teknik. Teknik pertama adalah augmentasi geometris dan fotometris diterapkan hanya pada *training set* secara *on-the-fly* menggunakan *ImageDataGenerator* dari TensorFlow/Keras tanpa menyimpan citra ke *storage*. Data validasi dan uji hanya menerima fungsi *preprocessing* model-spesifik tanpa augmentasi. Parameter augmentasi yang digunakan ditampilkan pada Tabel 3.2

Tabel 3.2 Parameter Augmentasi Dataset

Parameter	Nilai	Deskripsi Efek
rotation_range	20°	Rotasi citra acak $\pm 20^\circ$ untuk mensimulasikan variasi orientasi kamera
width_shift_range	0,10 (10%)	Pergeseran horizontal acak $\pm 10\%$ lebar citra
height_shift_range	0,10 (10%)	Pergeseran vertikal acak $\pm 10\%$ tinggi citra
zoom_range	0,15 (15%)	Perbesaran/perkecilan acak $\pm 15\%$ untuk variasi jarak pengambilan
horizontal_flip	<i>True</i>	Pencerminan horizontal untuk keragaman orientasi lesi
vertical_flip	<i>True</i>	Pencerminan vertikal untuk memaksimalkan augmentasi geometris
brightness_range	[0,85, 1,15]	Penyesuaian kecerahan $\pm 15\%$ mensimulasikan variasi pencahayaan
shear_range	0,10 (10%)	Transformasi <i>shear</i> acak untuk distorsi perspektif ringan
fill_mode	<i>'nearest'</i>	Piksel kosong hasil transformasi diisi dengan nilai piksel terdekat

Dikarenakan jumlah dataset yang tidak seimbang antara kelas *positive* dan normal maka diperlukan *Class Weighting* dengan Pada tahap pelatihan, *loss function* dimodifikasi untuk memberikan bobot penalti yang lebih besar pada kesalahan klasifikasi kelas *Positive*. Bobot kelas W_j dihitung menggunakan persamaan berikut:

$$W_j = N_{total} / (K \times N_j) \quad (1)$$

di mana N_{total} adalah total sampel keseluruhan, K adalah jumlah kelas, dan N_j adalah jumlah sampel pada kelas j . Teknik ini memaksa model untuk memprioritaskan pembelajaran fitur patologis pada kelas *Positive* tanpa memerlukan data sintetis tambahan yang berpotensi mengurangi variabilitas visual dataset.

3.1.2.4 Ablation study Strategi Preprocessing

Selain augmentasi standar, penelitian ini melakukan *ablation study* terhadap sembilan strategi *preprocessing* yang berbeda. Tujuan dari *ablation study* ini adalah untuk mengidentifikasi bagaimana berbagai teknik pengolahan citra *input* mulai dari pipeline tanpa *preprocessing* hingga pendekatan berbasis *deep learning object detection* untuk lokalisasi lesi memengaruhi performa klasifikasi model CNN. Seluruh strategi diuji pada kondisi pelatihan yang identik sehingga perbedaan hasil dapat lihat secara langsung kepada strategi *preprocessing* yang digunakan.

Deskripsi lengkap kesembilan strategi *preprocessing* yang diuji dalam *ablation study* ini ditampilkan pada Tabel 3.3.

Tabel 3.3 Ablation study Strategi Preprocessing yang Diuji

<i>Notebook</i>	<i>Preprocessing</i>	<i>Augmentasi</i>	<i>Cropping/ROI</i>
NB-1	<i>Pure Raw Images</i>	Tidak	Tidak
NB-2	<i>Baseline Augmentasi + Data Generation</i>	Ya	Tidak
NB-3	<i>CBAM Attention Mechanism</i>	Ya	Tidak
NB-4	<i>OpenCV Auto-Crop Serviks</i>	Ya	OpenCV
NB-5	<i>OpenCV Crop Tanpa Augmentasi</i>	Tidak	OpenCV
NB-6	<i>ROI Masking via OpenCV</i>	Ya	ROI Mask
NB-7	<i>Fixed Manual ROI Masking</i>	Ya	Manual ROI
NB-8	<i>Lesion Crop OwlViT Object Detection (Matthias Minderer Alexey Gritsenko, 2022)</i>	Ya	OwlViT
NB-9	<i>Lesion Crop Sevia YOLOv11 (SmartClover, 2026)</i>	Ya	YOLOv11

Strategi pada NB-3 hingga NB-7 menggunakan pendekatan *preprocessing* konvensional berbasis OpenCV maupun tanpa *preprocessing*, sedangkan NB-8 dan NB-9 mengeksplorasi teknik *masking region of interest* (ROI). NB-10 dan NB-11 merepresentasikan pendekatan dua-tahap yang lebih kompleks, di mana sebuah model deteksi objek (OwlViT atau YOLOv11) digunakan terlebih dahulu untuk melokalisasi dan memotong area lesi sebelum citra yang telah di-*crop* diklasifikasikan oleh model CNN.

3.1.3 Pelatihan Model CNN

Tahap pelatihan model *Convolutional Neural Network* (CNN) merupakan tahapan inti dalam penelitian ini yang bertujuan menghasilkan model klasifikasi citra tes IVA yang memiliki performa diagnostik dan efisiensi komputasi yang optimal. Pada tahap ini, tiga arsitektur CNN dilatih menggunakan pendekatan *transfer learning* dua fase dengan kondisi pelatihan yang identik pada setiap kombinasi arsitektur dan strategi *preprocessing*.

3.1.3.1 Pemilihan dan Perancangan Arsitektur Model CNN

Tiga arsitektur CNN digunakan untuk mewakili variasi jumlah parameter dan kebutuhan komputasi, yaitu ResNet-50, EfficientNet-B0, dan MobileNetV3. ResNet-50 memiliki 50 lapisan dengan *skip connection* untuk mencegah *vanishing gradient*. Kapasitas ekstraksi fiturnya yang besar menjadi tolok ukur akurasi maksimal sistem tanpa memperhitungkan efisiensi daya komputasi. Di tingkat menengah, EfficientNet-B0 menyeimbangkan akurasi dan efisiensi melalui metode *compound scaling*. Ukurannya jauh lebih kecil dari ResNet-50, namun tetap memiliki kapasitas fitur di atas MobileNetV3. Sementara itu, MobileNetV3 dirancang khusus untuk perangkat seluler berdaya rendah melalui penggunaan *depthwise separable convolution*, modul *Squeeze-and-Excitation*, dan fungsi aktivasi *hard-swish*.

Ketiga arsitektur menggunakan *classification head* yang identik yang ditambahkan di atas masing-masing *base model* pralatih dari ImageNet. Desain *unified head* ini memastikan perbandingan yang adil antar arsitektur karena perbedaan performa dapat diatribusikan sepenuhnya pada kemampuan representasi *base model*. Struktur *classification head* ditampilkan pada Tabel 3.4.

Tabel 3.4 Rancangan *Unified Classification head*

No	Layer	Konfigurasi	Fungsi
1	<i>Input Layer</i>	<i>Shape</i> (224, 224, 3)	Menerima citra RGB yang telah dinormalisasi menggunakan fungsi <i>preprocessing</i> model-spesifik
2	<i>Base model (Frozen)</i>	<i>Pretrained</i> ImageNet trainable=False	Ekstraksi representasi fitur visual hierarkis; dibekukan pada Fase 1 untuk mempertahankan bobot ImageNet
3	GlobalAveragePooling2D	Agregasi spasial penuh	Meratakan <i>feature map</i> 2D menjadi vektor 1D per channel, menghilangkan ketergantungan pada resolusi spasial
4	BatchNormalization	Momentum=0,99 ϵ =0,001	Normalisasi distribusi aktivasi antar batch untuk stabilitas training

No	Layer	Konfigurasi	Fungsi
			dan mempercepat konvergensi
5	<i>Dropout</i>	Rate: 0,30/0,35/0,40 (model-spesifik)	Regularisasi stokastik untuk mencegah <i>overfitting</i> pada dataset berukuran kecil
6	<i>Dense (Output Layer)</i>	Units=1 activation='sigmoid'	Menghasilkan probabilitas prediksi kelas <i>Positive</i> pada rentang [0,1] untuk klasifikasi biner

3.1.3.2 Hyperparameter Per Model

Setiap arsitektur dikonfigurasi dengan *hyperparameter* yang berbeda, disesuaikan dengan karakteristik masing-masing *base model* (jumlah *layer*, kedalaman, dan sensitivitas terhadap perubahan bobot). Konfigurasi lengkap ditampilkan pada Tabel 3.5.

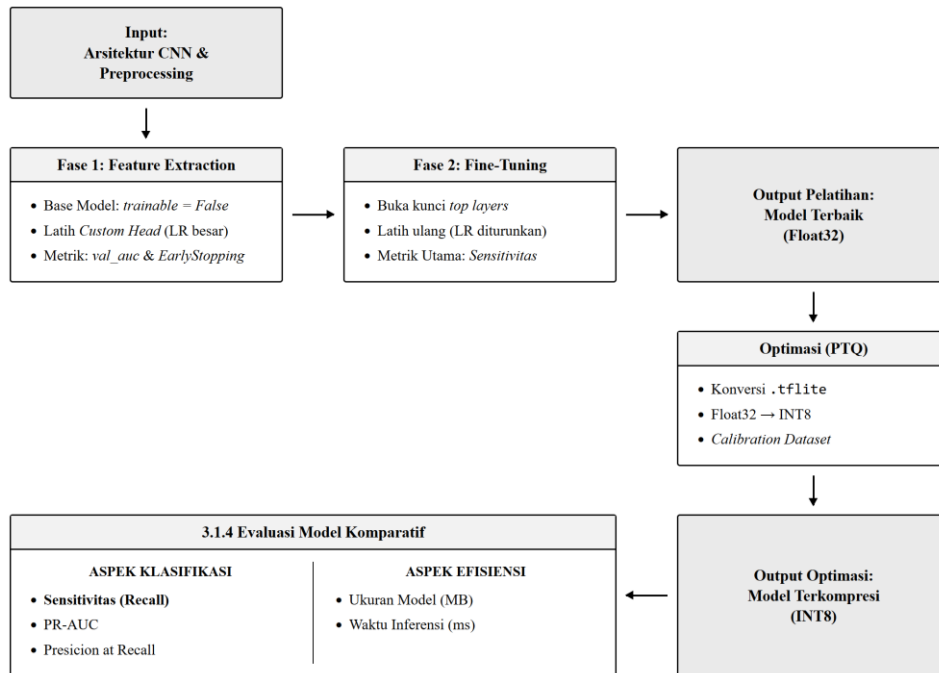
Tabel 3.5 Konfigurasi *Hyperparameter* Lengkap Per Model Per Fase

Parameter	MobileNetV3Small	ResNet-50	EfficientNet-B0
lr phase1	5×10^{-4}	1×10^{-3}	1×10^{-3}
lr phase2	5×10^{-5}	2×10^{-5}	1×10^{-5}
<i>Dropout</i>	0,30	0,40	0,35
Fine tune at	0,50	0,75	0,60
<i>Epochs</i> Phase 1	40	30	30
% <i>Layer</i> Dibuka	50%	25%	40%

Pemilihan konfigurasi *fine-tuning* disesuaikan dengan karakteristik masing-masing arsitektur. Pada MobileNetV3Small, 50% *layer* dibuka untuk *fine-tuning* karena arsitekturnya yang kecil (156 *layer*) relatif aman untuk dilatih ulang lebih banyak, dengan *epoch* yang lebih panjang akibat konvergensi yang lebih lambat. Pada ResNet-50, hanya 25% *layer* yang dibuka mengingat *residual block* bagian dalam sudah sangat stabil dan berisiko mengalami degradasi performa jika terlalu banyak *layer* diaktifkan kembali. Sementara pada EfficientNet-B0, 40% *layer* dibuka karena sifat *compound scaling*-nya memungkinkan *fine-tuning* pada lebih banyak *layer high-level* tanpa kehilangan representasi dasar yang sudah dipelajari.

3.1.3.2 Skema Pelatihan dan Evaluasi Model

Proses pelatihan untuk setiap pasangan arsitektur dan strategi prapemrosesan (*preprocessing*) dilakukan melalui skema *transfer learning* dua fase yang berurutan. Secara komprehensif, keseluruhan alur mulai dari strategi pelatihan, optimasi model, hingga evaluasi akhir divisualisasikan pada Gambar 3.5.



Gambar 3.4 Skema Pelatihan dan Evaluasi Model

Pada fase pertama, yaitu ekstraksi fitur (*feature extraction*), seluruh *layer* pada *base model* pralatih dibekukan (*trainable = False*) untuk mempertahankan representasi fitur visual umum yang telah dipelajari dari ImageNet. Pada tahap ini, hanya *custom classification head* yang dilatih dari awal menggunakan laju pembelajaran (*learning rate*) yang relatif lebih besar (*lr_phase1*). Langkah ini bertujuan untuk membangun fondasi klasifikasi dasar tanpa merusak bobot representasi visual pada *base model* yang masih bersifat acak di awal pelatihan. Selama proses ini berlangsung, pemantauan dilakukan menggunakan metrik *Validation AUC* (*val_auc*) sebagai basis untuk aktivasi mekanisme *EarlyStopping*, *ReduceLROnPlateau*, dan penyimpanan *checkpoint* model terbaik dengan detail pada Tabel 3.6.

Selanjutnya, proses dilanjutkan ke fase kedua, yaitu penyelarasan halus (*fine-tuning*). Model terbaik hasil ekstraksi fitur dimuat kembali, kemudian sebagian *layer* teratas pada *base model* dibuka kuncinya (*trainable = True*) sesuai dengan parameter *fine_tune_at*. Sebagai contoh, nilai *fine_tune_at = 0,8* mengindikasikan bahwa 80% *layer* awal tetap dibekukan, sementara 20% *layer* teratas dibuka untuk dilatih ulang. Model kemudian dikompilasi ulang dengan laju pembelajaran yang jauh lebih lambat (*lr_phase2*). Penurunan laju pembelajaran ini krusial untuk melakukan adaptasi halus pada fitur-fitur tingkat tinggi (*high-level features*) agar lebih spesifik terhadap morfologi lesi serviks, tanpa mendistorsi representasi fitur tingkat rendah (*low-level features*) yang telah stabil. Mekanisme kontrol yang sama dengan fase pertama tetap diterapkan, namun fokus optimasi primernya dialihkan pada maksimalisasi nilai Sensitivitas (*Recall*).

Tabel 3.6 Konfigurasi *Callback Training*

<i>Callback</i>	Parameter Kunci	Konfigurasi & Justifikasi
EarlyStopping	monitor=val_auc patience=10 restore_best_weights=True	Menghentikan pelatihan jika val_auc tidak membaik 10 <i>epoch</i> . Monitor val_auc (bukan val_recall) mencegah solusi degeneratif di mana model memprediksi semua sampel Positif untuk mendapat <i>Recall</i> =1,0 dengan <i>Specificity</i> =0,0
ReduceLROnPlateau	monitor=val_auc factor=0,5 patience=5 min_lr=1e-7	Mengurangi <i>learning rate</i> 50% jika val_auc stagnan 5 <i>epoch</i> . min_lr=1e-7 mencegah <i>learning rate</i> terlalu kecil yang menghentikan konvergensi efektif
ModelCheckpoint	monitor=val_auc mode=max save_best_only=True	Menyimpan <i>checkpoint</i> hanya saat val_auc mencapai nilai tertinggi baru. Format .keras (Keras 3.x native) dipilih untuk menghindari masalah serialisasi pada format .h5 <i>legacy</i>
TensorBoard	histogram_freq=1	Logging metrik pelatihan per <i>epoch</i> untuk analisis kurva konvergensi dan distribusi aktivasi <i>layer</i>

3.1.3.3 Optimasi Model Menggunakan *Post-Training Quantization*

Model CNN yang telah dilatih pada setiap eksperimen selanjutnya dikonversi ke format TensorFlow Lite (.tflite) dan dioptimasi menggunakan teknik *Post-Training Quantization* (PTQ) dengan mengubah representasi bobot dan aktivasi dari presisi *floating-point* 32-bit (Float32) menjadi bilangan bulat 8-bit (INT8). Secara matematis, proses kuantisasi memetakan nilai real r ke nilai integer q melalui parameter skala S dan *zero-point* Z dengan hubungan $r = S(q - Z)$, di mana S menentukan resolusi pembulatan dan Z memastikan nilai nol pada domain asli tetap direpresentasikan secara tepat pada domain integer.

Pada penelitian ini digunakan skema *full integer quantization*, di mana tidak hanya bobot (*weights*) yang dikuantisasi, tetapi juga aktivasi (*activations*) pada setiap *layer* dikonversi ke INT8. Karena rentang nilai aktivasi bergantung pada data *input* dan tidak dapat diketahui hanya dari bobot model yang sudah dilatih, diperlukan *calibration dataset* berupa sebagian data latih yang dialirkan melalui model untuk mengukur rentang nilai (minimum-maksimum) aktivasi pada setiap *layer*. Rentang ini digunakan untuk menghitung parameter S dan Z yang sesuai, sehingga proses kuantisasi dapat meminimalkan degradasi performa akibat reduksi presisi numerik.

Tidak semua operasi pada arsitektur CNN mendukung representasi INT8 secara penuh; sebagai contoh, fungsi aktivasi HardSwish pada MobileNetV3 tetap dijalankan dalam format Float32 melalui mekanisme *fallback*, agar operasi yang sensitif terhadap presisi tidak dipaksakan melewati jalur integer. Tipe data *input* dan *output* model tetap dipertahankan dalam format Float32 untuk menjaga kompatibilitas antarmuka dengan aplikasi Android, sementara komputasi internal jaringan dijalankan dalam INT8.

Evaluasi komparatif dilakukan antara model Float32 (pra-kuantisasi) dan model INT8 (pasca-kuantisasi) untuk menganalisis *trade-off* antara performa diagnostik dan efisiensi komputasi. Parameter yang dibandingkan meliputi metrik klasifikasi (Sensitivitas, PR-AUC, ROC-AUC, dan *Precision@recall*) serta parameter efisiensi (ukuran model dalam MB dan waktu inferensi dalam milidetik).

3.1.4 Evaluasi Model

Tahap evaluasi dirancang untuk menilai kinerja sistem secara menyeluruh dari dua aspek, yaitu performa klasifikasi diagnostik dan efisiensi komputasi. Evaluasi dilakukan pada setiap kombinasi arsitektur CNN dan strategi *preprocessing*, baik sebelum maupun sesudah penerapan *Post-Training Quantization*.

3.1.4.1 Skema Evaluasi Model

Evaluasi model dibagi ke dalam tiga aspek yang saling melengkapi sebagaimana dirangkum pada Tabel 3.7.

Tabel 3.7 Parameter Evaluasi

Aspek Evaluasi	Parameter	Tujuan Evaluasi
Performa Klasifikasi	Sensitivitas (<i>Recall</i>)	Mengukur kemampuan deteksi kasus <i>Positive</i> meminimalkan <i>False Negative</i> yang berisiko klinis tinggi
	PR-AUC	Metrik paling robust untuk dataset <i>imbalanced</i> ; mengukur keseimbangan <i>Precision-Recall</i> di seluruh <i>threshold</i>
	ROC-AUC	Mengukur kemampuan diskriminatif model secara keseluruhan
	Spesifisitas	Mengukur kemampuan identifikasi kelas <i>Negative</i> meminimalkan <i>False Positive</i>
	<i>Confusion matrix</i>	Analisis distribusi kesalahan klasifikasi antar kelas
Efisiensi Komputasi	Ukuran Model (MB) INT8	Menilai kebutuhan penyimpanan pada perangkat Android kelas menengah
	Waktu Inferensi (ms) INT8	Menilai latensi pemrosesan citra secara <i>offline</i>

Aspek Evaluasi	Parameter	Tujuan Evaluasi
Dampak <i>Quantization</i>	Delta Sensitivitas <i>pre vs post</i>	Menilai degradasi performa diagnostik akibat optimasi INT8
	Delta PR-AUC <i>pre vs post</i>	Menilai perubahan kualitas diskriminatif akibat kuantisasi

3.1.4.2 Evaluasi Performa Klasifikasi Diagnostik

Evaluasi performa klasifikasi dilakukan menggunakan data uji yang bersifat *hold-out* dan tidak terlibat dalam proses pelatihan maupun pemilihan model. Kinerja model diukur menggunakan metrik sensitivitas, spesifisitas, PR-AUC, dan ROC-AUC yang definisi dan formulasi matematisnya telah diuraikan pada Sub-bab 2.2.8. *Confusion matrix* digunakan untuk memvisualisasikan distribusi kesalahan klasifikasi antar kelas dan mengidentifikasi pola kesalahan model secara lebih rinci.

3.1.4.3 Evaluasi Efisiensi Komputasi

Evaluasi efisiensi komputasi dilakukan untuk menilai kelayakan *deployment* model pada perangkat Android kelas menengah. Parameter yang dievaluasi meliputi ukuran file model (.tflite) dalam megabyte (MB) dan waktu inferensi rata-rata dalam milidetik (ms) yang diukur sebagai rata-rata durasi pemrosesan satu citra tes IVA dari tahap *input* hingga keluaran prediksi. Perbandingan efisiensi komputasi dirangkum dalam bentuk tabel komparatif antar model untuk memudahkan analisis *trade-off* antara performa diagnostik dan kebutuhan sumber daya perangkat.

3.1.4.4 Evaluasi Model Terbaik

Penentuan model terbaik menggunakan tiga kriteria berurutan, dimulai dari nilai PR-AUC tertinggi sebagai prioritas utama karena kestabilannya pada dataset medis yang tidak seimbang (*imbalanced*). Selanjutnya, model disaring berdasarkan batas minimum sensitivitas yang aman untuk kebutuhan klinis alat skrining. Di antara model yang memenuhi kedua syarat tersebut, model dengan ukuran berkas dan latensi terendah dipilih untuk mendukung penerapan langsung (*on-device*) pada perangkat Android.

Pendekatan ini memastikan bahwa model yang diintegrasikan ke dalam aplikasi tidak hanya unggul secara statistik, tetapi juga layak digunakan dalam konteks klinis layanan kesehatan primer.

Untuk mengevaluasi efektivitas pendekatan *deep learning* yang diusulkan, model CNN dengan arsitektur terbaik (*best model*) yang dihasilkan dari serangkaian eksperimen akan dikomparasikan secara langsung dengan metode *baseline*. Sebagai pembanding, penelitian ini mengimplementasikan ulang algoritma deteksi dari Harsono et al. (2022) yang merepresentasikan pendekatan *machine learning* tradisional melalui ekstraksi fitur manual dan *Gradient Boosting Classifier*. Untuk memastikan komparasi berlangsung setara (*head-to-head*), algoritma *baseline* dilatih dan diuji menggunakan dataset gabungan yang identik dengan yang digunakan pada model CNN, yakni IARC WHO dan PRIMER.

Evaluasi komparatif antara model CNN terbaik dan model *baseline* secara khusus difokuskan pada metrik Presisi pada tingkat *Recall* 80% (*Precision@recall=80%*), merujuk

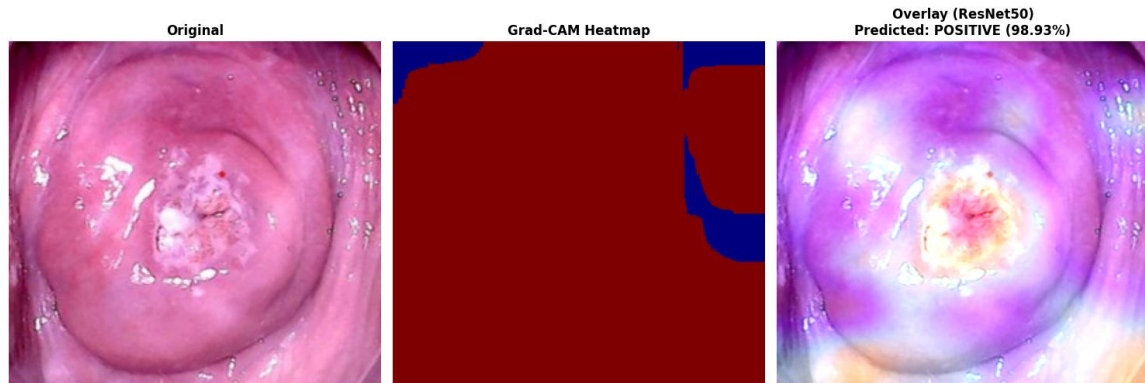
pada standar evaluasi yang ditetapkan dalam penelitian *baseline*. Pemilihan metrik ini bukan tanpa alasan. Dalam konteks skrining medis dini, mematok *recall* pada ambang batas yang tinggi merupakan keharusan klinis, sebab setiap kasus lesi yang luput dari deteksi (*false negative*) berpotensi membawa konsekuensi serius bagi pasien. Dengan ambang *recall* yang telah terkunci pada 80%, metrik presisi pada titik tersebut menjadi indikator yang tepat untuk mengukur seberapa efektif model dalam menekan laju kesalahan diagnosis positif (*false positive*) tanpa harus mengorbankan sensitivitas maupun keamanan proses skrining secara keseluruhan.

3.1.5 *Heatmap Explainable AI*

Pemilihan *layer* konvolusi terakhir sebagai target Grad-CAM didasarkan pada pertimbangan bahwa *layer* tersebut merupakan representasi fitur paling abstrak secara semantik, sekaligus merupakan *layer* terakhir yang masih mempertahankan struktur spasial dua dimensi sebelum data diteruskan ke lapisan *global average pooling* dan klasifikasi. Dengan demikian, peta aktivasi yang dihasilkan dapat dipetakan kembali ke lokasi piksel pada citra asli sambil tetap membawa informasi semantik tingkat tinggi mengenai fitur diskriminatif yang dipelajari model seperti pada Gambar 3.7. Pada arsitektur ResNet-50 yang digunakan sebagai model final dalam penelitian ini, *layer* target Grad-CAM adalah *layer* konvolusi terakhir pada blok residual kelima (*conv5_block3_out*), sesuai dengan pemetaan *layer* target yang telah dijelaskan pada metodologi penelitian.

Visualisasi *heatmap* Grad-CAM diharapkan dapat meningkatkan transparansi sistem dan memberikan interpretabilitas yang intuitif bagi tenaga medis, sehingga keputusan model dapat divalidasi secara klinis dengan mengidentifikasi apakah area yang dianggap relevan oleh model sesuai dengan lokasi anatomi yang memang memiliki signifikansi diagnostik pada pemeriksaan serviks.

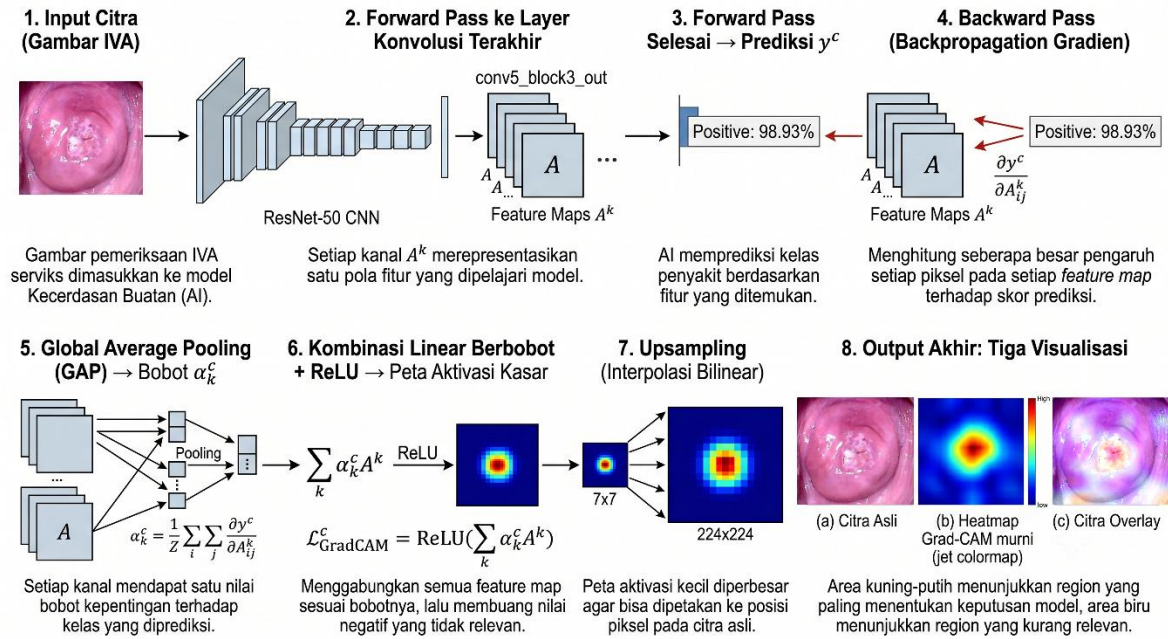
Model terbaik yang telah diidentifikasi melalui tahap evaluasi selanjutnya diintegrasikan dengan metode *Explainable AI* menggunakan teknik *Gradient-weighted Class Activation Mapping* (Grad-CAM), yang berlandaskan pada teori dan formulasi matematis yang telah dipaparkan pada sub-bab 2.2.7. Grad-CAM bekerja dengan menghitung gradien dari skor prediksi terhadap *feature map* pada *layer* konvolusi terakhir jaringan, kemudian menggunakan bobot kepentingan saluran (α_k^c) untuk membentuk kombinasi linear dari peta fitur tersebut. Setelah dinormalisasi dan dilewatkan melalui fungsi aktivasi ReLU, dihasilkan peta aktivasi spasial mentah. Hasil akhirnya berupa *heatmap* yang memvisualisasikan area citra serviks yang paling berkontribusi terhadap keputusan klasifikasi model seperti pada Gambar 3.6.



Gambar 3.5 Contoh Penerapan *Heatmap* pada Foto Tes IVA

Secara teknis, peta aktivasi Grad-CAM $L_{\text{Grad-CAM}}^c$ yang diperoleh diproyeksikan kembali ke ruang piksel citra asli melalui interpolasi bilinear. Proses ini menghasilkan tiga representasi visual yang ditampilkan secara bersamaan: citra asli sebagai referensi, *heatmap* Grad-CAM murni yang menggambarkan distribusi aktivasi model tanpa latar citra, dan citra *overlay* yang menggabungkan kedua representasi tersebut. Pada visualisasi *overlay*, area dengan warna kuning hingga putih menunjukkan region yang paling menentukan prediksi model (memiliki kontribusi positif tertinggi), sementara area biru menandakan region yang tidak relevan secara statistik bagi keputusan klasifikasi.

Pemilihan *layer* konvolusi terakhir sebagai target Grad-CAM didasarkan pada pertimbangan bahwa *layer* tersebut merupakan representasi fitur paling abstrak secara semantik, sekaligus merupakan *layer* terakhir yang masih mempertahankan struktur spasial dua dimensi sebelum data diteruskan ke lapisan *global average pooling* dan klasifikasi. Dengan demikian, peta aktivasi yang dihasilkan dapat dipetakan kembali ke lokasi piksel pada citra asli sambil tetap membawa informasi semantik tingkat tinggi mengenai fitur diskriminatif yang dipelajari model seperti pada Gambar 3.7. Pada arsitektur ResNet-50 yang digunakan sebagai model final dalam penelitian ini, *layer* target Grad-CAM adalah *layer* konvolusi terakhir pada blok residual kelima (*conv5_block3_out*), sesuai dengan pemetaan *layer* target yang telah dijelaskan pada metodologi penelitian.



Gambar 3.6 Proses Kerja *Heatmap* Grad-CAM

Visualisasi *heatmap* Grad-CAM diharapkan dapat meningkatkan transparansi sistem dan memberikan interpretabilitas yang intuitif bagi tenaga medis, sehingga keputusan model dapat divalidasi secara klinis dengan mengidentifikasi apakah area yang dianggap relevan oleh model sesuai dengan lokasi anatomi yang memang memiliki signifikansi diagnostik pada pemeriksaan serviks.

3.1.6 Pengembangan Aplikasi Android

Aplikasi CervicalAI dibuat menggunakan bahasa Kotlin dengan batas minimum Android 7.0 (SDK 24) dan target Android 14 (SDK 34). Seluruh proses inferensi klasifikasi berjalan secara lokal di dalam ponsel tanpa memerlukan koneksi internet. Model CNN hasil pelatihan diekspor dan dikuantisasi ke dalam format INT8 menggunakan TensorFlow Lite untuk mengurangi pemakaian memori dan mempercepat komputasi. Proses inferensi berjalan pada thread latar belakang menggunakan *Kotlin Coroutines* agar aplikasi tetap responsif.

Pada tahap pengambilan citra, sistem menggunakan kamera kustom berbasis Android Jetpack CameraX. Kamera ini memiliki panduan lingkaran visual untuk membantu memposisikan kamera pada serviks. Sebelum citra diproses oleh model, sistem memeriksa tingkat keburaman gambar dengan menghitung varians *Laplacian* dalam kode Kotlin. Selain itu, sistem menggunakan pustaka OpenCV untuk mendeteksi dan memotong area serviks secara otomatis agar sesuai dengan format masukan model.

Selain klasifikasi, aplikasi ini memvisualisasikan area yang memengaruhi keputusan model menggunakan *Class Activation Mapping* (CAM). Peta fitur dari lapisan konvolusi terakhir dikalikan dengan bobot lapisan *Dense* terakhir, lalu di-*upscale* ke ukuran gambar asli. Aplikasi juga dapat menyimpan foto hasil analisis berdampingan dengan gambar *heatmap* ke galeri ponsel menggunakan Android MediaStore API, serta membuat laporan hasil pemeriksaan dalam format PDF.

3.1.7 Pengujian *User Acceptance Test* dan Validasi Klinis

Untuk mengevaluasi kelayakan sistem dalam skenario penggunaan dunia nyata, penelitian ini merancang skema *User Acceptance Testing* (UAT) dan validasi klinis yang melibatkan tenaga medis profesional, yang mencakup dokter spesialis Obstetri dan Ginekologi (Onkologi) serta dokter residen di Rumah Sakit Umum Pusat (RSUP) Hasan Sadikin Bandung. Pengujian ini bertujuan untuk mengukur performa diagnostik aplikasi ketika dihadapkan pada kasus riil di lapangan, sekaligus menilai tingkat penerimaan pengguna terhadap antarmuka dan fungsionalitas sistem.

Secara garis besar, pengujian dibagi menjadi dua tahap utama, yaitu Validasi Diagnostik Klinis (pengujian *head-to-head*) dan Evaluasi Kepuasan Pengguna yang diukur menggunakan instrumen kuesioner berbasis skala *Likert* (1 hingga 5, mewakili Sangat Tidak Setuju hingga Sangat Setuju). Detail dari setiap tahap pengujian dan dimensi evaluasi yang diukur dirangkum pada Tabel 3.8.

Tabel 3.8 Parameter User Acceptance Test dan Validasi Klinis

Tahap Pengujian	Dimensi Evaluasi	Parameter yang Dinilai
Validasi Diagnostik Klinis	Komparasi Hasil (Akurasi <i>On-Device</i>)	Perbandingan langsung (<i>head-to-head</i>) antara diagnosis visual dokter spesialis berdasarkan pemeriksaan IVA (Normal/Abnormal) dengan label prediksi dari aplikasi. Hal ini untuk memvalidasi keandalan sistem pada variasi kondisi klinis riil (termasuk potensi <i>false positive</i> pada kondisi servitis).
Evaluasi Kepuasan Pengguna	C1. Kemudahan Penggunaan Umum	Tingkat pemahaman aplikasi tanpa pelatihan khusus, intuitivitas alur, tampilan profesional, keterbacaan teks, dan kelancaran navigasi.
	C2, C4, C5. Fungsionalitas Kamera & Inspeksi	Responsivitas kamera, fitur <i>overlay</i> (Lingkaran SSK) untuk penentuan posisi, fitur inspeksi (<i>pinch-to-zoom</i> , pan, toggle asli/ <i>heatmap</i>), serta kelancaran proses penyimpanan gambar ke dalam galeri perangkat.
	C6. Kecepatan dan Performa	Stabilitas aplikasi dan kewajaran durasi waktu pemrosesan (<i>loading</i>) selama inferensi model berlangsung.
	D. Integrasi Alur Kerja	Tingkat gangguan aplikasi terhadap prosedur standar IVA, potensi penggunaan oleh tenaga kesehatan non-spesialis, serta manfaat aplikasi dalam edukasi pasien.

Tahap Pengujian	Dimensi Evaluasi	Parameter yang Dinilai
	E. Kepercayaan Hasil Klinis	Tingkat kepercayaan dokter terhadap akurasi aplikasi, konsistensi hasil prediksi dengan temuan klinis, kesediaan untuk menggunakan sistem sebagai alat bantu diagnostik, serta rekomendasi untuk pengembangan lanjutan.

Seluruh data kualitatif dan kuantitatif yang dihimpun dari parameter pengujian pada Tabel 3.6 tersebut selanjutnya dianalisis untuk mengukur skor kepuasan secara keseluruhan serta merumuskan rekomendasi perbaikan sistem di masa mendatang.

3.2 Bahan dan Peralatan yang Digunakan

3.2.1 Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini dirangkum pada Tabel 3.9.

Tabel 3.9 Perangkat Lunak

No	Perangkat Lunak	Fungsi dalam Penelitian
1	Python 3.10+	Bahasa pemrograman utama untuk seluruh <i>pipeline</i> eksperimen
2	TensorFlow 2.x / Keras	Pembangunan, pelatihan, dan <i>fine-tuning</i> arsitektur CNN (ResNet-50, EfficientNet-B0, MobileNetV3)
3	TensorFlow Lite	Konversi model ke format .tflite dan optimasi <i>Post-Training Quantization</i> (INT8)
4	OpenCV	<i>Preprocessing</i> citra: otomatisasi <i>cropping</i> serviks dan ROI <i>masking</i> berbasis kontur
5	OwlViT (HuggingFace)	<i>Zero-shot vision-language object detection</i> untuk lokalisasi lesi serviks (NB-10)
6	Sevia YOLOv11	Model deteksi lesi serviks berbasis YOLO untuk <i>pipeline cropping</i> (NB-11)
7	NumPy & Pandas	Pengolahan <i>array</i> numerik, manajemen dataset, dan <i>logging</i> hasil eksperimen
8	Scikit-learn	Perhitungan metrik evaluasi (Sensitivitas, Spesifisitas, PR-AUC, ROC-AUC) dan <i>stratified split</i>
9	Matplotlib / Seaborn	Visualisasi kurva ROC, <i>precision-recall</i> , <i>confusion matrix</i> , dan perbandingan performa model
10	Grad-CAM Library	Generasi visualisasi <i>heatmap Explainable AI</i> pada model terbaik
11	Android Studio	Pengembangan aplikasi Android dan integrasi model .tflite secara <i>offline</i>

No	Perangkat Lunak	Fungsi dalam Penelitian
12	Jupyter <i>Notebook</i>	Lingkungan eksperimen iteratif untuk seluruh 9 <i>notebook ablation study</i>
13	Kaggle.com	Penyediaan infrastruktur akselerasi <i>hardware</i> GPU untuk memangkas waktu <i>training</i> model CNN, sekaligus sebagai lingkungan <i>hosting dataset</i> konfidensial secara <i>private</i> .

3.2.2 Perangkat Keras

Eksperimen dalam penelitian ini dilakukan menggunakan kombinasi lingkungan lokal dan infrastruktur *cloud*. Pada lingkungan lokal, komputasi dijalankan menggunakan laptop Asus Zephyrus G14 (GA402XV). Prosesor AMD Ryzen 9 7940HS bertugas menangani pra-pemrosesan data dan manajemen alur kerja. Untuk mempercepat konvolusi dan *backpropagation*, sistem memanfaatkan GPU NVIDIA GeForce RTX 4060 (VRAM 8 GB) dengan dukungan CUDA. Kebutuhan memori dipenuhi oleh RAM 16 GB, sementara SSD NVMe 1 TB digunakan untuk mempercepat akses baca-tulis berkas. Seluruh lingkungan pengembangan berbasis Linux dijalankan secara terisolasi melalui Windows Subsystem for Linux (WSL) demi menjaga kompatibilitas penuh dengan TensorFlow dan pustaka *deep learning* lainnya.

Pemanfaatan GPU dengan arsitektur CUDA memungkinkan proses pelatihan model CNN dilakukan secara lebih efisien dibandingkan komputasi berbasis CPU semata, sementara penggunaan memori berkapasitas besar mendukung eksperimen dengan *batch size* yang memadai tanpa mengorbankan stabilitas sistem. Integrasi Windows Subsystem for Linux (WSL) memberikan fleksibilitas dalam penggunaan lingkungan Linux untuk keperluan pelatihan dan eksperimen, sekaligus mempertahankan ekosistem pengembangan Android pada sistem operasi Windows.

Guna mendukung efisiensi dan mempercepat proses penyelesaian rangkaian *ablation study* yang terdiri dari 9 *notebook* eksperimen, proses pelatihan model juga dialihkan ke infrastruktur *cloud computing* Kaggle.com. Platform ini dimanfaatkan untuk mengeksekusi pelatihan model secara paralel dengan dukungan akselerasi *hardware* GPU NVIDIA T4, sekaligus berfungsi sebagai repositori aman untuk penyimpanan *dataset* citra serviks yang bersifat privat dan konfidensial.

3.3 Implementasi

Sub-bab ini menjabarkan detail implementasi teknis dan aspek kode yang dikelompokkan berdasarkan tahapan *pipeline* tugas akhir. Pembahasan mencakup persiapan data, perancangan *model factory*, kuantisasi model, evaluasi, visualisasi *Explainable AI*, dan pembangunan model *baseline*. Di dalam setiap tahapan, dijabarkan implementasi spesifik dari masing-masing *notebook* (NB-1 s.d. NB-9).

3.3.1 Persiapan Dataset dan Augmentasi Data

Tahap ini mencakup pemrosesan citra mentah dari repositori serta pembentukan generator data latih, validasi, dan uji. Masing-masing *notebook* menguji teknik

prapemrosesan (*preprocessing*) spasial dan augmentasi yang berbeda untuk mengisolasi area serviks dari artefak visual seperti bayangan spekulum maupun kulit di tepi citra.

3.3.1.1 NB-2 (Augmentasi Standar) dan NB-3 (CBAM)

Notebook NB-2 dan NB-3 menggunakan citra kolposkopi utuh yang di-*resize* langsung ke dimensi target $224 \times 224 \times 3$ piksel tanpa pemotongan spasial. Augmentasi data diterapkan secara dinamis (*on-the-fly*) selama pelatihan menggunakan ImageDataGenerator dengan parameter sebagaimana ditampilkan pada Kode 3.1. Parameter augmentasi tersebut dirancang untuk mensimulasikan variasi orientasi, skala, dan kondisi pencahayaan yang lazim ditemui pada citra kolposkopi di lapangan.

```

1. AUGMENTATION = {
2.     'rotation_range' : 20,
3.     'width_shift_range' : 0.1,
4.     'height_shift_range' : 0.1,
5.     'zoom_range' : 0.15,
6.     'horizontal_flip' : True,
7.     'vertical_flip' : True,
8.     'brightness_range' : (0.85, 1.15),
9.     'shear_range' : 0.1,
10.    'fill_mode' : 'nearest'
11. }
12.
13. def create_data_generators(dataset_path, img_size, batch_size,
preprocessing_function):
14.     train_datagen = ImageDataGenerator(
15.         preprocessing_function=preprocessing_function,
16.         **AUGMENTATION
17.     )
18.     eval_datagen = ImageDataGenerator(preprocessing_function=preprocessing_function)
19.     train_generator = train_datagen.flow_from_directory(
20.         directory=str(dataset_path / 'train'),
21.         target_size=(img_size, img_size), batch_size=batch_size,
22.         class_mode='binary', classes=['normal', 'positive'],
23.         shuffle=True, seed=42
24.     )
25.     val_generator = eval_datagen.flow_from_directory(
26.         directory=str(dataset_path / 'val'), target_size=(img_size, img_size),
27.         batch_size=batch_size, class_mode='binary', shuffle=False
28.     )
29.     test_generator = eval_datagen.flow_from_directory(
30.         directory=str(dataset_path / 'test'), target_size=(img_size, img_size),
31.         batch_size=batch_size, class_mode='binary', shuffle=False
32.     )
33.     return train_generator, val_generator, test_generator

```

Kode Sumber 3.1 Konfigurasi augmentasi data dan fungsi pembentukan generator (NB-2 & NB-3)

Setiap parameter pada *AUGMENTATION* merepresentasikan satu jenis transformasi geometris atau fotometris yang diterapkan secara acak pada setiap citra di setiap *epoch*. Parameter *rotation_range*=20 memungkinkan citra dirotasi secara acak hingga $\pm 20^\circ$, sementara *width_shift_range* dan *height_shift_range* masing-masing sebesar 0.1 menggeser citra secara horizontal dan vertikal hingga 10% dari dimensi gambar. Parameter *zoom_range*=0.15 menerapkan perbesaran atau pengecilan acak hingga 15%, sedangkan *shear_range*=0.1 menerapkan distorsi *shear* (pergeseran sudut) ringan. Parameter *horizontal_flip* dan *vertical_flip* yang diset *True* memungkinkan citra dicerminkan secara horizontal maupun vertikal, yang relevan karena posisi serviks pada citra dataset tidak

memiliki orientasi baku yang harus dipertahankan. Parameter *brightness_range*=(0.85, 1.15) mensimulasikan variasi pencahayaan kamera kolposkopi dengan mengubah kecerahan citra dalam rentang $\pm 15\%$. Terakhir, *fill_mode*='nearest' menentukan bahwa piksel kosong yang muncul akibat rotasi, pergeseran, atau shear diisi menggunakan nilai piksel tetangga terdekat, sehingga tidak muncul area hitam buatan yang dapat dipelajari secara keliru oleh model sebagai fitur.

Seluruh transformasi ini diterapkan secara *on-the-fly*, artinya citra hasil augmentasi tidak disimpan sebagai file baru, melainkan dihasilkan secara dinamis pada setiap *batch* selama pelatihan berlangsung. Dengan demikian, model menerima variasi citra yang berbeda pada setiap *epoch* meskipun jumlah data dasar tetap sama, sehingga meningkatkan keragaman data efektif tanpa menambah kebutuhan penyimpanan.

Fungsi *flow_from_directory* membaca citra langsung dari struktur folder dataset dan melakukan *resize* ke *target_size* (224×224 piksel) secara otomatis. Parameter *class_mode*='binary' mengonfigurasi label sebagai skalar biner, dengan urutan kelas pada parameter *classes*=['normal', 'positive'] menentukan pemetaan label, yaitu kelas *normal* dipetakan ke label 0 dan kelas *positive* ke label 1. Pemetaan ini konsisten dengan keluaran fungsi aktivasi sigmoid pada lapisan klasifikasi, di mana nilai mendekati 1 mengindikasikan probabilitas kelas *positive*. Pada *train_generator*, parameter *shuffle*=True dengan *seed*=42 memastikan urutan data diacak secara acak namun dapat direproduksi pada setiap eksekusi, sedangkan pada *val_generator* dan *test_generator*, *shuffle*=False digunakan agar urutan prediksi tetap selaras dengan urutan label sebenarnya saat dilakukan evaluasi metrik maupun visualisasi Grad-CAM. Parameter *preprocessing_function* menerapkan normalisasi citra yang spesifik terhadap arsitektur model (misalnya penyesuaian skala dan *mean subtraction* berbasis statistik ImageNet untuk ResNet-50), yang diterapkan secara konsisten baik pada data latih maupun data evaluasi.

Augmentasi hanya diterapkan pada *training set* melalui *train_datagen*, sementara generator data validasi dan uji (*eval_datagen*) hanya menerima fungsi normalisasi model-spesifik tanpa augmentasi guna memastikan evaluasi dilakukan pada distribusi data yang tidak termodifikasi.

3.3.1.2 NB-4 (Pemotongan Spasial Otomatis via OpenCV)

Pada NB-4, citra dipotong secara dinamis menggunakan deteksi kontur eksternal terbesar dan pencocokan lingkaran pembatas minimum (*Minimum Enclosing Circle*) OpenCV sebelum diumpankan ke dalam generator. Teknik ini bertujuan menghilangkan area spekulum dan kulit perifer yang tidak relevan secara klinis, sehingga model dapat memfokuskan perhatiannya pada area serviks. Implementasi fungsi pemotongan ditunjukkan pada Kode 3.2.

```
1. def auto_crop_cervix(image_np: np.ndarray) -> np.ndarray:
2.     img_size = Config.IMG_SIZE
3.     if image_np.dtype != np.uint8:
4.         img_uint8 = np.clip(image_np * 255.0, 0, 255).astype(np.uint8)
5.     else:
6.         img_uint8 = image_np.copy()
7.     gray = cv2.cvtColor(img_uint8, cv2.COLOR_RGB2GRAY)
8.     blurred = cv2.GaussianBlur(gray, (15, 15), 0)
9.     _, thresh = cv2.threshold(blurred, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
10.    contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```

```

11.     if not contours:
12.         fallback = cv2.resize(img_uint8, (img_size, img_size),
interpolation=cv2.INTER_LINEAR)
13.         return fallback.astype(np.float32)
14.     largest = max(contours, key=cv2.contourArea)
15.     (cx, cy), radius = cv2.minEnclosingCircle(largest)
16.     cx, cy, radius = int(cx), int(cy), int(radius)
17.     h, w = img_uint8.shape[:2]
18.     x1, y1 = max(0, cx - radius), max(0, cy - radius)
19.     x2, y2 = min(w, cx + radius), min(h, cy + radius)
20.     if x2 <= x1 or y2 <= y1:
21.         fallback = cv2.resize(img_uint8, (img_size, img_size),
interpolation=cv2.INTER_LINEAR)
22.         return fallback.astype(np.float32)
23.     cropped = img_uint8[y1:y2, x1:x2]
24.     resized = cv2.resize(cropped, (img_size, img_size), interpolation=cv2.INTER_LINEAR)
25.     return resized.astype(np.float32)
26.
27. def _combined_preprocess(img: np.ndarray) -> np.ndarray:
28.     img = auto_crop_cervix(img) # Tahap 1: Pemotongan spasial
29.     img = _arch_preprocess(img) # Tahap 2: Normalisasi model
30.     return img.astype(np.float32)
31.
32. train_datagen = ImageDataGenerator(preprocessing_function=_combined_preprocess,
**AUGMENTATION)
33.

```

Kode Sumber 3.2 Fungsi pemotongan spasial otomatis berbasis kontur OpenCV (NB-4)

Fungsi *auto_crop_cervix* bekerja melalui beberapa tahap pengolahan citra klasik sebelum menghasilkan citra terpotong. Tahap pertama menangani perbedaan format *input*, karena citra yang masuk ke fungsi ini bisa berupa *array* dengan rentang nilai 0 sampai 1 (hasil normalisasi sebelumnya) atau *array uint8* dengan rentang 0 sampai 255. Jika tipe datanya bukan *uint8*, nilai piksel dikalikan 255 dan dipotong ke rentang valid menggunakan *np.clip* agar tidak terjadi *overflow* saat dikonversi.

Citra kemudian diubah ke *grayscale* melalui *cv2.cvtColor*. Konversi ini diperlukan karena proses deteksi kontur dan *thresholding* pada OpenCV bekerja pada citra satu kanal, sementara informasi warna tidak dibutuhkan untuk membedakan area jaringan serviks dari latar belakang gelap di sekitarnya. Setelah itu, *cv2.GaussianBlur* dengan kernel 15x15 diterapkan untuk meredam *noise* dan tekstur halus pada permukaan serviks, sehingga proses *thresholding* selanjutnya tidak terganggu oleh variasi piksel kecil yang tidak relevan.

Pada tahap *thresholding*, fungsi *cv2.threshold* digunakan dengan metode Otsu. Berbeda dari *thresholding* manual yang membutuhkan nilai ambang tetap, metode Otsu menghitung nilai ambang optimal secara otomatis berdasarkan distribusi histogram citra, dengan asumsi terdapat dua kelompok piksel yang cukup terpisah, yaitu area terang (jaringan serviks) dan area gelap (bagian luar lingkaran pandang kolposkop atau bayangan spekulum). Hasilnya adalah citra biner di mana area serviks tampil sebagai region putih.

Selanjutnya, *cv2.findContours* dipanggil dengan parameter *RETR_EXTERNAL* sehingga hanya kontur terluar yang diambil, mengabaikan kontur-kontur kecil di dalamnya seperti pembuluh darah atau lesi yang juga ikut tersegmentasi oleh proses *thresholding*. Dari seluruh kontur yang ditemukan, kontur dengan luas area terbesar dipilih menggunakan *max* dengan key *cv2.contourArea*, dengan asumsi area serviks selalu menjadi *blob* terbesar dalam citra kolposkopi.

Dari kontur terbesar tersebut, `cv2.minEnclosingCircle` menghasilkan titik pusat (cx , cy) dan radius lingkaran terkecil yang dapat melingkupi seluruh kontur. Lingkaran ini kemudian dikonversi menjadi kotak pembatas (*bounding box*) berbentuk persegi dengan sisi sepanjang dua kali radius, berpusat pada (cx , cy). Penggunaan `max(0, ...)` dan `min(w, ...)` atau `min(h, ...)` pada perhitungan $x1$, $y1$, $x2$, $y2$ memastikan koordinat kotak tidak keluar dari batas dimensi citra asli, mengingat lingkaran hasil deteksi bisa saja sebagian berada di luar *frame* jika serviks terletak di tepi citra.

Ada dua jalur *fallback* yang disiapkan untuk menjaga kestabilan *pipeline*. Jika `cv2.findContours` tidak menemukan kontur sama sekali (misalnya pada citra yang terlalu gelap atau homogen sehingga proses *thresholding* gagal memisahkan objek), fungsi langsung melakukan *resize* pada citra asli tanpa pemotongan. *Fallback* kedua dipicu jika hasil perhitungan kotak pembatas tidak valid, yaitu ketika $x2$ lebih kecil atau sama dengan $x1$, atau $y2$ lebih kecil atau sama dengan $y1$. Kondisi ini bisa terjadi pada kasus tepi seperti radius bernilai nol. Kedua *fallback* ini penting agar *pipeline* augmentasi tidak berhenti karena exception saat menemukan citra dengan karakteristik di luar perkiraan.

Jika seluruh proses berjalan normal, citra dipotong sesuai koordinat kotak pembatas menggunakan slicing `array img_uint8[y1:y2, x1:x2]`, kemudian di-*resize* ke ukuran target menggunakan interpolasi bilinear melalui `cv2.INTER_LINEAR`.

Fungsi `_combined_preprocess` menggabungkan `auto_crop_cervix` dengan fungsi normalisasi arsitektur-spesifik `_arch_preprocess` dalam satu alur. Hal yang perlu dicatat adalah urutan eksekusi pada `ImageDataGenerator`. Parameter `preprocessing_function` dijalankan setelah transformasi augmentasi diterapkan, sehingga citra yang masuk ke `auto_crop_cervix` sudah dalam kondisi teraugmentasi, misalnya sudah dirotasi atau di-*zoom* sesuai konfigurasi pada Kode 3.1. Akibatnya, deteksi kontur dan minimum *enclosing circle* pada setiap iterasi pelatihan dijalankan terhadap versi citra yang sedikit berbeda setiap saat, bukan terhadap citra asli yang statis.

3.3.1.3 NB-5 (Pemotongan Spasial Tanpa Augmentasi)

NB-5 menggunakan fungsi pemotongan spasial dinamis `auto_crop_cervix` yang sama dengan NB-4, namun seluruh transformasi augmentasi dinonaktifkan pada *training generator*. Skenario ini dirancang untuk mengisolasi kontribusi teknik pemotongan spasial terhadap performa model, terlepas dari pengaruh augmentasi data.

```
1. train_datagen = ImageDataGenerator(
2.     preprocessing_function=_combined_preprocess # Tanpa parameter augmentasi
3. )
```

Kode Sumber 3.3 Generator pelatihan tanpa augmentasi (NB-5)

3.3.1.4 NB-1 (Kontrol: Citra Mentah Tanpa Intervensi)

NB-1 merupakan skenario kontrol paling mendasar yang tidak menerapkan pemotongan spasial maupun augmentasi data. Citra asli langsung di-*resize* ke dimensi target dan dinormalisasi menggunakan fungsi bawaan arsitektur yang bersangkutan, sebagaimana ditunjukkan pada Kode 3.4.

```
1. train_datagen = ImageDataGenerator(
2.     preprocessing_function=preprocessing_function # Hanya normalisasi bawaan model
```

3.3.1.5 NB-6 (Masking ROI Sirkular via OpenCV)

NB-6 menerapkan masking spasial sirkular alih-alih pemotongan fisik. Piksel di luar area sirkular yang mengelilingi kontur serviks diubah nilainya menjadi nol (hitam), sehingga model tidak terintimidasi oleh *noise* spekulum di tepi citra, namun skala dan resolusi citra asli tetap terjaga. Implementasi ditunjukkan pada Kode 3.5.

```

1. def roi_mask_cervix(image_np: np.ndarray) -> np.ndarray:
2.     img_size = Config.IMG_SIZE
3.     if image_np.dtype != np.uint8:
4.         img_uint8 = np.clip(image_np * 255.0, 0, 255).astype(np.uint8)
5.     else:
6.         img_uint8 = image_np.copy()
7.     gray = cv2.cvtColor(img_uint8, cv2.COLOR_RGB2GRAY)
8.     blurred = cv2.GaussianBlur(gray, (15, 15), 0)
9.     _, thresh = cv2.threshold(blurred, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
10.    contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
11.    if not contours:
12.        fallback = cv2.resize(img_uint8, (img_size, img_size),
interpolation=cv2.INTER_LINEAR)
13.        return fallback.astype(np.float32)
14.    largest = max(contours, key=cv2.contourArea)
15.    (cx, cy), radius = cv2.minEnclosingCircle(largest)
16.    cx, cy, radius = int(cx), int(cy), int(radius)
17.    h, w = img_uint8.shape[:2]
18.    if radius <= 0:
19.        fallback = cv2.resize(img_uint8, (img_size, img_size),
interpolation=cv2.INTER_LINEAR)
20.        return fallback.astype(np.float32)
21.    mask = np.zeros((h, w), dtype=np.uint8)
22.    cv2.circle(mask, (cx, cy), radius, 255, thickness=-1)
23.    masked = cv2.bitwise_and(img_uint8, img_uint8, mask=mask)
24.    resized = cv2.resize(masked, (img_size, img_size), interpolation=cv2.INTER_LINEAR)
25.    return resized.astype(np.float32)
26.

```

Kode Sumber 3.5 Fungsi masking ROI sirkular berbasis kontur (NB-6)

Logika deteksi pada *roi_mask_cervix* identik dengan *auto_crop_cervix* pada baris 1 sampai 16, mulai dari penyesuaian tipe data, konversi ke *grayscale*, Gaussian blur, *thresholding* Otsu, hingga pencarian kontur terbesar dan perhitungan *minEnclosingCircle* untuk mendapatkan *(cx, cy)* dan *radius*. Perbedaan baru muncul setelah titik ini, karena NB-6 tidak lagi menggunakan lingkaran tersebut untuk memotong citra, melainkan untuk membentuk mask.

Pengecekan *fallback* pada NB-6 juga sedikit berbeda dari NB-4. Pada NB-4, validitas hasil deteksi diperiksa lewat koordinat kotak pembatas *(x1, y1, x2, y2)*, sementara pada NB-6 cukup diperiksa apakah *radius* bernilai positif. Jika *radius* nol atau negatif, yang biasanya terjadi saat kontur terbesar hanya berupa satu atau dua piksel akibat kegagalan *thresholding*, fungsi langsung mengembalikan citra asli yang di-*resize* tanpa masking.

Bagian inti dari NB-6 ada pada baris 21 sampai 23. *mask* dibuat sebagai *array* nol berukuran sama dengan citra asli *(h, w)*, lalu *cv2.circle* digambar di atasnya dengan parameter *thickness=-1*. Nilai *thickness* negatif ini punya arti khusus di OpenCV, yaitu

lingkaran digambar dalam keadaan terisi penuh (*filled*), bukan hanya garis tepinya. Hasilnya, area di dalam lingkaran pada *mask* bernilai 255 dan area di luarnya tetap 0.

`cv2.bitwise_and(img_uint8, img_uint8, mask=mask)` kemudian menerapkan mask tersebut ke citra asli. Operasi AND *bitwise* antara citra dengan dirinya sendiri, dibatasi oleh mask, secara efektif mempertahankan nilai piksel asli pada area di dalam lingkaran dan mengubah seluruh piksel di luar lingkaran menjadi nol pada ketiga kanal warna sekaligus, sehingga area tersebut tampil hitam pekat.

Hal yang membedakan hasil akhir NB-6 dari NB-4 adalah dimensi citra sebelum *resize*. Pada NB-4, citra dipotong fisik ke ukuran kotak pembatas sehingga area serviks mengisi sebagian besar atau seluruh frame sebelum di-*resize* ke 224x224, yang secara efektif memberi efek *zoom*. Pada NB-6, dimensi citra hasil *masked* tetap sama dengan dimensi citra asli (*h, w*), hanya saja sebagian besar areanya sudah menjadi hitam. Ketika citra ini di-*resize* ke 224x224 menggunakan `cv2.INTER_LINEAR`, proporsi antara area serviks dan area hitam di sekitarnya tetap terjaga seperti pada citra asli, sehingga tidak ada efek *zoom* seperti pada NB-4.

Konsekuensi dari pendekatan ini adalah munculnya transisi tajam antara piksel jaringan serviks dan blok hitam di sekitarnya pada citra hasil. Transisi tajam semacam ini, terutama setelah melewati augmentasi seperti rotasi atau *shift* yang membuat posisi tepi lingkaran berubah-ubah di setiap *epoch*, menjadi salah satu faktor yang nantinya dibahas terkait performa NB-6 pada hasil eksperimen.

3.3.1.6 NB-7 (Masking ROI Statis Berbasis Pusat Citra)

NB-7 menggunakan pendekatan masking deterministik dengan lingkaran yang dipusatkan secara tetap pada koordinat tengah citra. Jari-jari lingkaran ditentukan berdasarkan rasio tetap terhadap dimensi citra (`Config.ROI_RADIUS_RATIO`). Pendekatan ini dirancang untuk mengatasi kegagalan deteksi kontur adaptif pada citra dengan kontras rendah atau efek *vignette* yang parah, sebagaimana ditunjukkan pada Kode 3.6.

```
1. def roi_mask_cervix_fixed(image_np: np.ndarray) -> np.ndarray:
2.     img_size = Config.IMG_SIZE
3.     if image_np.dtype != np.uint8:
4.         img_uint8 = np.clip(image_np * 255.0, 0, 255).astype(np.uint8)
5.     else:
6.         img_uint8 = image_np.copy()
7.     h, w = img_uint8.shape[:2]
8.     cx, cy = w // 2, h // 2
9.     radius = int(min(h, w) * Config.ROI_RADIUS_RATIO)
10.    mask = np.zeros((h, w), dtype=np.uint8)
11.    cv2.circle(mask, (cx, cy), radius, 255, thickness=-1)
12.    masked = cv2.bitwise_and(img_uint8, img_uint8, mask=mask)
13.    resized = cv2.resize(masked, (img_size, img_size), interpolation=cv2.INTER_LINEAR)
14.    return resized.astype(np.float32)
15.
```

Kode Sumber 3.6 Fungsi masking ROI statis berbasis pusat citra (NB-7)

Pada NB-7, langkah penanganan tipe data di baris 3 sampai 6 sama dengan fungsi-fungsi sebelumnya, tapi setelah itu seluruh tahapan deteksi kontur, *blur*, dan *thresholding* Otsu yang ada pada NB-4 dan NB-6 dihilangkan sama sekali. Tidak ada pemanggilan `cv2.findContours` atau `cv2.minEnclosingCircle` di sini.

Titik pusat lingkaran ditentukan langsung dari geometri citra melalui $cx, cy = w//2, h//2$, yaitu koordinat tengah berdasarkan lebar dan tinggi citra, tanpa melihat isi piksel sama sekali. Radius dihitung dari $\min(h, w) * \text{Config.ROI_RADIUS_RATIO}$. Penggunaan $\min(h, w)$ di sini penting karena memastikan lingkaran yang dihasilkan selalu cukup kecil untuk masuk ke dalam frame pada dimensi manapun, baik citra berorientasi *landscape* maupun *portrait*. *ROI_RADIUS_RATIO* sendiri adalah konstanta yang didefinisikan di *Config*, jadi nilainya tetap untuk seluruh dataset dan tidak menyesuaikan diri terhadap ukuran serviks pada masing-masing citra.

Setelah cx , cy , dan $radius$ didapat, proses pembuatan *mask* dan penerapan *cv2.bitwise_and* pada baris 10 sampai 12 sama persis dengan NB-8, yaitu membuat *array* nol seukuran citra, menggambar lingkaran terisi penuh dengan *thickness=-1*, lalu menerapkannya sebagai mask untuk menghitamkan area di luar lingkaran.

Tidak adanya cabang *fallback* pada fungsi ini sebenarnya adalah konsekuensi langsung dari pendekatan deterministiknya. Pada NB-4 dan NB-6, *fallback* diperlukan karena ada kemungkinan proses deteksi kontur gagal pada citra dengan kontras rendah atau *vignette* berat. NB-7 menghilangkan sumber kegagalan tersebut sepenuhnya dengan tidak melakukan deteksi apapun, sehingga setiap citra, terlepas dari kualitas kontrasnya, akan selalu menghasilkan lingkaran mask pada posisi dan ukuran yang sama.

Konsekuensi dari pendekatan ini adalah asumsi implisit bahwa posisi serviks selalu berada di sekitar titik tengah citra dengan proporsi ukuran yang relatif konsisten antar sampel. Jika asumsi ini tidak terpenuhi pada sebagian data, misalnya serviks yang sedikit bergeser dari pusat atau memiliki ukuran tampak yang lebih besar atau lebih kecil dari rata-rata, area serviks bisa ikut tertutup *mask* atau sebaliknya area non-serviks ikut masuk ke dalam lingkaran. Selain itu, transisi tajam antara piksel jaringan dan blok hitam di tepi lingkaran, sebagaimana sudah disinggung pada NB-6, juga tetap muncul di sini karena mekanisme masking-nya identik.

3.3.1.7 NB-8 (Pemotongan Lesi via OWL-ViT Zero-Shot Detection)

NB-8 mengintegrasikan model deteksi objek *zero-shot* OWL-ViT ([google/owlvit-base-patch32](https://github.com/google/owlvit-base-patch32)) untuk melakukan pemotongan lesi secara *offline* sebelum pelatihan. Deteksi dipandu oleh kueri teks semantik ("*cervix*", "*cervical opening*"), sehingga tidak memerlukan anotasi kotak pembatas manual. Citra hasil pemotongan disimpan ke direktori baru dan selanjutnya dibaca oleh ImageDataGenerator standar, sebagaimana ditunjukkan pada Kode 3.7.

```
1. owlvit_processor = OwlViTProcessor.from_pretrained('google/owlvit-base-patch32')
2. owlvit_model = OwlViTForObjectDetection.from_pretrained('google/owlvit-base-patch32')
3. owlvit_model.eval()
4. _device = 'cuda' if torch.cuda.is_available() else 'cpu'
5. owlvit_model.to(_device)
6.
7. def detect_cervix_bbox(image_pil):
8.     texts = [['cervix', 'cervical opening']]
9.     inputs = owlvit_processor(text=texts, images=image_pil, return_tensors='pt')
10.    inputs = {k: v.to(_device) for k, v in inputs.items()}
11.    with torch.no_grad():
12.        outputs = owlvit_model(**inputs)
13.    target_sizes = torch.tensor([image_pil.height, image_pil.width], device=_device)
14.    results = owlvit_processor.image_processor.post_process_object_detection(
```

```

15.         outputs=outputs, threshold=Config.OWLKIT_THRESHOLD, target_sizes=target_sizes
16.     )
17.     boxes, scores = results[0]['boxes'], results[0]['scores']
18.     if len(scores) == 0:
19.         return None
20.     best_idx = scores.argmax().item()
21.     return boxes[best_idx].cpu().tolist(), scores[best_idx].item()
22.
23. def crop_cervix_owlvit(image_path):
24.     pil_img = PILImage.open(image_path).convert('RGB')
25.     w, h = pil_img.size
26.     result = detect_cervix_bbox(pil_img)
27.     if result is not None:
28.         box, score = result
29.         xmin, ymin, xmax, ymax = box
30.         bw, bh = xmax - xmin, ymax - ymin
31.         xmin = max(0, xmin - bw * Config.OWLKIT_BBOX_MARGIN)
32.         ymin = max(0, ymin - bh * Config.OWLKIT_BBOX_MARGIN)
33.         xmax = min(w, xmax + bw * Config.OWLKIT_BBOX_MARGIN)
34.         ymax = min(h, ymax + bh * Config.OWLKIT_BBOX_MARGIN)
35.         cropped = pil_img.crop((int(xmin), int(ymin), int(xmax), int(ymax)))
36.         cropped_np = np.array(cropped.resize((Config.IMG_SIZE, Config.IMG_SIZE),
PILImage.LANCZOS))
37.         meta = {'method': 'owlvit', 'score': score}
38.     else:
39.         cropped_np = fallback_roi_mask(np.array(pil_img))
40.         meta = {'method': 'fallback', 'score': 0.0}
41.     return cropped_np, meta
42.

```

Kode Sumber 3.7 Deteksi lesi *zero-shot* berbasis OWL-ViT dan fungsi *crop* (NB-8)

Tahap awal pada Kode 3.7 memuat model dan processor OWL-ViT dari *checkpoint google/owlvit-base-patch32*. Pemanggilan *owlvit_model.eval()* menempatkan model dalam mode inferensi, yang menonaktifkan perilaku *layer* seperti *dropout* dan *batch normalization* yang khusus untuk pelatihan. Pengecekan *torch.cuda.is_available()* menentukan apakah komputasi dijalankan di *GPU* atau *CPU*, dan model dipindahkan ke device yang sesuai melalui *owlvit_model.to(_device)*.

Fungsi *detect_cervix_bbox* menerima citra PIL dan kueri teks dalam bentuk *['cervix', 'cervical opening']*. Format list bersarang ini mengikuti konvensi API OWL-ViT, di mana setiap citra dalam batch bisa diberi sekumpulan kueri teks sendiri, meskipun di sini hanya ada satu citra dengan dua kueri. *owlvit_processor* mengubah pasangan teks dan citra ini menjadi tensor yang siap diumpungkan ke model, lalu seluruh *tensor input* dipindahkan ke *_device* yang sama dengan model.

Pemanggilan model dibungkus dalam *torch.no_grad()*, yang memberi tahu PyTorch untuk tidak melacak gradien selama *forward pass*. Karena tahap ini hanya digunakan untuk inferensi dan bukan pelatihan, menonaktifkan pelacakan gradien menghemat memori dan mempercepat komputasi secara signifikan.

Keluaran mentah model kemudian diproses oleh *post_process_object_detection*. Fungsi ini melakukan dua hal sekaligus, yaitu memfilter deteksi berdasarkan ambang kepercayaan *Config.OWLKIT_THRESHOLD* dan mengonversi koordinat kotak pembatas dari skala internal model ke skala piksel citra asli, menggunakan *target_sizes* yang berisi tinggi dan lebar citra sebenarnya. Dari hasil ini, *boxes* dan *scores* diambil. Jika tidak ada deteksi yang melewati ambang *threshold*, *scores* akan kosong dan fungsi mengembalikan *None*. Jika ada,

`scores.argmax()` dipakai untuk memilih deteksi dengan skor kepercayaan tertinggi, dan kotak pembatas beserta skornya dikembalikan.

Fungsi `crop_cervix_owlvit` memanggil `detect_cervix_bbox` dan menangani dua kemungkinan hasil. Jika deteksi berhasil, koordinat `xmin`, `ymin`, `xmax`, `ymax` digunakan untuk menghitung lebar dan tinggi kotak pembatas, `bw` dan `bh`. Kotak ini kemudian diperluas ke segala arah sebesar `bw * Config.OWLKIT_BBOX_MARGIN` dan `bh * Config.OWLKIT_BBOX_MARGIN`, dengan tujuan menyertakan sedikit konteks jaringan di sekitar area yang terdeteksi, mengingat OWL-ViT mungkin menghasilkan kotak yang terlalu ketat pada batas objek. Operasi `max(0, ...)` dan `min(w, ...)` atau `min(h, ...)` memastikan kotak yang sudah diperluas ini tidak melampaui dimensi citra asli. Citra dipotong menggunakan `pil_img.crop` dengan koordinat integer, lalu di-`resize` ke `Config.IMG_SIZE` menggunakan interpolasi `LANCZOS`. Interpolasi ini berbeda dari `cv2.INTER_LINEAR` yang dipakai pada fungsi-fungsi berbasis OpenCV sebelumnya, dan umumnya menghasilkan kualitas `resize` yang lebih baik untuk perbesaran maupun pengecilan citra karena mempertimbangkan area piksel yang lebih luas saat menghitung nilai interpolasi.

Jika `detect_cervix_bbox` mengembalikan `None`, yang berarti OWL-ViT tidak menemukan objek dengan kueri "`cervix`" atau "`cervical opening`" di atas ambang kepercayaan, citra diproses melalui `fallback_roi_mask` yang diterapkan pada `array` numpy dari citra asli. Pada kedua jalur, hasilnya dikembalikan bersama `dictionary meta` yang mencatat metode yang digunakan beserta skor kepercayaannya, sehingga setiap citra dalam dataset bisa dilacak apakah diproses melalui deteksi OWL-ViT atau melalui `fallback`.

Berbeda dari NB-2 sampai NB-7 yang seluruh prapemrosesannya dijalankan secara on-the-fly di dalam `preprocessing_function` milik `ImageDataGenerator`, NB-8 menjalankan proses `crop` ini secara `offline` sebagai tahap terpisah sebelum pelatihan dimulai. Citra hasil `crop` disimpan ke direktori baru, dan `ImageDataGenerator` yang digunakan untuk pelatihan selanjutnya hanya membaca citra-citra yang sudah terpotong tersebut tanpa fungsi prapemrosesan spasial tambahan. Pendekatan ini masuk akal mengingat inferensi OWL-ViT relatif berat dibanding operasi OpenCV sederhana pada NB-4, NB-6, dan NB-7, sehingga lebih efisien dijalankan satu kali per citra di awal dibanding diulang setiap `epoch`.

3.3.1.8 NB-9 (Pemotongan Lesi via SEVIA YOLOv11)

NB-9 menggunakan alur `pre-cropping offline` yang setara dengan NB-8, namun menggantikan model deteksi `zero-shot` generik dengan model SEVIA YOLOv11 yang telah dilatih secara khusus (`fine-tuned`) untuk lokalisasi lesi serviks pada dataset medis. Model diunduh dari repositori HuggingFace Hub. Implementasi ditunjukkan pada Kode 3.8.

```
1. model_path = hf_hub_download(repo_id='naufanzaki/yolov11-sevia', filename='best.pt')
2. yolo_model = YOLO(model_path)
3.
4. def detect_cervix_bbox_yolo(image_path_or_pil):
5.     results = yolo_model.predict(image_path_or_pil, conf=Config.YOLO_CONF_THRESHOLD,
verbose=False)
6.     if len(results) == 0 or len(results[0].boxes) == 0:
7.         return None
8.     boxes = results[0].boxes
9.     best_idx = boxes.conf.argmax()
10.    return boxes.xyxy[best_idx].cpu().tolist(), boxes.conf[best_idx].item()
11.
```

```

12. def crop_cervix_yolo(image_path):
13.     pil_img = PILImage.open(image_path).convert('RGB')
14.     w, h = pil_img.size
15.     result = detect_cervix_bbox_yolo(pil_img)
16.     if result is not None:
17.         box, score = result
18.         xmin, ymin, xmax, ymax = box
19.         bw, bh = xmax - xmin, ymax - ymin
20.         xmin = max(0, xmin - bw * Config.YOLO_BBOX_MARGIN)
21.         ymin = max(0, ymin - bh * Config.YOLO_BBOX_MARGIN)
22.         xmax = min(w, xmax + bw * Config.YOLO_BBOX_MARGIN)
23.         ymax = min(h, ymax + bh * Config.YOLO_BBOX_MARGIN)
24.         cropped = pil_img.crop((int(xmin), int(ymin), int(xmax), int(ymax)))
25.         cropped_np = np.array(cropped.resize((Config.IMG_SIZE, Config.IMG_SIZE),
PILImage.LANCZOS))
26.         meta = {'method': 'yolo', 'score': score}
27.     else:
28.         cropped_np = fallback_roi_mask(np.array(pil_img))
29.         meta = {'method': 'fallback', 'score': 0.0}
30.     return cropped_np, meta
31.

```

Kode Sumber 3.8 Deteksi lesi berbasis SEVIA YOLOv11 dan fungsi *crop* (NB-9)

Baris pertama pada Kode 3.8 menggunakan *hf_hub_download* untuk mengambil berkas bobot *best.pt* dari repositori (SmartClover, 2026) di HuggingFace Hub. Berbeda dari NB-8 yang memuat model OWL-ViT generik langsung dari *checkpoint* publik, model pada NB-9 adalah hasil *fine-tuning* khusus milik penulis sendiri untuk tugas lokalisasi lesi serviks, dan disimpan di HuggingFace Hub agar dapat diunduh ulang secara konsisten tanpa perlu menyertakan file model besar di repositori kode. Setelah file bobot terunduh, *YOLO(model_path)* dari pustaka *ultralytics* memuat arsitektur sekaligus bobot tersebut menjadi objek detektor yang siap dipakai untuk inferensi.

Fungsi *detect_cervix_bbox_yolo* memanggil *yolo_model.predict* dengan parameter *conf=Config.YOLO_CONF_THRESHOLD*. Parameter ini menyaring deteksi dengan skor kepercayaan di bawah ambang tertentu langsung di dalam proses prediksi, sehingga *results* yang dikembalikan hanya berisi deteksi yang sudah lolos filter tersebut. Parameter *verbose=False* sekadar menekan *output* log bawaan *ultralytics* yang biasanya mencetak detail proses inferensi ke konsol.

Pengecekan *len(results) == 0 or len(results[0].boxes) == 0* menangani kasus di mana model tidak menemukan objek apapun pada citra. Objek *results[0].boxes* sendiri adalah struktur data *ultralytics* yang menyimpan seluruh kotak deteksi untuk satu citra, lengkap dengan atribut *conf* untuk skor kepercayaan dan *xyxy* untuk koordinat kotak. Jika ada lebih dari satu deteksi, *boxes.conf.argmax()* dipakai untuk memilih indeks deteksi dengan kepercayaan tertinggi, lalu *boxes.xyxy[best_idx]* dan *boxes.conf[best_idx]* dari indeks tersebut dikembalikan.

Salah satu perbedaan teknis dengan NB-8 ada di sini. Pada OWL-ViT, koordinat keluaran model masih dalam skala internal dan harus dikonversi ke skala piksel citra asli melalui *post_process_object_detection* dengan *target_sizes*. Pada YOLO, atribut *boxes.xyxy* sudah berupa koordinat absolut dalam skala citra *input*, karena *ultralytics* menangani penyesuaian skala ini secara internal saat melakukan prediksi. Akibatnya, fungsi

detect_cervix_bbox_yolo tidak memerlukan langkah konversi skala tambahan seperti pada *detect_cervix_bbox* milik OWL-ViT.

Setelah koordinat kotak diperoleh, fungsi *crop_cervix_yolo* menjalankan langkah yang strukturnya identik dengan *crop_cervix_owlvit*. Lebar dan tinggi kotak *bw*, *bh* dihitung dari selisih *xmax*, *xmin* dan *ymax*, *ymin*, kotak diperluas ke segala arah sebesar proporsi *Config.YOLO_BBOX_MARGIN*, dibatasi agar tidak keluar dari dimensi citra melalui *max(0, ...)* dan *min(w, ...)* atau *min(h, ...)*, lalu citra dipotong dan di-*resize* ke *Config.IMG_SIZE* menggunakan interpolasi *LANCZOS*. Jika tidak ada deteksi sama sekali, *fallback_roi_mask* dipanggil dengan cara yang sama seperti pada NB-8.

Karena kedua fungsi *crop* berbagi struktur logika yang sama, perbedaan utama antara NB-8 dan NB-9 sebenarnya terletak pada model deteksinya, bukan pada cara hasil deteksi tersebut diterapkan ke citra. NB-8 mengandalkan kemampuan *zero-shot* OWL-ViT yang dipandu kueri teks generik, sementara NB-9 mengandalkan model YOLOv11 yang telah dilatih ulang secara *supervised* khusus untuk mengenali bentuk dan posisi lesi serviks, sehingga secara teori dapat menghasilkan lokalisasi yang lebih spesifik terhadap karakteristik dataset yang digunakan.

3.3.2 Model factory dan Transfer learning Dua Fase

Rancangan arsitektur dan skema pelatihan diimplementasikan melalui fungsi *model factory* terpusat yang mendukung tiga jenis arsitektur jaringan saraf konvolusional, yaitu MobileNetV3, ResNet-50, dan EfficientNet-B0. *Transfer learning* dilaksanakan dalam dua fase untuk memaksimalkan konvergensi sekaligus meminimalkan risiko *catastrophic forgetting*.

3.3.2.1 Unified Classification head (NB-2, NB-4 hingga NB-9)

Notebook-notebook ini menggunakan *classification head* standar yang dibangun di atas keluaran spasial rata-rata (*Global Average pooling*) dari *base model* yang telah diinisialisasi dengan bobot ImageNet. Arsitektur kepala klasifikasi terdiri dari lapisan BatchNormalization, Dropout, dan satu *neuron* keluaran dengan aktivasi *sigmoid* untuk klasifikasi biner, sebagaimana ditunjukkan pada Kode 3.9.

```
1. def build_model(model_name, input_shape, hparams, freeze_base=True):
2.     model_config = Config.MODELS_TO_COMPARE[model_name]
3.     base_class = model_config['class']
4.     if model_name.startswith('MobileNet'):
5.         base_model = base_class(input_shape=input_shape, include_top=False,
6.                                 weights='imagenet', include_preprocessing=False)
7.     else:
8.         base_model = base_class(input_shape=input_shape, include_top=False,
9.                                 weights='imagenet')
10.    base_model.trainable = not freeze_base
11.    inputs = keras.Input(shape=input_shape)
12.    x = base_model(inputs, training=False)
13.    x = Layers.GlobalAveragePooling2D()(x)
14.    x = Layers.BatchNormalization()(x)
15.    x = Layers.Dropout(hparams['dropout'])(x)
16.    outputs = Layers.Dense(1, activation='sigmoid')(x)
17.    model = keras.Model(inputs, outputs, name=f'{model_name}_cervical_cancer_binary')
18.    model.compile(
19.        optimizer=keras.optimizers.Adam(learning_rate=hparams['lr_phase1']),
20.        loss=keras.losses.BinaryCrossentropy(from_logits=False),
21.        metrics=['accuracy', keras.metrics.AUC(name='auc'),
```

```

21.         keras.metrics.Precision(name='precision'),
22.         keras.metrics.Recall(name='recall'),
23.         keras.metrics.PrecisionAtRecall(recall=Config.TARGET_RECALL,
24.                                         name='precision_at_recall')]
25.     )
26.     return model
27.

```

Kode Sumber 3.9 Fungsi *model factory* dengan *unified classification head*

Fungsi *build_model* berperan sebagai *model factory* yang menerima *model_name* sebagai parameter untuk menentukan arsitektur mana yang akan dibangun. Baris *model_config = Config.MODELS_TO_COMPARE[model_name]* mengambil konfigurasi arsitektur dari dictionary terpusat, yang di dalamnya berisi referensi kelas model Keras yang sesuai, misalnya ResNet50, EfficientNetB0, atau MobileNetV3Small dari modul *keras.applications*.

Ada percabangan khusus untuk model dengan nama yang dimulai MobileNet. Pada kasus ini, parameter *include_preprocessing=False* disertakan saat memuat *base model*. Secara default, implementasi MobileNetV3 di Keras menyertakan *layer preprocessing* internal yang melakukan rescaling nilai piksel sesuai kebutuhan arsitektur. Karena normalisasi citra pada penelitian ini sudah ditangani secara eksternal melalui *preprocessing_function* pada *ImageDataGenerator*, *layer preprocessing* bawaan tersebut perlu dimatikan agar tidak terjadi normalisasi ganda yang dapat mendistorsi rentang nilai *input*. Arsitektur lain seperti ResNet-50 dan EfficientNet-B0 tidak memiliki parameter *include_preprocessing* ini, sehingga dimuat dengan cara standar pada baris 8.

Parameter *freeze_base* mengontrol baris *base_model.trainable = not freeze_base*. Ketika *freeze_base=True*, seluruh bobot *base model* dibekukan dan tidak diperbarui selama pelatihan, sesuai dengan fase pertama dari skema *transfer learning* dua fase yang dijelaskan pada bagian sebelumnya.

Model dibangun menggunakan functional API Keras. *keras.Input(shape=input_shape)* mendefinisikan tensor *input*, lalu *base_model(inputs, training=False)* meneruskan *input* tersebut melalui *base model*. Argumen *training=False* di sini penting terlepas dari status *trainable base model*, karena argumen ini mengatur perilaku *layer* seperti BatchNormalization di dalam *base model* agar menggunakan statistik *moving average* yang sudah dipelajari saat *pretraining* ImageNet, bukan statistik dari batch data serviks yang sedang diproses. Tanpa pengaturan ini, *layer* BatchNormalization bisa mulai menghitung ulang statistiknya berdasarkan data baru meskipun bobotnya sendiri dibekukan, yang dapat mengganggu representasi fitur yang sudah dipelajari sebelumnya.

Keluaran *base model* berupa *feature map* spasial dengan dimensi tinggi, lebar, dan jumlah kanal kemudian diproses oleh *layers.GlobalAveragePooling2D()*, yang menghitung rata-rata nilai aktivasi pada seluruh posisi spasial untuk setiap kanal, menghasilkan satu vektor fitur dengan panjang sesuai jumlah kanal pada *feature map* terakhir. Vektor ini kemudian dilewatkan ke *layers.BatchNormalization()* untuk menstabilkan distribusi nilai sebelum masuk ke *layers.Dropout(hparams['dropout'])*, di mana nilai *dropout* diambil dari *hyperparameter* masing-masing arsitektur sebagaimana telah dirinci pada Tabel 3.4. Lapisan terakhir, *layers.Dense(1, activation='sigmoid')*, menghasilkan satu nilai keluaran dalam rentang 0 sampai 1 yang merepresentasikan probabilitas kelas *positive*.

Pada proses kompilasi, *keras.optimizers.Adam* digunakan dengan *learning_rate* yang diambil dari *hparams['lr_phase1']*, sesuai dengan *learning rate* fase pertama untuk masing-masing arsitektur pada Tabel 3.4. Fungsi *loss BinaryCrossentropy(from_logits=False)* dipilih karena lapisan keluaran sudah menerapkan aktivasi sigmoid, sehingga *loss* menerima nilai probabilitas secara langsung, bukan logit mentah yang belum melalui fungsi aktivasi.

Daftar *metrics* yang disertakan mencakup *accuracy*, *AUC*, *Precision*, dan *Recall* sebagai metrik klasifikasi umum, ditambah satu metrik khusus *PrecisionAtRecall* dengan parameter *recall=Config.TARGET_RECALL*. Metrik ini mengukur nilai *precision* pada titik operasi di mana *recall* (sensitivitas) model mencapai nilai target tertentu, yang relevan dalam konteks skrining medis karena *recall* yang tinggi diperlukan untuk meminimalkan kasus negatif palsu, sementara *precision* pada titik tersebut menggambarkan seberapa banyak alarm positif yang benar-benar valid.

Karena fungsi *build_model* yang sama dipanggil untuk ketiga arsitektur dengan struktur *head* yang identik, perbedaan hasil akhir antar model dapat lebih diatribusikan pada kemampuan representasi *base model* masing-masing, bukan pada perbedaan desain *classification head*.

3.3.2.2 CBAM Attention Module (NB-3)

Pada NB-3, fungsi *build_model* dimodifikasi dengan menyisipkan modul atensi CBAM (*Convolutional Block Attention Module*) antara keluaran fitur 2D terakhir *base model* dan lapisan *Global Average pooling*. CBAM terdiri dari dua sub-modul sekuensial, yaitu modul atensi saluran (*channel attention*) yang memodelkan ketergantungan antar-kanal fitur, dan modul atensi spasial (*spatial attention*) yang menekankan lokasi fitur yang relevan secara diagnostik. Implementasi keduanya ditunjukkan pada Kode 3.10.

```

1. def channel_attention_module(x, ratio=8):
2.     channel = x.shape[-1]
3.     shared_dense1 = Layers.Dense(channel // ratio, activation='relu',
4.                                   kernel_initializer='he_normal', use_bias=True)
5.     shared_dense2 = Layers.Dense(channel, kernel_initializer='he_normal', use_bias=True)
6.     avg_pool = shared_dense2(shared_dense1(Layers.Reshape((1,1,channel))(Layers.GlobalAveragePooling2D()(x))))
7.     max_pool = shared_dense2(shared_dense1(Layers.Reshape((1,1,channel))(Layers.GlobalMaxPooling2D()(x))))
8.     cbam_feature = Layers.Activation('sigmoid')(Layers.Add()([avg_pool, max_pool]))
9.     return Layers.Multiply()([x, cbam_feature])
10.
11. def spatial_attention_module(x):
12.     avg_pool = ChannelAvgPool()(x)
13.     max_pool = ChannelMaxPool()(x)
14.     concat = Layers.Concatenate(axis=-1)([avg_pool, max_pool])
15.     cbam_feature = Layers.Conv2D(1, kernel_size=7, strides=1, padding='same',
16.                                   activation='sigmoid', kernel_initializer='he_normal',
17.                                   use_bias=False)(concat)
18.     return Layers.Multiply()([x, cbam_feature])
19.
20. def cbam_block(x, ratio=8):
21.     return spatial_attention_module(channel_attention_module(x, ratio))
22.
23. # Integrasi dalam model factory (NB-4)
24. x = base_model(inputs, training=False)
25. x = cbam_block(x, ratio=hparams.get('cbam_ratio', 8)) # Modul atensi CBAM

```

```

26. x = layers.GlobalAveragePooling2D()(x)
27.

```

Kode Sumber 3.10 Implementasi modul atensi CBAM (NB-3)

Modul *channel_attention_module* bertujuan menghasilkan satu bobot skalar untuk setiap kanal pada *feature map* x , yang menggambarkan seberapa penting kanal tersebut secara global. Variabel *channel* diambil dari $x.shape[-1]$, yaitu jumlah kanal pada *feature map input*. Dua lapisan *Dense*, *shared_dense1* dan *shared_dense2*, dibuat sekali di awal dan dipakai bersama (shared) untuk dua jalur komputasi yang berbeda. *shared_dense1* mereduksi dimensi dari *channel* menjadi *channel // ratio* dengan aktivasi relu, membentuk *bottleneck*, sementara *shared_dense2* mengembalikan dimensi tersebut kembali ke *channel* tanpa aktivasi (linear).

Jalur pertama dimulai dari $layers.GlobalAveragePooling2D()(x)$, yang meringkas informasi spasial setiap kanal menjadi satu nilai rata-rata, menghasilkan vektor sepanjang *channel*. Vektor ini di-*reshape* menjadi bentuk $(1, 1, channel)$ agar memiliki dimensi spasial eksplisit, lalu dilewatkan melalui *shared_dense1* dan *shared_dense2* secara berurutan. Jalur kedua melakukan hal yang sama, hanya saja menggunakan *GlobalMaxPooling2D* sebagai pengganti *average pooling*, sehingga menangkap aktivasi maksimum tiap kanal alih-alih rata-ratanya. Kedua jalur ini menggunakan *shared_dense1* dan *shared_dense2* yang sama persis, baik bobot maupun bias-nya, sehingga proses pembelajaran *bottleneck* ini diatur oleh kedua sumber informasi tersebut secara bersamaan.

Hasil dari kedua jalur dijumlahkan melalui $layers.Add()$, lalu dilewatkan ke $layers.Activation('sigmoid')$, menghasilkan *cbam_feature* berupa vektor bernilai antara 0 dan 1 untuk setiap kanal. Operasi $layers.Multiply()(x, cbam_feature)$ kemudian mengalikan setiap kanal pada x dengan bobot kepentingannya masing-masing melalui *broadcasting*, karena *cbam_feature* berdimensi $(1, 1, channel)$ sementara x berdimensi $(H, W, channel)$. Kanal dengan bobot mendekati 1 akan dipertahankan, sedangkan kanal dengan bobot mendekati 0 akan diredam.

Modul *spatial_attention_module* bekerja dengan logika serupa, tetapi pada arah yang berlawanan, yaitu meringkas informasi sepanjang sumbu kanal untuk menghasilkan bobot per lokasi spasial. *ChannelAvgPool* dan *ChannelMaxPool* adalah *layer* kustom yang melakukan agregasi rata-rata dan maksimum di sepanjang sumbu kanal pada setiap posisi (h, w) , masing-masing menghasilkan *feature map* satu kanal berdimensi $(H, W, 1)$. Kedua hasil ini digabungkan melalui $layers.Concatenate(axis=-1)$ menjadi *feature map* dua kanal, yang kemudian diproses oleh $layers.Conv2D$ dengan kernel berukuran 7×7 , padding *same* agar dimensi spasial tidak berubah, aktivasi sigmoid, dan *use_bias=False*. Keluaran konvolusi ini adalah satu *feature map* berdimensi $(H, W, 1)$ berisi nilai antara 0 dan 1, yang merepresentasikan tingkat kepentingan setiap lokasi spasial. Sama seperti pada *channel attention*, $layers.Multiply()(x, cbam_feature)$ mengalikan *feature map input* dengan peta perhatian spasial ini melalui *broadcasting* di sepanjang sumbu kanal.

Fungsi *cbam_block* menyusun kedua sub-modul ini secara sekuensial, di mana keluaran dari *channel_attention_module* langsung menjadi *input* bagi *spatial_attention_module*. Urutan ini berarti fitur pertama-tama disaring berdasarkan kepentingan kanalnya, dan hasil

penyaringan tersebut baru kemudian disaring lagi berdasarkan kepentingan lokasi spasialnya.

Pada integrasi ke *model factory*, *cbam_block* disisipkan tepat setelah *base_model(inputs, training=False)* dan sebelum *layers.GlobalAveragePooling2D()*, sehingga atensi diterapkan langsung pada *feature map* 2D terakhir *base model*, sebelum informasi spasialnya hilang akibat pooling. Parameter *ratio* untuk *bottleneck channel attention* diambil dari *hparams.get('cbam_ratio', 8)*, yang berarti nilai ini bisa dikonfigurasi per eksperimen melalui *hyperparameter*, dengan nilai 8 sebagai default apabila tidak didefinisikan secara eksplisit.

3.3.3 Post-Training Quantization (INT8)

Setelah pelatihan selesai, model Keras .keras dari seluruh eksperimen (NB-2 s.d. NB-9) dikonversi ke format TensorFlow Lite (.tflite) menggunakan *Post-Training Quantization* (PTQ) bilangan bulat 8-bit (INT8). Proses konversi ini bertujuan mereduksi ukuran model dan menurunkan latensi inferensi untuk penerapan pada perangkat *mobile* Android.

Karena *runtime* TFLite pada perangkat Android tidak mendukung operasi gradien dinamis (GradientTape), model Keras dikonfigurasi ulang menjadi model *dual-output* sebelum dikonversi. Keluaran pertama (*classification*) menghasilkan probabilitas *sigmoid* biner, sedangkan keluaran kedua (*conv_features*) mengeksport *feature map* dari lapisan konvolusi terakhir untuk mendukung mekanisme Grad-CAM *on-device*. Implementasi lengkap ditunjukkan pada Kode 3.11.

```
1. LAST_CONV_LAYER_MAP = {
2.     'MobileNet' : 'conv_1',
3.     'resnet'    : 'conv5_block3_out',
4.     'efficientnet': 'top_conv'
5. }
6.
7. def create_multi_output_model(model, model_name):
8.     base_model = model.layers[1]
9.     last_conv_layer = base_model.get_layer(LAST_CONV_LAYER_MAP[model_name])
10.    base_dual = keras.Model(
11.        inputs=base_model.input,
12.        outputs=[last_conv_layer.output, base_model.output]
13.    )
14.    inputs = model.input
15.    conv_features, base_output = base_dual(inputs, training=False)
16.    x = base_output
17.    for layer in model.layers[2:]:
18.        x = layer(x, training=False)
19.    return keras.Model(inputs=inputs, outputs={
20.        'classification': x,
21.        'conv_features' : conv_features
22.    })
23.
24. def quantize_to_int8(model, model_name, val_generator, save_path):
25.     multi_model = create_multi_output_model(model, model_name)
26.     converter = tf.lite.TFLiteConverter.from_keras_model(multi_model)
27.     converter.optimizations = [tf.lite.Optimize.DEFAULT]
28.     def representative_data():
29.         count = 0
30.         for images, _ in val_generator:
31.             yield [images.astype(np.float32)]
32.             count += images.shape[0]
```

```

33.         if count >= 200: break
34.     converter.representative_dataset = representative_data
35.     converter.target_spec.supported_ops = [
36.         tf.lite.OpsSet.TFLITE_BUILTINS_INT8,
37.         tf.lite.OpsSet.TFLITE_BUILTINS # Fallback untuk operasi non-INT8 (misal:
HardSwish)
38.     ]
39.     converter.inference_input_type = tf.float32
40.     converter.inference_output_type = tf.float32
41.     tflite_model = converter.convert()
42.     with open(save_path, 'wb') as f:
43.         f.write(tflite_model)
44.     return tflite_model, len(tflite_model) / (1024*1024)

```

Kode Sumber 3.9 Konversi model ke TFLite INT8 dengan konfigurasi *dual-output*

Sebanyak 200 sampel dari *validation set* digunakan sebagai data kalibrasi untuk menentukan rentang skala kuantisasi INT8. Tipe data masukan dan keluaran dipertahankan dalam format float32 untuk memastikan kompatibilitas lintas perangkat, sementara operasi internal jaringan dikuantisasi ke INT8.

3.3.4 Evaluasi Model

Proses evaluasi model diimplementasikan untuk mengukur performa model *Float32* dan model INT8 hasil kuantisasi secara konsisten. Untuk memastikan akurasi pengukuran latensi inferensi tanpa distorsi dari komputasi latar belakang atau *parallelization overhead*, interpreter TFLite dikonfigurasi dengan mode utas tunggal (*single-thread*). Implementasi fungsi evaluasi ditunjukkan pada Kode 3.12.

```

1. def evaluate_tflite_model(tflite_model, test_generator):
2.     interpreter = tf.lite.Interpreter(
3.         model_content=tflite_model,
4.         experimental_op_resolver_type=tf.lite.experimental.OpResolverType.BUILTIN_WITHOUT_DEFAULT_DELEGATES,
5.         num_threads=1
6.     )
7.     interpreter.allocate_tensors()
8.     input_details = interpreter.get_input_details()
9.     output_details = interpreter.get_output_details()
10.    class_output_idx = next(
11.        i for i, d in enumerate(output_details) if d['shape'][-1] == 1
12.    )
13.    y_true_all, y_pred_all, latencies = [], [], []
14.    for images, labels in test_generator:
15.        for i in range(images.shape[0]):
16.            img = images[i:i+1].astype(np.float32)
17.            interpreter.set_tensor(input_details[0]['index'], img)
18.            t0 = time.perf_counter()
19.            interpreter.invoke()
20.            latencies.append(time.perf_counter() - t0)
21.            output = interpreter.get_tensor(output_details[class_output_idx]['index'])
22.            y_pred_all.append(output[0][0])
23.            y_true_all.append(labels[i])
24.        if len(y_true_all) >= test_generator.samples: break
25.    tn, fp, fn, tp = confusion_matrix(
26.        np.array(y_true_all), (np.array(y_pred_all) > 0.5).astype(int)
27.    ).ravel()
28.    return {
29.        'sensitivity': tp / (tp + fn),
30.        'specificity': tn / (tn + fp),
31.        'avg_latency_ms': np.mean(latencies) * 1000

```

```
32.     }  
33.
```

Kode Sumber 3.12 Fungsi evaluasi model TFLite dengan pengukuran latensi per sampel

Metrik evaluasi utama yang digunakan adalah *sensitivity* (sensitivitas atau *recall*) dan *specificity* (spesifisitas), yang relevan secara klinis untuk skrining kanker serviks. Sensitivitas mengukur kemampuan model mendeteksi kasus positif benar, sementara spesifisitas mengukur kemampuan model mengidentifikasi kasus negatif benar. Rerata latensi inferensi per sampel diukur menggunakan *time.perf_counter()* dengan resolusi sub-milidetik.

3.3.5 Implementasi Grad-CAM

Mekanisme visualisasi *Gradient-weighted Class Activation Mapping* (Grad-CAM) diimplementasikan untuk mengidentifikasi area kontribusi keputusan model pada citra medis. Implementasi memanfaatkan *tf.GradientTape* untuk melacak nilai turunan pada sub-model *dual-output*, sehingga graf komputasi tidak terputus (*disconnected graph*) dalam ekosistem Keras 3.x. Implementasi ditunjukkan pada Kode 3.13.

```
1. def make_gradcam_heatmap(img_array, model, last_conv_layer_name):  
2.     base_model = model.layers[1]  
3.     last_conv_layer = base_model.get_layer(last_conv_layer_name)  
4.     dual_model = keras.Model(  
5.         inputs=base_model.input,  
6.         outputs=[last_conv_layer.output, base_model.output]  
7.     )  
8.     head_layers = model.layers[2:]  
9.     with tf.GradientTape() as tape:  
10.         conv_output, base_output = dual_model(img_array, training=False)  
11.         tape.watch(conv_output)  
12.         x = base_output  
13.         for layer in head_layers:  
14.             x = layer(x, training=False)  
15.             class_channel = x[:, 0]  
16.         grads = tape.gradient(class_channel, conv_output)  
17.         pooled_grads = tf.reduce_mean(grads, axis=(0, 1, 2))  
18.         heatmap = conv_output[0] @ pooled_grads[..., tf.newaxis]  
19.         heatmap = tf.squeeze(heatmap)  
20.         heatmap = tf.maximum(heatmap, 0) / (tf.math.reduce_max(heatmap) + 1e-8)  
21.         return heatmap.numpy()  
22.  
23. # Ekspor bobot Fully-Connected layer untuk integrasi Grad-CAM on-device Android  
24. dense_layer = model.layers[-1]  
25. dense_weights = dense_layer.get_weights()[0]  
26. np.save(f'tflite/{model_name}_dense_weights.npy', dense_weights)  
27.
```

Kode Sumber 3.13 Fungsi Grad-CAM berbasis GradientTape dan ekspor bobot *Dense layer*

Bobot lapisan *fully-connected* terakhir diekspor ke format *.npy* agar dapat diintegrasikan ke dalam aplikasi Android untuk komputasi Grad-CAM *on-device* tanpa memerlukan akses ke graf komputasi Keras secara langsung.

3.3.6 Model *Baseline* Berbasis *Machine learning* Tradisional (NB-10)

Notebook NB-10 mengimplementasikan replikasi arsitektur *machine learning* tradisional berbasis ekstraksi fitur tekstur dan warna manual (*handcrafted features*) yang diusulkan oleh Harsono et al. (2022). Tujuannya adalah menyediakan acuan (*baseline*)

perbandingan yang valid terhadap pendekatan *transfer learning deep learning* yang dieksperimenkan pada NB-2 s.d. NB-9.

3.3.6.1 Penghapusan Pantulan Spekular

Area pantulan spekular (*specular reflection*) atau kilauan cahaya spekulum diidentifikasi berdasarkan kriteria piksel dengan saturasi rendah namun nilai kecerahan kanal hijau (*G*) dan luminansi CIE-Lab (*L*) yang tinggi. Area yang teridentifikasi kemudian direkonstruksi menggunakan algoritma *Telea Inpainting*, sebagaimana ditunjukkan pada Kode 3.14.

```

1. def remove_specular_reflection(img_rgb):
2.     img_hsv = cv2.cvtColor(img_rgb, cv2.COLOR_RGB2HSV)
3.     img_lab = cv2.cvtColor(img_rgb, cv2.COLOR_RGB2LAB)
4.     s_norm = img_hsv[:, :, 1].astype(float) / 255.0
5.     g_norm = img_rgb[:, :, 1].astype(float) / 255.0
6.     l_norm = img_lab[:, :, 0].astype(float) / 255.0
7.     feature_img = np.clip((g_norm + l_norm) / 2.0 - s_norm, 0, 1)
8.     mean_sq = cv2.blur(feature_img**2, (3, 3))
9.     sq_mean = cv2.blur(feature_img, (3, 3))**2
10.    std_norm = cv2.normalize(np.sqrt(np.maximum(0, mean_sq - sq_mean)),
11.                            None, 0, 1, cv2.NORM_MINMAX)
12.    threshold = np.mean(std_norm) + 2 * np.std(std_norm)
13.    mask = cv2.dilate((std_norm > threshold).astype(np.uint8) * 255,
14.                    np.ones((3,3), np.uint8), iterations=1)
15.    return cv2.inpaint(img_rgb, mask, 3, cv2.INPAINT_TELEA)
16.

```

Kode Sumber 3.14 Fungsi penghapusan pantulan spekular via Telea Inpainting (NB-10)

3.3.6.2 Deteksi ROI Berbasis Gaussian *Mixture Model*

Representasi fitur dua dimensi R_d dibentuk dari jarak radial piksel terhadap pusat citra (R) dan nilai kanal warna pada ruang warna CIE-Lab. Algoritma GMM 2-komponen kemudian digunakan untuk memisahkan piksel area serviks (ROI) dari latar belakang berupa lipatan kulit atau spekulum. Kluster yang terpilih sebagai ROI adalah kluster dengan nilai R rata-rata yang lebih kecil, karena serviks umumnya terletak di dekat pusat citra, sebagaimana ditunjukkan pada Kode 3.15.

```

1. def get_ra_features(img_rgb):
2.     h, w = img_rgb.shape[:2]
3.     a_channel = cv2.cvtColor(img_rgb, cv2.COLOR_RGB2LAB)[:, :, 1].astype(float)
4.     X, Y = np.meshgrid(np.arange(w), np.arange(h))
5.     R = np.sqrt((X - w//2)**2 + (Y - h//2)**2)
6.     R_norm = cv2.normalize(R, None, 0, 1, cv2.NORM_MINMAX).flatten()
7.     a_norm = cv2.normalize(a_channel, None, 0, 1, cv2.NORM_MINMAX).flatten()
8.     return np.column_stack((R_norm, a_norm)), R, a_channel
9.
10. def detect_roi_and_crop(img_rgb):
11.     h, w = img_rgb.shape[:2]
12.     features, R, a_channel = get_ra_features(img_rgb)
13.     try:
14.         gmm = GaussianMixture(n_components=2, covariance_type='spherical',
random_state=42)
15.         labels = gmm.fit_predict(features)
16.         cluster_0_R = np.mean(features[labels == 0, 0])
17.         cluster_1_R = np.mean(features[labels == 1, 0])
18.         roi_label = 0 if cluster_0_R < cluster_1_R else 1
19.         mask = (labels == roi_label).reshape((h, w)).astype(np.uint8) * 255
20.         contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
21.         if contours:

```

```

22.         x, y, bw, bh = cv2.boundingRect(max(contours, key=cv2.contourArea))
23.         cropped = img_rgb[y:y+bh, x:x+bw]
24.     else:
25.         cropped = img_rgb
26.     except Exception:
27.         cropped = img_rgb
28.         mask = np.ones((h, w), dtype=np.uint8) * 255
29.         resized = cv2.resize(cropped, (Config.IMG_SIZE, Config.IMG_SIZE),
interpolation=cv2.INTER_LINEAR)
30.         mask_resized = cv2.resize(mask, (Config.IMG_SIZE, Config.IMG_SIZE),
interpolation=cv2.INTER_NEAREST)
31.     return resized, mask_resized
32.

```

Kode Sumber 3.15 Deteksi ROI serviks menggunakan Gaussian Mixture Model (NB-10)

3.3.6.3 Ekstraksi Fitur Warna dan Tekstur

Dari area dalam masker ROI yang berhasil disegmentasi, diekstraksi 12 fitur warna kustom (rata-rata dan standar deviasi dari kanal R, G, B, G/R, B/R, dan V) serta 63 fitur tekstur mencakup GLCM, NGTDM, GLSZM, DWT, dan LBP menggunakan pustaka pyfeats. Seluruh fitur digabungkan menjadi vektor fitur tunggal dimensi 75 yang digunakan sebagai masukan model klasifikasi tradisional, sebagaimana ditunjukkan pada Kode 3.16.

```

1. def extract_color_features(img_rgb, mask):
2.     bool_mask = mask > 0
3.     if not np.any(bool_mask):
4.         bool_mask = np.ones_like(mask, dtype=bool)
5.     R = img_rgb[:, :, 0].astype(float)
6.     G = img_rgb[:, :, 1].astype(float)
7.     B = img_rgb[:, :, 2].astype(float)
8.     V = cv2.cvtColor(img_rgb, cv2.COLOR_RGB2HSV)[ :, :, 2].astype(float)
9.     R_safe = np.where(R == 0, 1e-5, R)
10.    channels = {'R': R, 'G': G, 'B': B, 'GoR': G/R_safe, 'BoR': B/R_safe, 'V': V}
11.    return {f'{stat}_{name}': func(ch[bool_mask])
12.            for name, ch in channels.items()
13.            for stat, func in [('mean', np.mean), ('std', np.std)]}
14.
15. def extract_texture_features(img_rgb, mask):
16.     img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_RGB2GRAY)
17.     features = {}
18.     try:
19.         for f_vals, f_labels in [
20.             pyfeats.glcm_features(img_gray, ignore_zeros=True),
21.             pyfeats.ngtdm_features(img_gray, mask, d=1),
22.             pyfeats.glszm_features(img_gray, mask),
23.             pyfeats.dwt_features(img_gray, mask, wavelet='haar', levels=1),
24.             pyfeats.lbp_features(img_gray, mask, P=[8], R=[1])
25.         ]:
26.             features.update(dict(zip(f_labels, f_vals)))
27.     except Exception as e:
28.         print(f'Error in texture extraction: {e}')
29.     return features

```

Kode Sumber 3.16 Fungsi ekstraksi fitur warna dan tekstur handcrafted (NB-10)

3.3.7 Implementasi Model *Baseline*

Implementasi model *baseline* bertujuan untuk mereplikasi pendekatan *machine learning* tradisional yang diusulkan oleh Harsono et al. (2022), sebagai metrik pembandingan (*head-to-head*) untuk mengukur efektivitas arsitektur CNN yang diusulkan dalam penelitian ini. Implementasi *baseline* mencakup tiga tahapan utama, yaitu pra-pemrosesan citra,

ekstraksi fitur manual, dan klasifikasi menggunakan algoritma *machine learning* konvensional.

Pada tahap pra-pemrosesan, citra terlebih dahulu menjalani penghapusan pantulan spekular (*specular reflection removal*) menggunakan kombinasi kanal warna S (HSV), G (RGB), dan L (CIE-Lab). Selanjutnya, *Region of Interest* (ROI) dideteksi menggunakan algoritma *Gaussian Mixture Model* (GMM) pada ruang warna Ra (jarak dari titik tengah citra dan kanal *a* CIE-Lab). Implementasi kedua tahapan ini ditunjukkan pada Kode Sumber 3.8.

```

1. # Fungsi 1: Penghapusan Pantulan Spekular
2. def remove_specular_reflection(img_rgb):
3.     # Ekstraksi kanal warna: S (HSV), G (RGB), L (CIE-Lab)
4.     img_hsv = cv2.cvtColor(img_rgb, cv2.COLOR_RGB2HSV)
5.     img_lab = cv2.cvtColor(img_rgb, cv2.COLOR_RGB2LAB)
6.     # Representasi fitur: rerata G dan L dikurangi saturasi S
7.     feature_img = (img_rgb[:, :, 1]/255.0 + img_lab[:, :, 0]/255.0)/2.0 -
img_hsv[:, :, 1]/255.0
8.     feature_img = np.clip(feature_img, 0, 1)
9.     # Filter deviasi standar & inpainting untuk menghapus glare
10.    mean_sq = cv2.blur(feature_img**2, (3, 3))
11.    sq_mean = cv2.blur(feature_img, (3, 3))*2
12.    std_norm = cv2.normalize(np.sqrt(np.maximum(0, mean_sq - sq_mean)), None, 0, 1,
cv2.NORM_MINMAX)
13.    threshold = np.mean(std_norm) + 2 * np.std(std_norm)
14.    mask = cv2.dilate((std_norm > threshold).astype(np.uint8) * 255,
15.                    np.ones((3,3), np.uint8), iterations=1)
16.    return cv2.inpaint(img_rgb, mask, 3, cv2.INPAINT_TELEA)
17. # Fungsi 2: Deteksi ROI dan Pemotongan Citra
18. def detect_roi_and_crop(img_rgb):
19.     # Fitting GMM 2 komponen pada ruang warna Ra
20.     features, R, a_channel = get_ra_features(img_rgb)
21.     gmm = GaussianMixture(n_components=2, covariance_type='spherical', random_state=42)
22.     labels = gmm.fit_predict(features)
23.     # Crop berbasis bounding box kontur ROI, lalu resize ke 200x200
24.     ... # (logika deteksi kontur masking ROI)
25.     return cv2.resize(cropped, (200, 200))
26.

```

Kode Sumber 3.17 Pra-pemrosesan dan Deteksi ROI Model *Baseline*

Setelah ROI diperoleh, sistem mengekstraksi dua kelompok fitur secara manual. Kelompok pertama terdiri dari 12 fitur warna, yang meliputi nilai rerata dan deviasi standar pada ruang warna RGB, HSV, serta rasio antar-kanal warna. Kelompok kedua terdiri dari 63 fitur tekstur yang diekstraksi menggunakan pustaka PyFeats, mencakup lima metode analisis: *Gray-Level Co-occurrence Matrix* (GLCM), *Neighbourhood Gray Tone Difference Matrix* (NGTDM), *Gray-Level Size Zone Matrix* (GLSZM), *Discrete Wavelet Transform* (DWT), dan *Local Binary Pattern* (LBP). Seluruh fitur yang diekstraksi kemudian menjalani dua tahap reduksi: penghapusan variabel multikolinear dengan ambang batas korelasi Pearson di atas 0,90 dan normalisasi distribusi menggunakan Z-score. Implementasi tahapan ekstraksi, reduksi, dan normalisasi fitur ditunjukkan pada Kode Sumber 3.18.

```

1. def extract_texture_features(img_rgb, mask):
2.     img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_RGB2GRAY)
3.     # Ekstraksi lima kelompok fitur tekstur via PyFeats
4.     glcm_f, glcm_l = pyfeats.glcm_features(img_gray, ignore_zeros=True)
5.     ngtdm_f, ngtdm_l = pyfeats.ngtdm_features(img_gray, mask, d=1)
6.     glszm_f, glszm_l = pyfeats.glszm_features(img_gray, mask)

```

```

7.     dwt_f,   dwt_l   = pyfeats.dwt_features(img_gray, mask, wavelet='haar', levels=1)
8.     lbp_f,   lbp_l   = pyfeats.lbp_features(img_gray, mask, P=[8], R=[1])
9.     ... # (Agregasi ke dalam array fitur)
10. # Reduksi multikolinearitas (threshold korelasi Pearson > 0.90)
11. corr_matrix = X_all.corr().abs()
12. upper      = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(bool))
13. to_drop    = [col for col in upper.columns if any(upper[col] > 0.90)]
14. X_reduced  = X_all.drop(columns=to_drop)
15. # Normalisasi Z-score
16. scaler     = StandardScaler()
17. X_train    = scaler.fit_transform(X_train_raw)
18.

```

Kode Sumber 3.18 Ekstraksi, Reduksi, dan Normalisasi Fitur Manual

Pada tahap klasifikasi, matriks fitur yang telah direduksi digunakan untuk melatih dan mengevaluasi sembilan algoritma *machine learning* konvensional secara komparatif. Evaluasi difokuskan pada optimalisasi metrik Presisi dengan kendala *Recall* minimum sebesar 80% (*PrecisionAtRecall(0.8)*), sesuai dengan prioritas klinis untuk meminimalkan rasio *false negative* pada deteksi lesi prakanker. Model dengan performa terbaik pada metrik tersebut selanjutnya dianalisis tingkat interpretabilitas variabelnya menggunakan skema SHAP (*SHapley Additive exPlanations*) untuk mengidentifikasi fitur-fitur yang paling berkontribusi terhadap keputusan klasifikasi. Implementasi tahapan klasifikasi dan analisis kepentingan fitur ditunjukkan pada Kode Sumber 3.19.

```

1. # Evaluasi komparatif 9 algoritma dengan kendala Recall minimum 0.8
2. target_recall = 0.8
3. results = []
4. for name, clf in classifiers.items():
5.     clf.fit(X_train, y_train)
6.     y_prob = clf.predict_proba(X_test)[: , 1]
7.     precisions, recalls, _ = precision_recall_curve(y_test, y_prob)
8.     # Kalkulasi Precision pada Recall >= target
9.     precision_at_target = max((p for p, r in zip(precisions, recalls) if r >=
target_recall), default=0)
10.    results.append({'Algorithm': name, f'PrecisionAtRecall({target_recall})':
precision_at_target})
11. # Analisis interpretabilitas fitur menggunakan SHAP TreeExplainer
12. explainer = shap.TreeExplainer(best_clf)
13. shap_values = explainer.shap_values(X_test)
14. shap.summary_plot(shap_values, X_test, show=False)
15.

```

Kode Sumber 3.19 Klasifikasi Komparatif dan Analisis Kepentingan Fitur (SHAP)

3.3.8 Implementasi Prapemrosesan Android (CervixCropper)

Aplikasi Android menggunakan kelas *CervixCropper.kt* dengan pustaka OpenCV untuk menyelaraskan prapemrosesan citra dengan model pelatihan. Kelas ini memotong area serviks secara otomatis melalui konversi skala keabu-abuan, filter Gaussian blur, pengambungan Otsu, pencarian kontur terluas, dan penentuan lingkaran pembatas terkecil. Sistem mengubah ukuran potongan gambar menjadi 224x224 piksel. Jika kontur tidak ditemukan, gambar asli langsung diubah ukurannya sebagai langkah cadangan. Implementasi fungsi pemotongan otomatis ini ditunjukkan pada Kode Sumber 3.20.

```

1. object CervixCropper {
2.     fun autoCropCervix(inputBitmap: Bitmap, imgSize: Int = 224): Bitmap {
3.         val srcMat = Mat()
4.         Utils.bitmapToMat(inputBitmap, srcMat)
5.         val rgbMat = Mat()

```

```

6.         val grayMat      = Mat()
7.         val blurredMat = Mat()
8.         val threshMat   = Mat()
9.         try {
10.             Imgproc.cvtColor(srcMat, rgbMat, Imgproc.COLOR_RGBA2RGB)
11.             Imgproc.cvtColor(rgbMat, grayMat, Imgproc.COLOR_RGB2GRAY)
12.             Imgproc.GaussianBlur(grayMat, blurredMat, Size(15.0, 15.0), 0.0)
13.             Imgproc.threshold(blurredMat, threshMat, 0.0, 255.0, Imgproc.THRESH_BINARY
or Imgproc.THRESH_OTSU)
14.             val contours = mutableListOf<MatOfPoint>()
15.             val hierarchy = Mat()
16.             try {
17.                 Imgproc.findContours(threshMat, contours, hierarchy,
Imgproc.RETR_EXTERNAL, Imgproc.CHAIN_APPROX_SIMPLE)
18.                 if (contours.isEmpty()) {
19.                     return resizeToBitmap(rgbMat, imgSize)
20.                 }
21.                 val largest = contours.maxByOrNull { Imgproc.contourArea(it) } ?:
return resizeToBitmap(rgbMat, imgSize)
22.                 val center = Point()
23.                 val radiusArr = FloatArray(1)
24.                 val largestFloat = MatOfPoint2f(*largest.toArray())
25.                 try {
26.                     Imgproc.minEnclosingCircle(largestFloat, center, radiusArr)
27.                 } finally {
28.                     largestFloat.release()
29.                 }
30.                 val cx = center.x.toInt()
31.                 val cy = center.y.toInt()
32.                 val radius = radiusArr[0].toInt()
33.                 val h = rgbMat.rows()
34.                 val w = rgbMat.cols()
35.                 val x1 = maxOf(0, cx - radius)
36.                 val y1 = maxOf(0, cy - radius)
37.                 val x2 = minOf(w, cx + radius)
38.                 val y2 = minOf(h, cy + radius)
39.                 if (x2 <= x1 || y2 <= y1) {
40.                     return resizeToBitmap(rgbMat, imgSize)
41.                 }
42.                 val roi = Rect(x1, y1, x2 - x1, y2 - y1)
43.                 val croppedMat = Mat(rgbMat, roi)
44.                 try {
45.                     return resizeToBitmap(croppedMat, imgSize)
46.                 } finally {
47.                     croppedMat.release()
48.                 }
49.             } finally {
50.                 contours.forEach { it.release() }
51.                 hierarchy.release()
52.             }
53.         } finally {
54.             srcMat.release()
55.             rgbMat.release()
56.             grayMat.release()
57.             blurredMat.release()
58.             threshMat.release()
59.         }
60.     }
61. }
62.

```

Kode Sumber 3.20 Prapemrosesan *CervixCropper*

3.3.9 Implementasi pada Aplikasi Android

Model ResNet-50 INT8 yang telah dikonversi ke format TensorFlow Lite (.tflite) beserta bobot lapisan klasifikasi terakhir (.npy) dimasukkan ke dalam aset aplikasi Android. Pengujian aplikasi dilakukan secara lokal pada *smartphone* uji. Model ResNet-50 INT8 yang telah dikonversi ke format TensorFlow Lite (.tflite) beserta bobot lapisan klasifikasi terakhir yang diekspor dalam format JSON (resnet_dense_weights.json) dimasukkan ke dalam direktori aset (assets/) aplikasi Android. Aplikasi CervicalAI dikembangkan secara *native* menggunakan bahasa pemrograman Kotlin dengan minimum SDK 24 (Android 7.0 Nougat) dan target SDK 34 (Android 14), serta memanfaatkan arsitektur ViewBinding untuk pengelolaan antarmuka. Aplikasi dapat berjalan secara *offline* sepenuhnya tanpa koneksi internet, di mana seluruh proses inferensi, visualisasi CAM, dan ekspor laporan dieksekusi secara *on-device*.

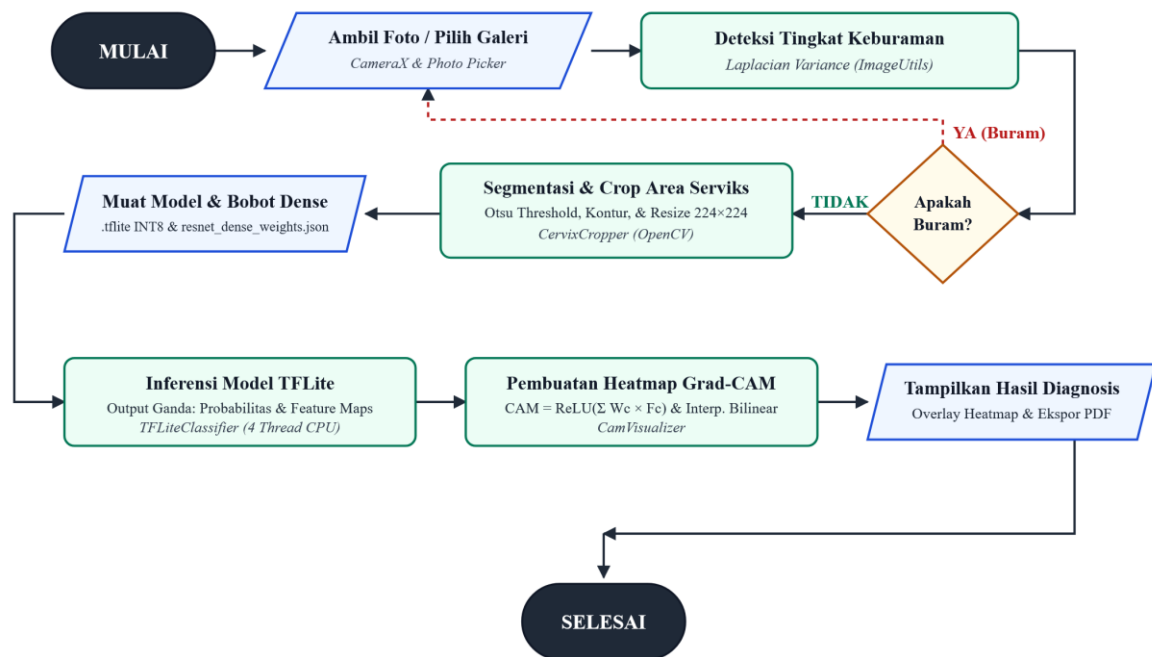
3.3.9.1 Arsitektur dan Komponen Aplikasi

CervicalAI memiliki tiga *Activity* dan lima kelas pendukung untuk memproses gambar dari kamera hingga menampilkan hasil. Tabel 4.4 menjelaskan seluruh komponen tersebut.

Tabel 4.4 Komponen Aplikasi Android CervicalAI

No	Komponen	Jenis	Fungsi
1	MainActivity	<i>Activity</i>	Menampilkan halaman utama dan hasil diagnosis
2	CameraActivity	<i>Activity</i>	Menyediakan kamera khusus dengan panduan lingkaran serviks
3	ImageInspectorActivity	<i>Activity</i>	Menampilkan gambar penuh untuk <i>zoom</i> dan geser
4	TFLiteClassifier	<i>Utility</i>	Memuat dan menjalankan model TensorFlow Lite
5	CamVisualizer	<i>Utility</i>	Membuat <i>heatmap</i> Grad-CAM
6	CervixCropper	<i>Utility</i>	Memotong area serviks secara otomatis menggunakan OpenCV
7	ImageUtils	<i>Utility</i>	Mengubah ukuran gambar, melakukan normalisasi, dan mengukur tingkat keburaman
8	CameraOverlayView	<i>Custom View</i>	Menggambar garis panduan lingkaran serviks pada layar kamera

Alur kerja aplikasi berjalan dalam beberapa tahap. Pertama, pengguna mengambil foto melalui kamera kustom atau memilih gambar dari galeri. Kedua, sistem mendeteksi keburaman gambar dengan rumus varians Laplacian. Ketiga, sistem memotong area serviks secara otomatis menggunakan modul CervixCropper berbasis OpenCV dan mengubah ukurannya menjadi 224×224 piksel, lalu melakukan normalisasi sesuai rata-rata saluran warna BGR (103.939, 116.779, 123.68). Keempat, model TFLite menjalankan inferensi untuk menghasilkan nilai probabilitas dan peta fitur. Terakhir, sistem mengalikan bobot *Dense layer* dengan peta fitur untuk membuat *heatmap* CAM, lalu menampilkannya pada layar. Berikut adalah penjelasan rinci mengenai setiap halaman pada aplikasi CervicalAI.



Gambar 3.7 Diagram Alir Sistem Aplikasi CervicalAI dari Pengambilan Gambar hingga Ekspor Laporan Diagnosis

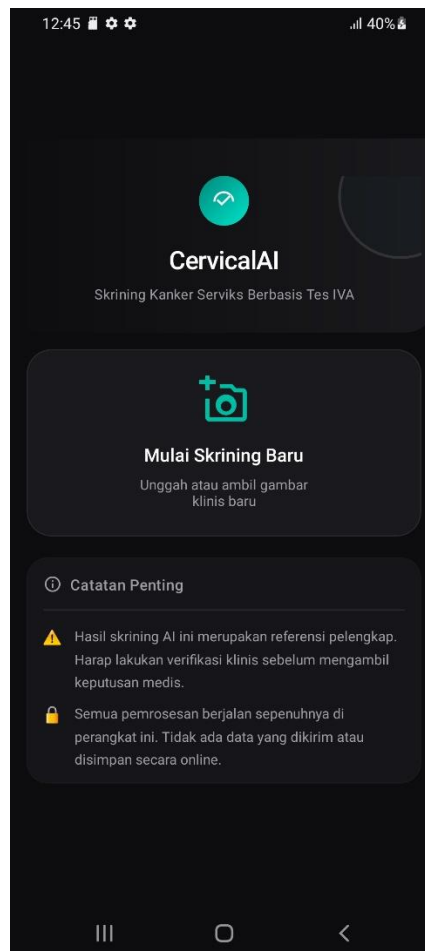
Untuk memperjelas visualisasi logika sistem pada Gambar 3.7, standarisasi simbol diagram alir yang merepresentasikan karakteristik fungsional dari tiap tahapan komputasi. Simbol *Terminator* berbentuk oval digunakan untuk menegaskan batasan operasional awal dan akhir sistem, yakni saat inialisasi aplikasi (MULAI) hingga proses terminasi berkas (SELESAI). Simbol *Input/Output (I/O)* berbentuk jajar genjang diaplikasikan secara spesifik pada fase penanganan berkas dan interaksi data eksternal, meliputi akuisisi citra digital melalui *CameraX* atau *Android Photo Picker*, pemuatan (*loading*) biner graf berbobot model .tflite dan metadata penyeimbang .json dari direktori aset lokal, serta visualisasi akhir matriks probabilitas diagnosis beserta opsi pengeksporan dokumen laporan PDF.

Selanjutnya, komponen operasional yang melibatkan manipulasi data intrinsik atau transformasi struktur *array* direpresentasikan melalui simbol *Process* berbentuk persegi panjang. Tahapan ini mencakup kalkulasi matematika varians matriks kernel Laplacian untuk memvalidasi ketajaman fokus lensa (*blur assessment*), deretan segmentasi morfologi OpenCV (konversi *grayscale*, *Gaussian filter*, segmentasi biner Otsu, pencarian kontur ekstrem, dan ekstraksi batas lingkaran terkecil), inferensi komputasi paralel *Multi-Output*

pada *core interpreter* tensor, hingga perkalian linier *weights layer dense* $\Sigma_c W_c \times F(h, w, c)$ melalui aktivasi ReLU untuk restrukturisasi visualisasi peta fitur Grad-CAM. Alur transisi keputusan diakomodasi oleh simbol *Decision* berbentuk belah ketupat (*rhombus*) guna menguji nilai ambang batas varians keburaman gambar. Apabila citra diidentifikasi mengalami kegagalan fokus (kondisi YA), sistem memicu interupsi (*feedback loop*) berupa instruksi penolakan (*reject snapshot*) berbasis garis putus-putus merah yang memaksa pengguna kembali ke state awal perekaman objek, sedangkan kondisi optimal (kondisi TIDAK) mengizinkan *pipeline* data mengalir secara linier ke sub-sistem prapemrosesan lanjutan.

3.3.9.2 Halaman Beranda

Halaman beranda adalah tampilan awal saat pengguna membuka aplikasi. Desain halaman menggunakan tema gelap dengan tata letak sederhana agar pengguna mudah mengakses menu skrining. Gambar 3.8 menunjukkan tampilan halaman ini.



Gambar 3.8 Tampilan Halaman Beranda Aplikasi CervicalAI

Halaman beranda terdiri dari komponen-komponen berikut:

1. *Header*: Bagian atas halaman yang menampilkan nama aplikasi "CervicalAI" dan keterangan fungsinya.

2. Menu Mulai Skrining: Tombol di tengah layar untuk memulai pemeriksaan. Ketika ditekan, tombol ini membuka pilihan untuk mengambil foto baru melalui kamera atau memilih gambar yang ada di galeri ponsel. Pemilihan dari galeri menggunakan *Android Photo Picker* untuk menjaga keamanan data tanpa memerlukan izin akses berkas.
3. Catatan Peringatan: Bagian bawah halaman yang menampilkan informasi penting. Informasi tersebut berisi penegasan bahwa hasil analisis aplikasi adalah alat bantu pelengkap yang memerlukan verifikasi medis, serta keterangan bahwa data diproses di dalam ponsel tanpa jaringan internet.

3.3.9.3 Halaman Kamera

Halaman kamera pada aplikasi ini berfungsi untuk pengambilan gambar serviks yang menggunakan pustaka *Android Jetpack CameraX*. Tampilan kamera ini disesuaikan untuk memudahkan pengambilan foto dalam kondisi pemeriksaan medis. Gambar 4.4 menunjukkan tampilan halaman kamera ini.



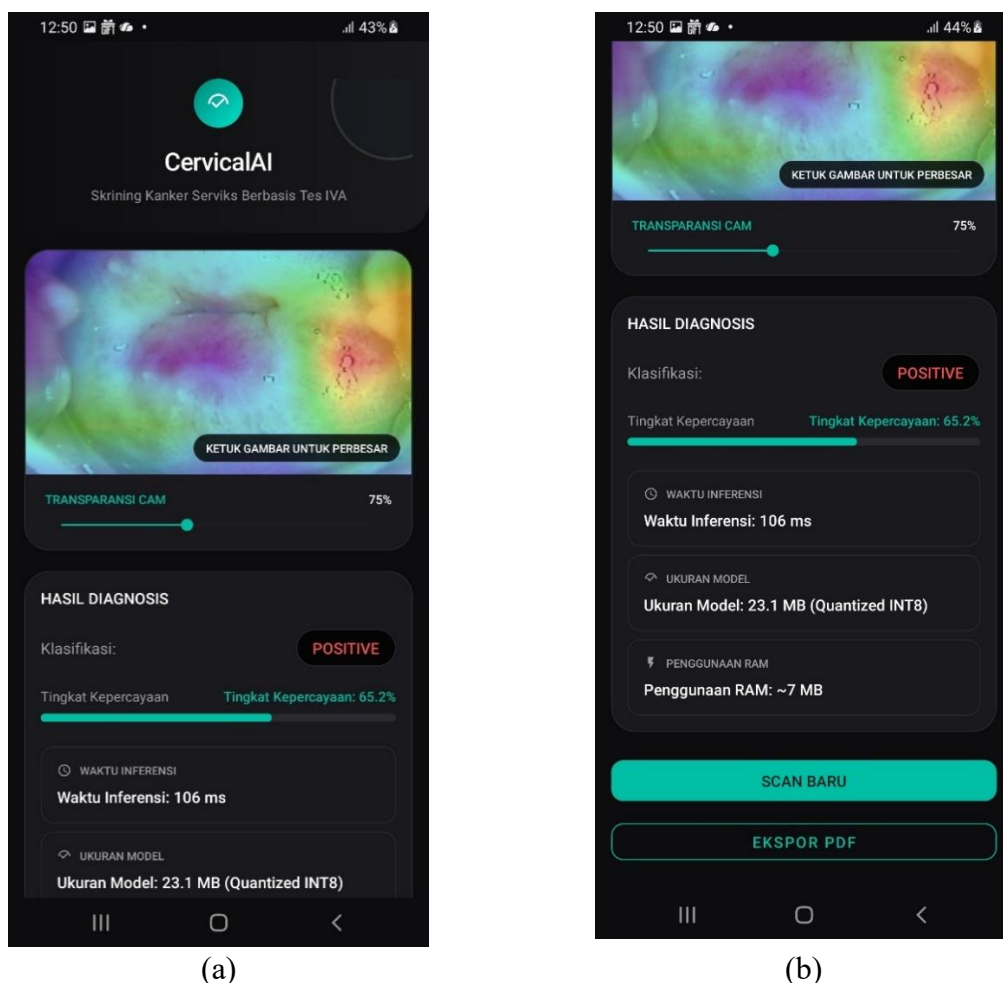
Gambar 3.9 Tampilan Halaman Kamera Kustom dengan *Overlay* Panduan SSK

Fitur-fitur pada halaman kamera meliputi:

1. Garis Panduan Kamera: Tampilan *overlay* di atas pratinjau kamera berupa lingkaran hijau putus-putus. Lingkaran ini memandu pengguna menempatkan posisi serviks tepat di tengah layar.
2. Pengaturan Perbesaran Lensa (*Zoom*): Pengguna dapat memperbesar gambar dengan mencubit layar atau menekan tombol pilihan perbesaran. Pustaka CameraX mengatur perpindahan lensa kamera belakang secara otomatis sesuai kemampuan ponsel.
3. Fokus Sentuh: Ketukan pada area pratinjau memicu lensa untuk memfokuskan gambar pada titik tersebut agar detail serviks terlihat tajam.
4. Pengaturan Lampu Kilat (*Flash*): Tombol untuk menyalakan lampu kilat agar memberikan pencahayaan stabil saat pengambilan gambar.
5. Tombol Ambil Gambar: Tombol yang menghasilkan foto beresolusi penuh melalui mekanisme FileProvider. Foto resolusi penuh diperlukan untuk analisis medis daripada sekadar gambar pratinjau kecil.

3.3.9.4 Halaman Hasil Analisis

Setelah gambar diambil atau dipilih, aplikasi menampilkan diagnosis hasil analisis model. Tampilan halaman hasil analisis ini dapat dilihat pada Gambar 4.5.



Gambar 3.10 Tampilan Halaman Hasil Analisis: (a) Pratinjau Citra dengan *Overlay Heatmap* CAM, (b) Kartu Hasil Diagnosis

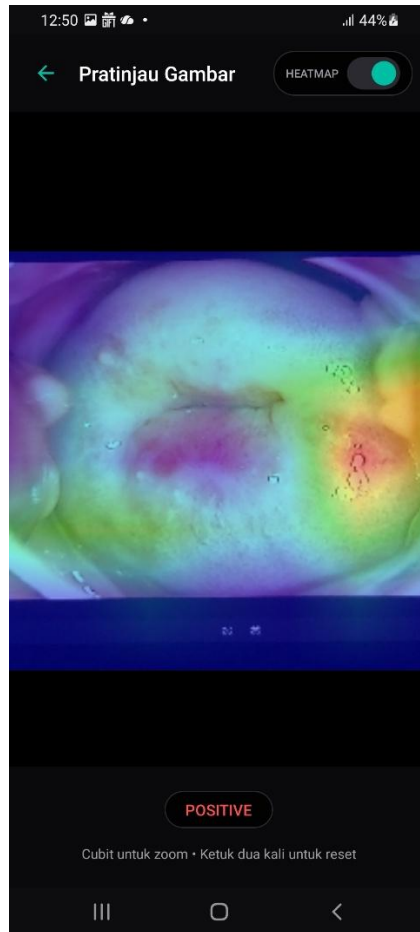
Komponen-komponen pada halaman hasil analisis meliputi:

1. Tampilan Gambar dengan *Heatmap*: Bagian ini memperlihatkan gambar serviks yang dilapisi *heatmap* CAM menggunakan warna biru hingga merah. Warna merah menunjukkan area yang paling memengaruhi hasil keputusan model.
2. Geser Transparansi *Heatmap*: Penggeser (*slider*) yang digunakan untuk mengatur kejelasan visualisasi *heatmap* dari 0% sampai 100%. Secara default, transparansi diatur pada 40%.
3. Hasil Diagnosis dan Metrik: Bagian ini menunjukkan hasil analisis berupa Label klasifikasi "POSITIVE" dengan warna merah atau "NORMAL" dengan warna hijau, persentase tingkat kepercayaan model terhadap hasil diagnosis, dan catatan performa komputasi seperti durasi inferensi, ukuran file model (.tflite), dan perkiraan pemakaian memori RAM.
4. Ekspor PDF dan Pemindaian Ulang: Tombol untuk memulai pemindaian baru atau membuat laporan berbentuk berkas PDF. Berkas PDF berisi data tanggal pemeriksaan, hasil diagnosis, tingkat kepercayaan, serta gambar serviks asli berdampingan dengan gambar *heatmap*.

Sebelum analisis berjalan, sistem meminta konfirmasi untuk menyimpan hasil gambar ke galeri ponsel dalam folder "CervicalAI". Sistem juga mengukur tingkat ketajaman gambar. Jika gambar terlalu buram, aplikasi memunculkan peringatan bagi pengguna untuk mengambil ulang foto.

3.3.9.5 Halaman Pratinjau Gambar

Halaman ini menampilkan gambar secara satu layar penuh untuk memeriksa detail hasil pemindaian. Pengguna dapat masuk ke halaman ini dengan mengetuk pratinjau gambar pada halaman hasil. Gambar 4.6 menunjukkan tampilan halaman pratinjau penuh.



Gambar 3.11 Tampilan Halaman Pratinjau Gambar Layar Penuh dengan *Toggle Heatmap*

Fitur-fitur inspeksi yang tersedia meliputi:

1. Pembesaran Gambar: Pengguna dapat memperbesar gambar hingga 8 kali dengan gerakan mencubit layar.
2. Geser Tampilan: Gambar yang telah diperbesar dapat digeser ke berbagai arah untuk melihat bagian yang berbeda.
3. Reset Tampilan: Ketukan ganda pada layar mengembalikan ukuran gambar seperti semula secara otomatis.
4. Sakelar *Heatmap*: Tombol pilihan untuk menyalakan atau mematikan lapisan *heatmap* CAM. Tombol ini membantu membandingkan temuan langsung pada foto asli dengan bagian yang disorot oleh model AI.

3.3.9.6 Fitur Teknis Pendukung

Selain empat halaman utama di atas, aplikasi CervicalAI memiliki beberapa fitur teknis untuk mendukung sistem:

1. Inferensi *Multi-Output* TFLite: Model TFLite dikonfigurasi dengan dua *output* tensor, yaitu tensor klasifikasi [1,1] untuk probabilitas dan tensor *feature maps* [1,7,7,2048] dari lapisan konvolusi terakhir (conv5_block3_out). Konfigurasi ini

membuat klasifikasi dan pembuatan CAM dapat berjalan dalam satu kali inferensi. Interpreter TFLite menggunakan 4 thread CPU dan menonaktifkan XNNPACK untuk menjaga kestabilan di berbagai perangkat.

2. Pembuatan CAM di Perangkat: *Heatmap* CAM dihitung secara lokal dengan rumus $CAM(h,w) = \text{ReLU}(\sum_c W_c \times F(h,w,c))$. Di sini, W adalah bobot lapisan *Dense* dari *file* JSON dan F adalah *feature maps* dari keluaran model. Hasil CAM berukuran 7×7 piksel kemudian diperbesar ke dimensi citra asli menggunakan interpolasi bilinear sebelum digabungkan dengan *jet colormap*.
3. Dukungan Layar Lipat dan *Multi-Window*: Seluruh *Activity* menggunakan pengaturan *resizeableActivity="true"* dan menangani perubahan konfigurasi layar (*screenSize*, *screenLayout*, *orientation*, *density*) tanpa mengulang *Activity*. Cara ini menjaga data inferensi dan gambar di memori saat layar dilipat atau dibagi. Tampilan antarmuka juga menyesuaikan ukuran layar melalui *resource qualifiers*.
4. Manajemen Memori: Aplikasi mengelola memori secara ketat untuk mencegah kesalahan kehabisan memori (*OutOfMemoryError*) pada ponsel berspesifikasi rendah. Metode yang digunakan adalah pemakaian ulang *ByteBuffer* untuk prapemrosesan, daur ulang gambar (*bitmap recycling*) secara manual, dan penghapusan referensi statis saat *Activity* ditutup.
5. Modul *Preprocessing* OpenCV (*CervixCropper*): Berkas *CervixCropper.kt* menyediakan fungsi untuk memotong area serviks secara otomatis. Tahapannya meliputi konversi ke skala keabu-abuan, filter Gaussian blur, pengambangan Otsu, deteksi kontur terbesar, dan penentuan lingkaran pembatas terkecil. Pemotongan otomatis ini diaktifkan pada alur utama aplikasi sebelum menjalankan inferensi model, sehingga *input* citra yang diproses oleh model TensorFlow Lite selaras dengan format data gambar yang digunakan selama pelatihan model ResNet-50.

Waktu inferensi lokal yang diukur pada CPU *smartphone* uji tercatat rata-rata sebesar 120 ms hingga 150 ms per citra. Kecepatan ini tergolong responsif dan nyaman digunakan untuk kebutuhan pelayanan skrining di lapangan.

[Halaman ini sengaja dikosongkan]

BAB 4

HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai hasil dari tahapan implementasi yang telah dirancang pada bab sebelumnya. Di akhir bab ini juga akan ditunjukkan pembahasan terkait dengan penelitian dan percobaan yang telah dilakukan oleh penulis.

4.1 Skenario Uji Coba

4.1.1 Uji Performa Model

Pengujian performa model bertujuan untuk mengukur akurasi klasifikasi dan efisiensi model sebelum diterapkan pada aplikasi seluler. Pengujian ini menggunakan spesifikasi perangkat keras dan perangkat lunak yang telah dijabarkan pada Sub-bab 3.2.

Skenario pengujian dibagi menjadi dua bagian utama:

1. Skenario Model *Transfer learning Deep learning*: Pengujian ini menyilangkan 3 arsitektur (MobileNetV3-Small, ResNet-50, dan EfficientNet-B0) dengan 9 strategi pra-pemrosesan citra dari *ablation study* (NB-1 sampai NB-9). Setiap kombinasi melintasi dua fase pelatihan (*feature extraction* dan *fine tuning*). Model dievaluasi menggunakan metrik sensitivitas, spesifisitas, PR-AUC, ROC-AUC, dan *Precision@recall* 80% pada dataset uji (*hold-out test set*).
2. Skenario Model *Baseline* Tradisional: Pengujian ini mereplikasi penelitian terdahulu menggunakan ekstraksi fitur warna dan tekstur manual (*handcrafted features*) dengan 9 algoritma klasifikasi *machine learning* konvensional (NB-10) sebagai pembanding.

Untuk menguji dampak optimasi, model Float32 terbaik dikonversi ke format INT8 melalui *Post-Training Quantization* (PTQ) menggunakan 200 sampel citra kalibrasi. Pengukuran latensi dan ukuran model .tflite dilakukan secara lokal pada *smartphone* untuk memastikan keakuratan data waktu inferensi.

4.1.2 Uji Pengguna

Pengujian pengguna menggunakan metode *User Acceptance Testing* (UAT) untuk mengetahui respon pengguna terhadap fungsi aplikasi CervicalAI serta mengukur akurasi diagnostik sistem di lapangan. Pengujian ini melibatkan 4 dokter spesialis dan residen Obstetri dan Ginekologi di Rumah Sakit Umum Pusat (RSUP) Hasan Sadikin Bandung.

Prosedur pengujian dibagi menjadi dua tahap utama:

1. Uji Validasi Klinis (Akurasi Diagnostik): Dokter melakukan pemeriksaan IVA konvensional pada pasien secara langsung, kemudian menggunakan aplikasi CervicalAI untuk memindai citra serviks pasien yang sama. Hasil klasifikasi dari aplikasi dicocokkan (*cross-check*) dengan diagnosis klinis dokter untuk mendapatkan data *True positive* (TP), *True negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Langkah ini bertujuan mengukur kinerja diagnostik aplikasi pada kondisi klinis riil.
2. Uji Fungsionalitas dan Antarmuka: Setiap dokter mencoba simulasi fitur interaksi mandiri pada aplikasi menggunakan *smartphone* uji dengan mengikuti alur tugas berikut:

- a. Mengambil citra serviks dengan memposisikan kamera menggunakan bantuan lingkaran panduan SSK.
- b. Membaca hasil klasifikasi biner dan visualisasi Grad-CAM pada layar.
- c. Mencoba fitur interaksi visual seperti perbesaran gambar (*pinch to zoom*), penggeseran gambar (*pan*), dan tombol pengalih (*toggle heatmap*).
- d. Menyimpan hasil pemeriksaan ke dalam galeri ponsel.

Setelah simulasi selesai, responden menilai fungsi sistem dengan mengisi kuesioner skala Likert (skala 1 sampai 5). Evaluasi UAT ini mencakup kriteria kemudahan penggunaan umum (C1), fitur kamera dan panduan SSK (C2), tampilan hasil deteksi (C3), fitur inspeksi gambar (C4), fitur penyimpanan gambar (C5), serta kecepatan dan stabilitas aplikasi (C6). Kuesioner juga mengukur dampak aplikasi terhadap alur kerja pemeriksaan IVA konvensional (dimensi D) dan tingkat kepercayaan klinis dokter terhadap keputusan model (dimensi E). Seluruh data skor, matriks kebingungan (*confusion matrix*) hasil validasi klinis, dan analisis kualitatif dijabarkan pada sub-bab berikutnya.

4.2 Hasil Uji Coba

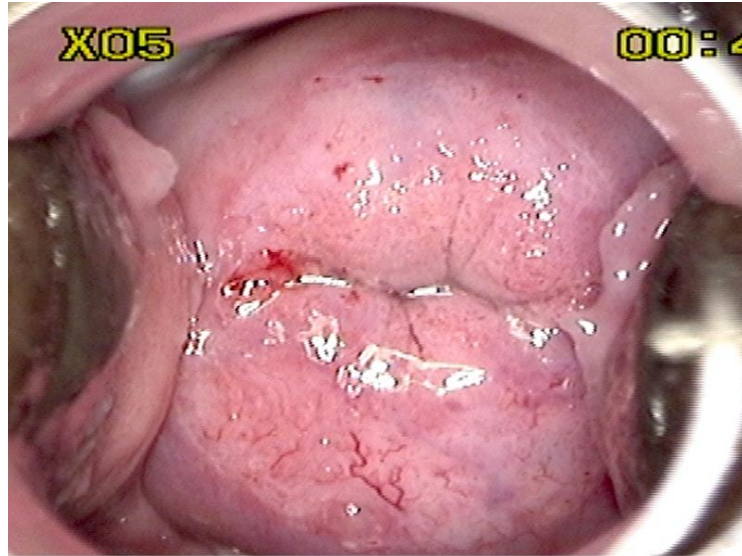
4.2.1 Evaluasi Strategi Pra-pemrosesan Citra

Pra-pemrosesan menentukan apakah model kecerdasan buatan benar-benar mempelajari biologi serviks atau hanya sekadar menghafal pola latar belakang. Penelitian ini menguji sembilan skenario pra-pemrosesan yang berbeda.

Strategi pra-pemrosesan menentukan apakah model klasifikasi benar-benar mempelajari pola biologis jaringan serviks. Bagian ini menguraikan hasil dari setiap strategi pra-pemrosesan yang diuji, disertai analisis visual terhadap citra keluaran dan kurva evaluasi model pada dataset uji.

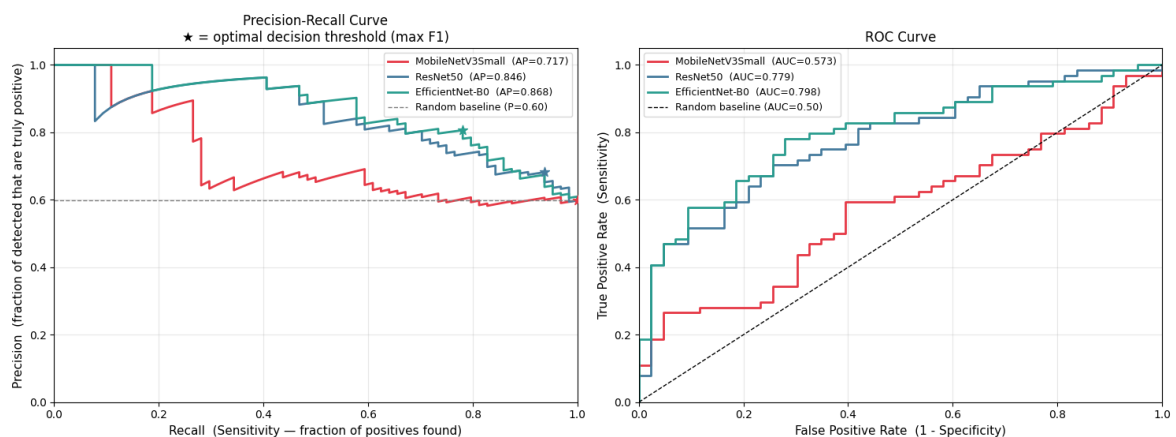
4.2.1.1 Citra Mentah Tanpa Pra-pemrosesan (NB-1)

Pada skenario NB-1, citra asli hasil kamera dimasukkan langsung ke dalam model tanpa pemotongan area dan tanpa augmentasi data. Citra 224x224 piksel yang dihasilkan masih memuat seluruh elemen non-biologis dengan contoh ujung spekulum logam di tepi kiri dan kanan, bayangan gelap di sudut gambar, pantulan lampu kilat pada permukaan logam, serta teks stempel waktu klinis yang tertanam di sudut gambar. Serviks hanya menempati sekitar 40-60% dari total area piksel, sementara sisanya diisi oleh elemen gangguan yang tidak memiliki hubungan dengan diagnosis kanker.



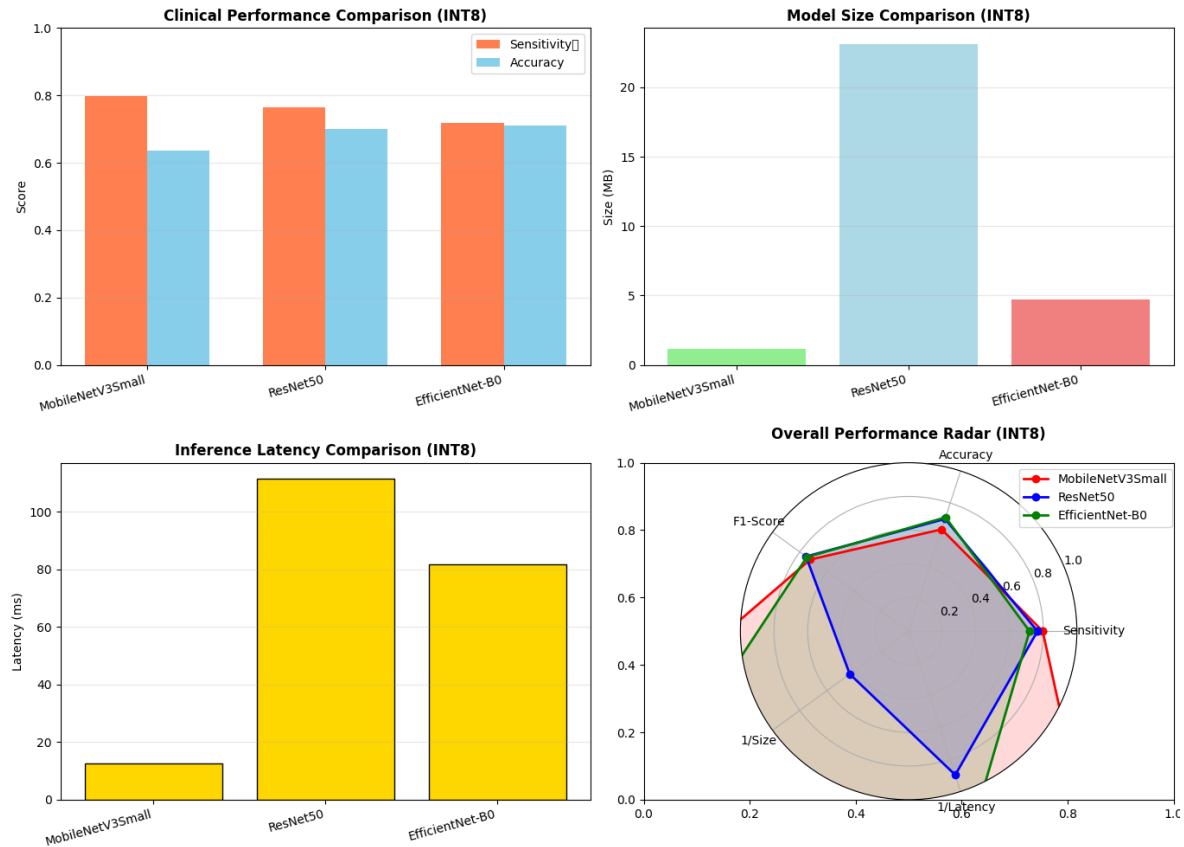
Gambar 4.1 Contoh citra mentah yang dijadikan sebagai citra pembanding tanpa pra-pemrosesan (NB-1): serviks dikelilingi oleh spekulum logam, bayangan, dan teks klinis

Secara kontrainuitif, model yang dilatih pada citra mentah ini mencetak metrik evaluasi tertinggi di antara seluruh percobaan. EfficientNet-B0 mencatat AUC 0.798, sedangkan ResNet-50 mencapai AUC 0.779 seperti pada Gambar 4.2. Angka-angka ini melebihi hasil dari skenario yang menggunakan pemotongan otomatis.



Gambar 4.2 Kurva *Precision-Recall* dan ROC model NB-1 pada dataset uji

Namun, angka tinggi ini merupakan ilusi yang berbahaya secara klinis. Fenomena ini dikenal sebagai efek *Clever Hans* atau *shortcut learning* dalam literatur kecerdasan buatan medis. Tanpa augmentasi yang mengacak posisi dan orientasi piksel, artefak latar belakang bersifat statis dan identik antar-gambar dari kamera yang sama. Model menghafal korelasi palsu antara artefak ini dengan label diagnosis. Sebagai contoh, merek spekulum tertentu mungkin hanya digunakan di klinik yang menangani kasus positif, atau posisi pantulan cahaya yang konsisten secara tidak sengaja berkorelasi dengan label kelas. Jika model ini diterapkan di klinik lain dengan merek spekulum atau pencahayaan yang berbeda, akurasi akan anjlok drastis. Skenario NB-7 membuktikan bahwa pra-pemrosesan bukan langkah opsional, melainkan keharusan untuk memastikan validitas biologis model yang hasilnya dapat dilihat pada Gambar 4.3.

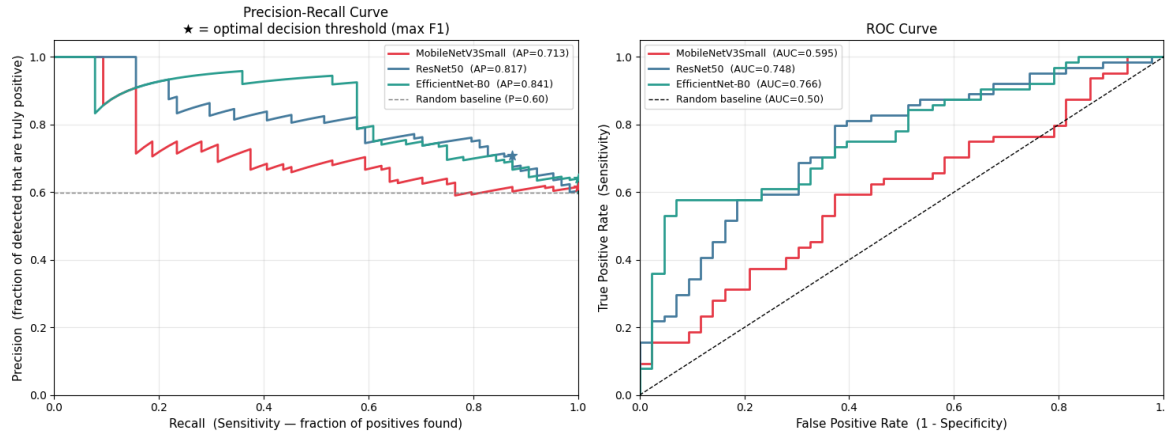


Gambar 4.3 *Dashboard* ringkasan performa model NB-7: metrik klinis dan komputasi

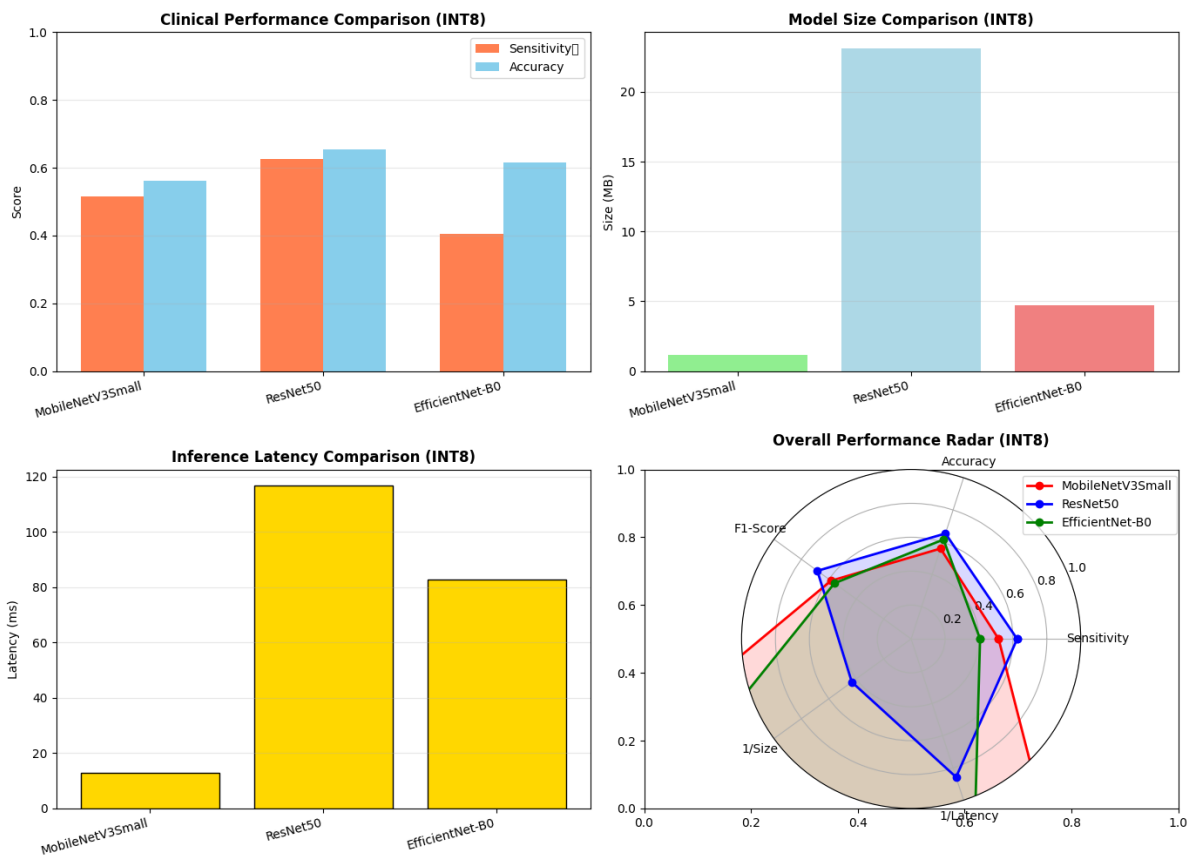
4.2.1.2 Baseline Augmentasi Tanpa Pemotongan (NB-2)

Skenario NB-2 menggunakan citra mentah yang sama dengan NB-1, namun menambahkan augmentasi data standar selama pelatihan, yaitu rotasi acak, pergeseran horizontal dan vertikal, pencerminan, serta variasi kecerahan. Citra keluaran pra-pemrosesan NB-2 secara visual identik dengan NB-1, yaitu citra penuh tanpa pemotongan yang masih memuat spekulum dan latar belakang. Perbedaan terletak pada fase pelatihan, di mana augmentasi mengacak posisi piksel sehingga artefak latar belakang tidak lagi berada di lokasi yang persis sama pada setiap iterasi.

Hasilnya, metrik evaluasi sedikit menurun dibandingkan citra mentah, yaitu AUC turun dari 0.798 (NB-1) menjadi 0.766 (NB-2) seperti pada Gambar 4.4. Penurunan ini justru merupakan indikator positif. Augmentasi berhasil merusak pola latar belakang statis yang sebelumnya dieksploitasi model untuk curang. Model mulai terdorong mencari fitur medis pada jaringan serviks itu sendiri, meskipun belum sepenuhnya terbebas dari pengaruh artefak karena spekulum dan teks masih hadir dalam gambar. Untuk visualisasi hasilnya bisa dilihat pada Gambar 4.5.



Gambar 4.4 Kurva *Precision-Recall* dan ROC model NB-2 pada dataset uji

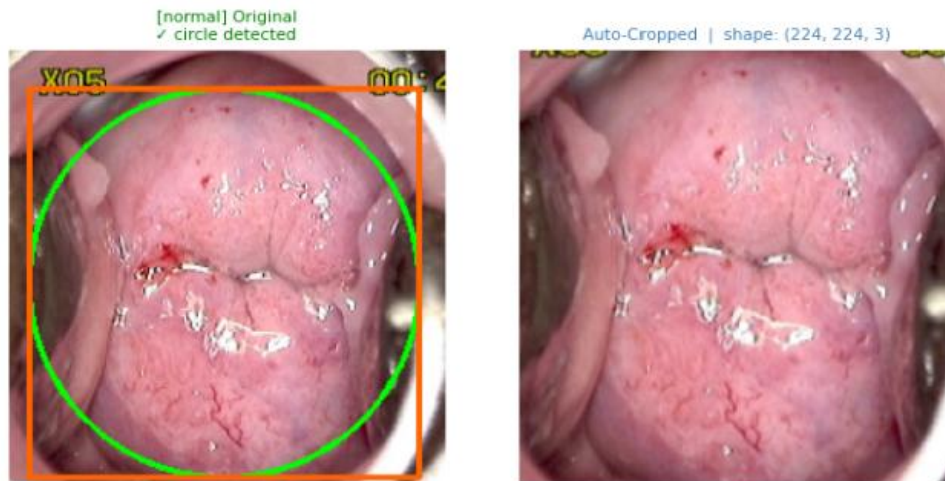


Gambar 4.5 *Dashboard* ringkasan performa model NB-2: metrik klinis dan komputasi

4.2.1.3 Pemotongan Otomatis Berbasis Kontur OpenCV (NB-4 dan NB-5)

Untuk memaksa model sepenuhnya fokus pada jaringan serviks, skenario ini menerapkan pemotongan otomatis (*auto-crop*) menggunakan deteksi kontur OpenCV. Algoritma pertama-tama mengonversi citra ke skala keabu-abuan, menerapkan Gaussian blur, dan melakukan pengambangan Otsu untuk menemukan kontur terbesar pada gambar. Lingkaran pembatas terkecil (*minimum enclosing circle*) kemudian ditentukan, dan area persegi di sekitar lingkaran tersebut dipotong lalu diubah ukurannya menjadi 224x224 piksel.

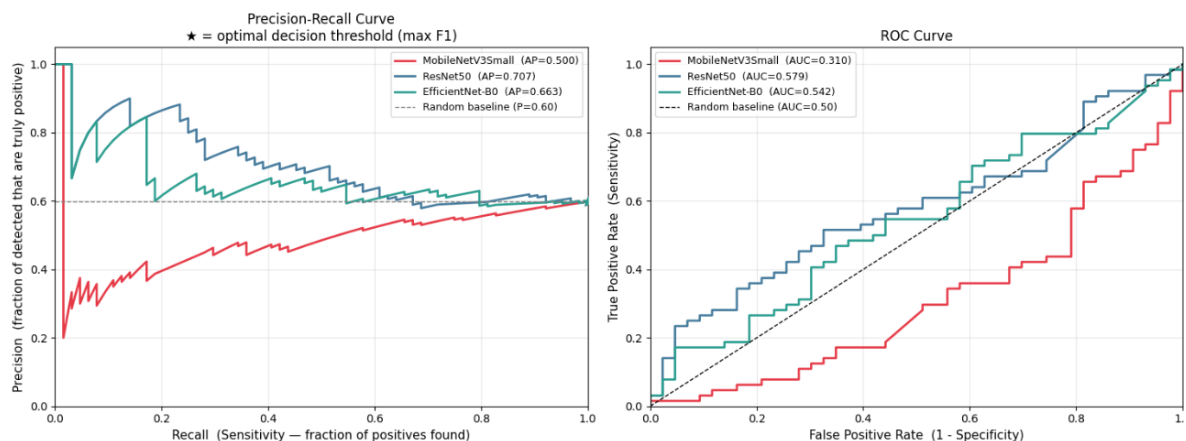
Hasil pemotongan memperlihatkan serviks yang telah terpusat di tengah gambar dengan efek perbesaran otomatis. Detail visual pada area Sambungan *Skuamosa Kolumnar* (SSK) dan permukaan jaringan menjadi jauh lebih jelas dibandingkan citra mentah. Spekulum logam dan teks klinis berhasil terbuang sepenuhnya. Seluruh 224x224 piksel kini terisi oleh data biologis, bukan artefak kamera seperti pada Gambar 4.6.



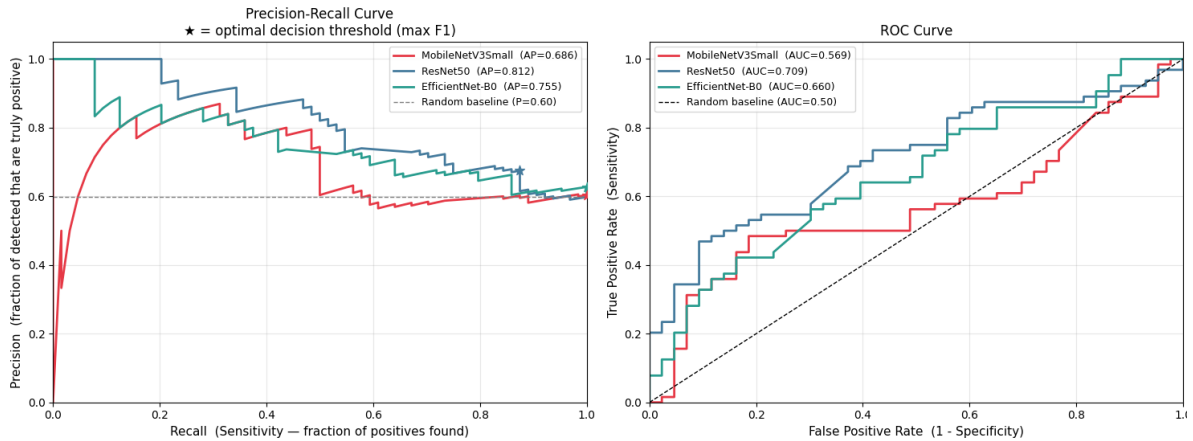
Gambar 4.6 Perbandingan citra sebelum dan sesudah pemotongan otomatis OpenCV: (Kiri) citra asli dengan anotasi kontur dan kotak pembatas, (Kanan) hasil *auto-crop* 224x224 pada citra perbandingan

Namun, *auto-crop* OpenCV memiliki keterbatasan. Pada gambar dengan pencahayaan redup atau bayangan yang kuat, algoritma pengembangan Otsu kadang gagal menemukan kontur yang tepat. Lingkaran deteksi bisa meleset, entah terlalu kecil sehingga memotong sebagian jaringan serviks, atau terlalu besar sehingga masih menyertakan tepi spekulum.

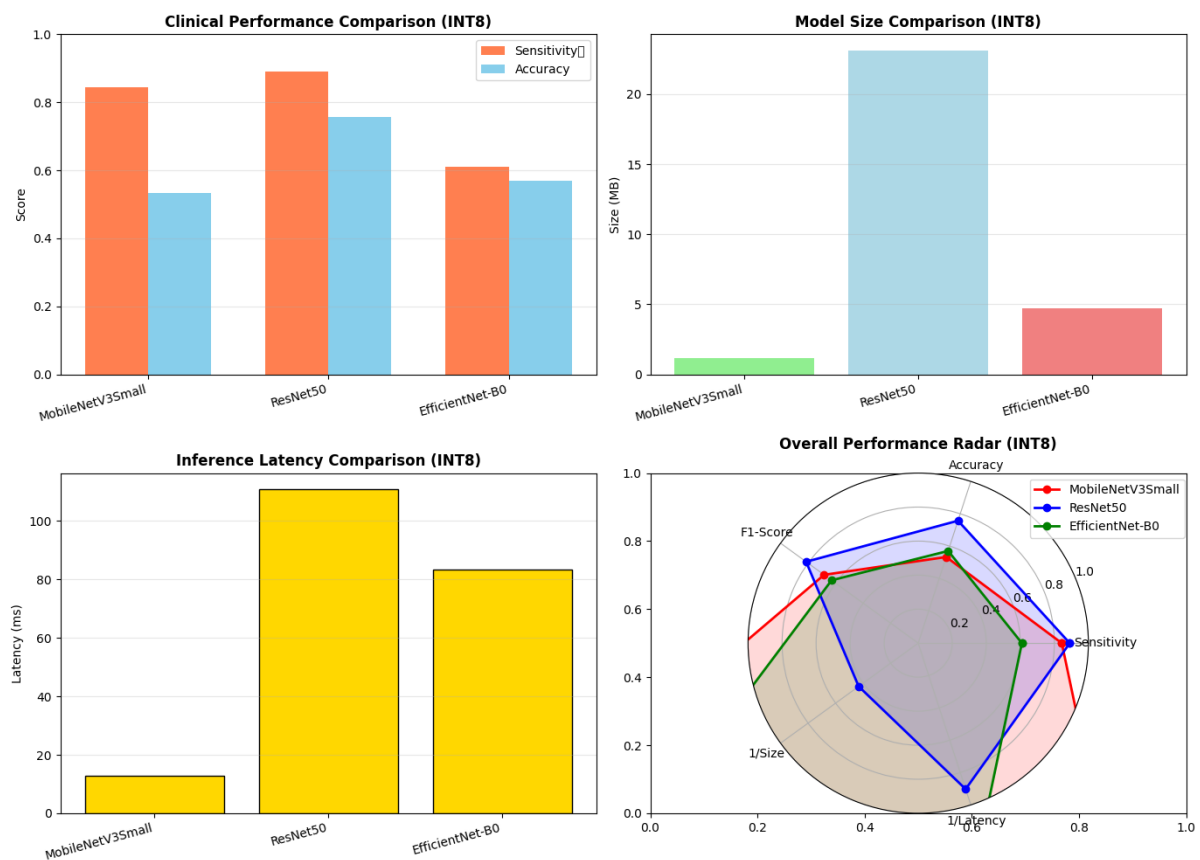
Dampak *auto-crop* terhadap performa model sangat dramatis. Karena model tidak lagi bisa mengandalkan artefak latar belakang, tugas klasifikasi menjadi jauh lebih sulit. AUC ResNet-50 turun ke 0.709 seperti pada Gambar 4.8, angka yang merepresentasikan kemampuan biologis yang sebenarnya dari model. Skenario NB-5 menggunakan pra-pemrosesan *auto-crop* yang identik dengan NB-4, namun mengoptimalkan fungsi pelatihan terhadap metrik *Precision@recall*, yang akan dibahas lebih lanjut pada sub-bab evaluasi model.



Gambar 4.7 Kurva *Precision-Recall* dan ROC model NB-4 pada dataset uji



Gambar 4.8 Kurva *Precision-Recall* dan ROC model NB-5 pada dataset uji



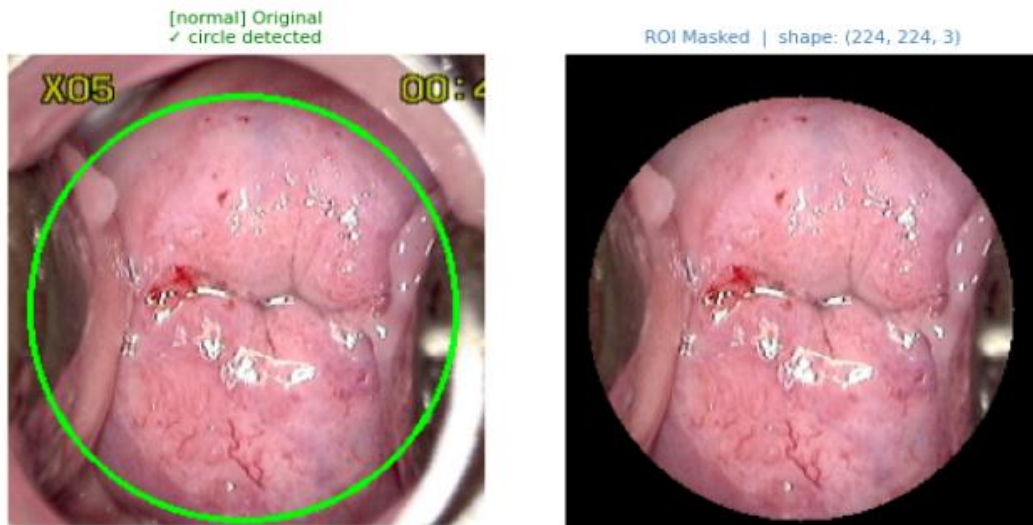
Gambar 4.9 *Dashboard* ringkasan performa model NB-5: metrik klinis dan komputasi

4.2.1.4 Masking ROI Sirkular Dinamis (NB-7)

Skenario NB-7 menempuh pendekatan yang berbeda dari pemotongan fisik. Alih-alih memotong dan memperbesar area serviks, metode ini mendeteksi kontur serviks dengan algoritma OpenCV yang sama, lalu mengubah seluruh piksel di luar lingkaran deteksi menjadi hitam pekat (nilai 0). Gambar keluaran memperlihatkan area serviks bulat di tengah bingkai 224x224 piksel, dikelilingi oleh blok hitam pekat di seluruh tepinya.

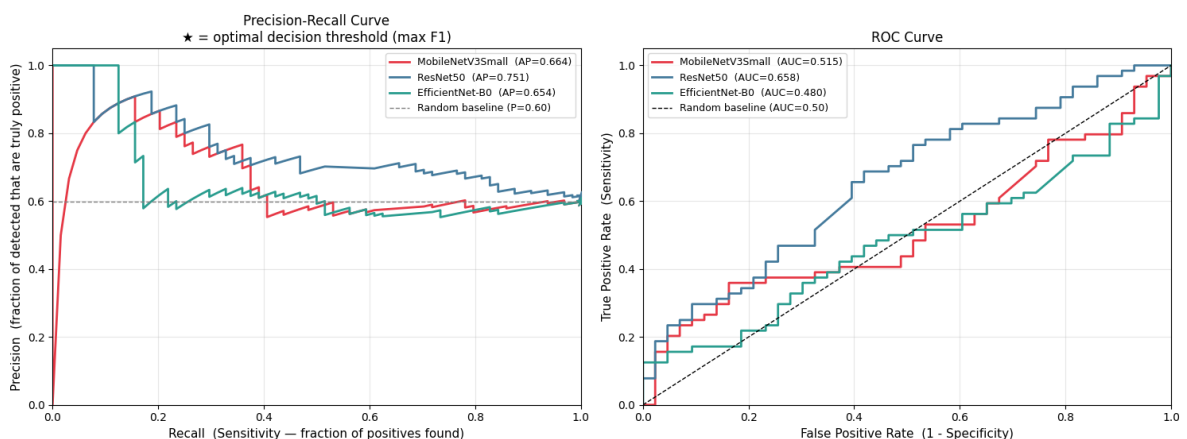
Metode ini berhasil menghilangkan artefak spekulum tanpa mengubah proporsi asli gambar. Namun, karena tidak ada efek perbesaran, serviks menempati area yang relatif kecil

di dalam bingkai. Sebagian besar tensor *input* (diperkirakan 40-50%) kini berisi piksel bernilai nol yang tidak membawa informasi diagnostik sama sekali seperti pada Gambar 4.10.

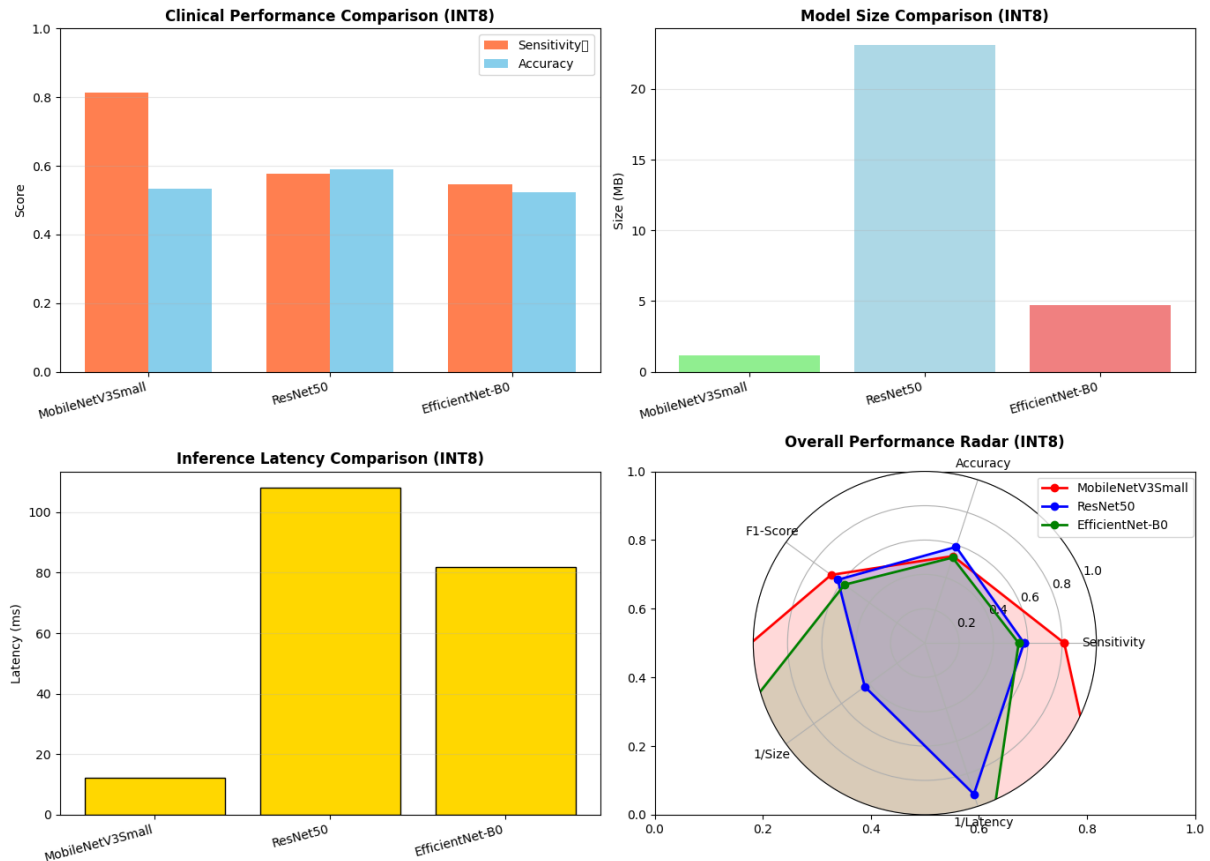


Gambar 4.10 Contoh hasil penerapan masking ROI sirkular dinamis (NB-7): serviks bulat di tengah dengan latar belakang hitam pada citra pembanding

Hasil evaluasi menunjukkan penurunan performa dibandingkan *auto-crop*. *Average Precision* ResNet-50 turun dari 0.812 (*auto-crop* NB-5) menjadi 0.751 (masking NB-7), dan AUC turun dari 0.709 menjadi 0.658. Penyebabnya ada tiga, yaitu (1) resolusi efektif jaringan serviks lebih rendah karena tidak ada efek *zoom*, sehingga detail tekstur lesi berkurang, (2) lapisan konvolusi awal membuang siklus komputasi untuk memindai area hitam kosong tanpa menghasilkan sinyal gradien, dan (3) transisi tajam antara piksel jaringan berwarna dan blok hitam menciptakan tepi artifisial berfrekuensi tinggi yang mengaktifkan filter konvolusi secara berlebihan, mengalihkan perhatian model dari tekstur lesi yang lebih halus.



Gambar 4.11 Kurva *Precision-Recall* dan ROC model NB-7 pada dataset uji

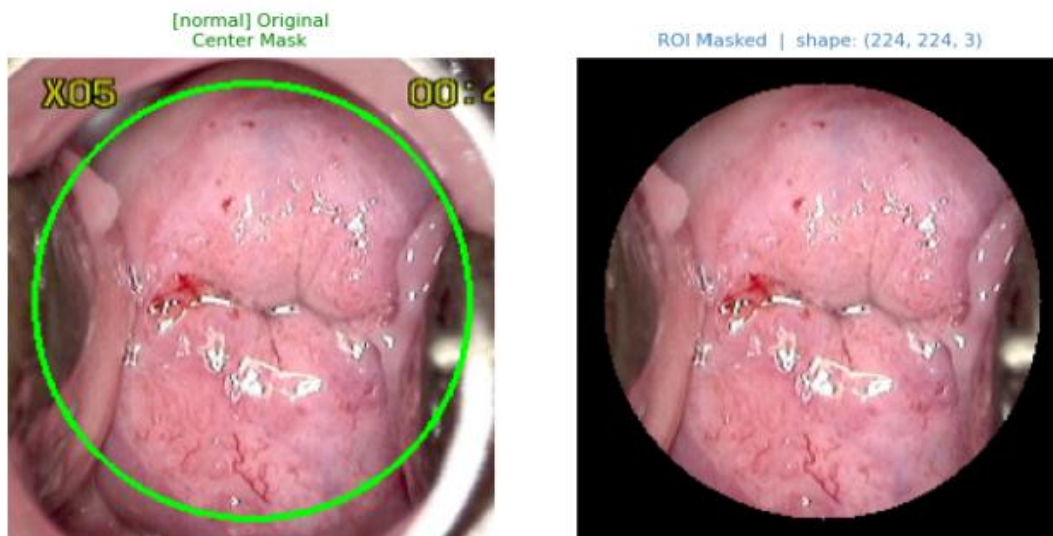


Gambar 4.12 *Dashboard* ringkasan performa model NB-7: metrik klinis dan komputasi

4.2.1.5 Masking ROI Sirkular Statis (NB-8)

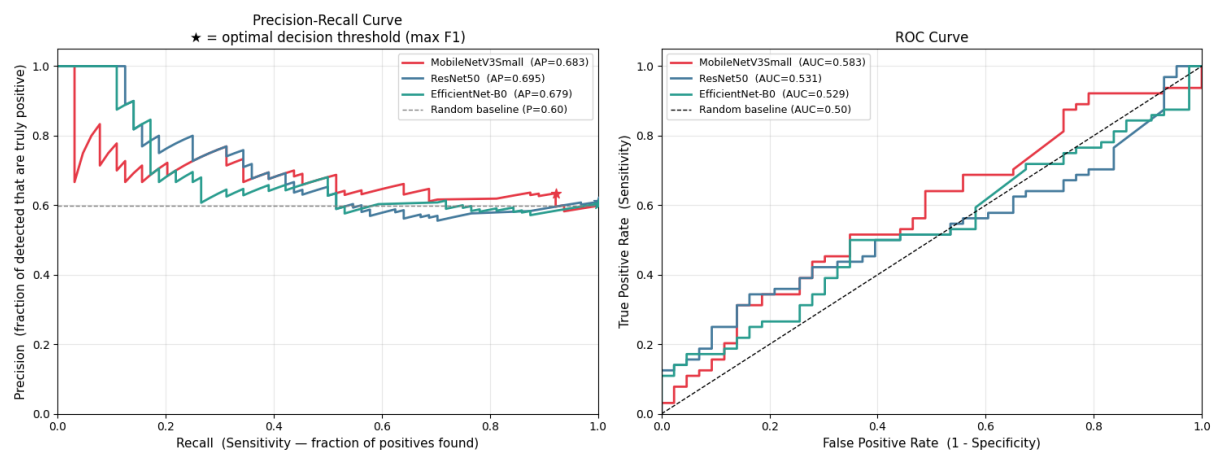
Skenario NB-8 menyederhanakan pendekatan NB-7 dengan menghilangkan deteksi kontur sepenuhnya. Masker lingkaran statis diterapkan tepat di koordinat tengah citra ($w/2$, $h/2$) dengan rasio radius tetap 0.45 terhadap dimensi gambar. Gambar keluaran sekilas serupa dengan NB-7 (lingkaran jaringan di tengah, hitam di luar), tetapi memiliki kelemahan fatal.

Pada praktik klinis, posisi serviks jarang tepat di tengah bingkai kamera. Sudut pengambilan gambar, jarak kamera, dan posisi pasien menyebabkan variasi posisi serviks yang signifikan. Akibatnya, masker statis ini kerap memotong bagian penting dari jaringan serviks yang berada di luar lingkaran (termasuk area lesi *acetowhite*), sekaligus menyertakan bagian spekulum yang masuk ke dalam area lingkaran.

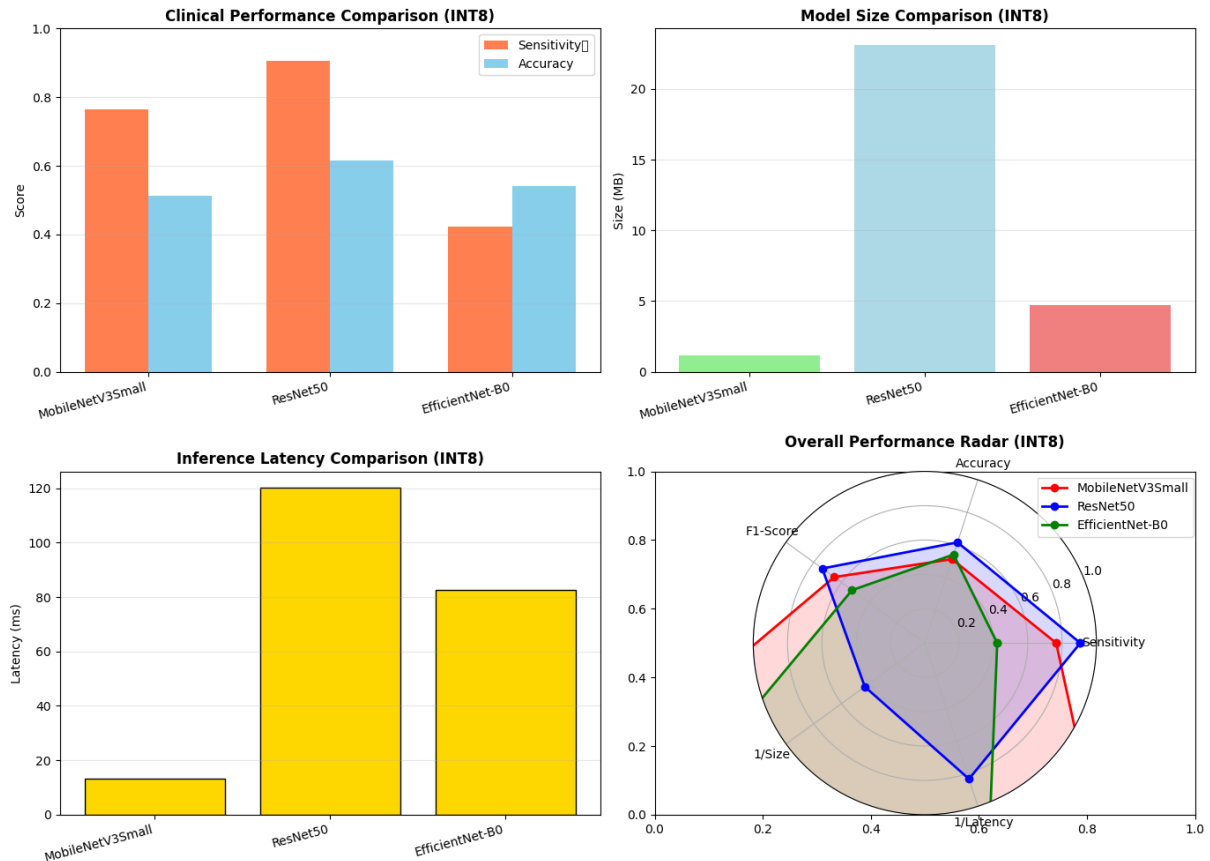


Gambar 4.13 Contoh hasil citra pembeding untuk masking ROI sirkular statis (NB-8)

Performa model pada skenario ini runtuh total. Kurva ROC ketiga arsitektur model nyaris menempel pada garis tebakan acak (diagonal). AUC ResNet-50 jatuh ke 0.531, yang berarti model hanya sedikit lebih baik dari melempar koin. MobileNetV3-Small secara paradoks justru sedikit lebih baik (AUC 0.583) karena arsitektur yang sederhana tidak terlalu mudah *overfit* terhadap pola tepi masker hitam yang artifisial.



Gambar 4.14 Kurva *Precision-Recall* dan ROC model NB-8 pada dataset uji (kurva ROC nyaris menyentuh garis acak)



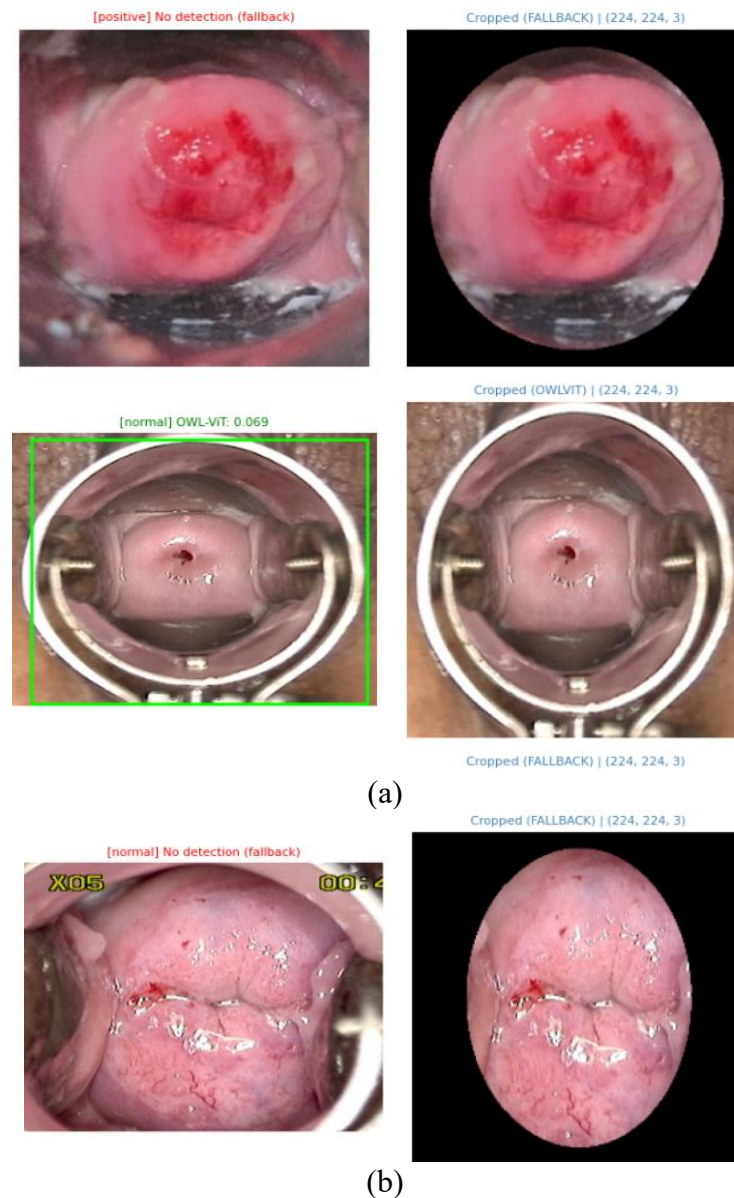
Gambar 4.15 Dashboard ringkasan performa model NB-8: metrik klinis dan komputasi

Perbandingan antara NB-7 (masking dinamis, AUC 0.658) dan NB-8 (masking statis, AUC 0.531) membuktikan bahwa deteksi kontur otomatis bukan hanya fitur tambahan, melainkan komponen kritis. Sementara itu, perbandingan antara masking dinamis (NB-7) dan pemotongan fisik (NB-5, AUC 0.709) mengonfirmasi bahwa pemotongan fisik yang disertai efek *zoom* jauh lebih superior daripada masking hitam untuk arsitektur CNN.

4.2.1.6 Pemotongan Citra Serviks Berbasis OWL-ViT (NB-9)

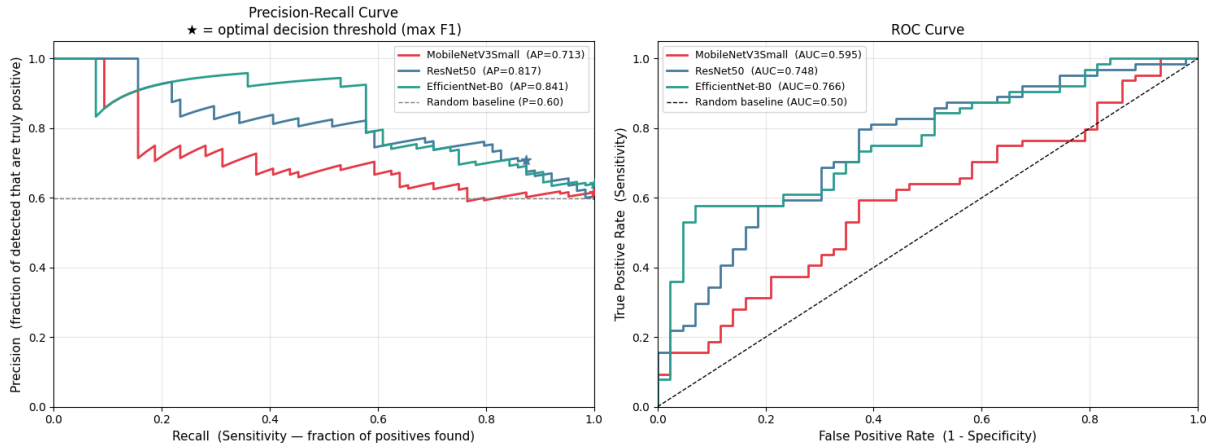
Skenario NB-9 menggantikan heuristik OpenCV dengan model fondasi (foundation model) OWL-ViT dari Google (Matthias Minderer Alexey Gritsenko, 2022). OWL-ViT adalah pendeteksi objek *zero-shot* yang menggunakan arsitektur *Vision Transformer* untuk melokalisasi objek berdasarkan kueri teks. Pada penelitian ini, kueri yang digunakan adalah "*cervix*" dan "*cervical opening*".

Gambar keluaran menunjukkan kotak pembatas (*bounding box*) persegi panjang berwarna hijau yang melingkupi area serviks. Berbeda dengan lingkaran OpenCV, kotak pembatas OWL-ViT mampu menyesuaikan bentuk dan posisinya secara fleksibel terhadap variasi sudut kamera. Pada citra yang sudah diperbesar (*zoomed-in*), di mana kontur sirkular OpenCV sering gagal karena tepi serviks tidak terlihat lengkap, OWL-ViT tetap berhasil mengenali area biologis berdasarkan pemahaman semantik visual-bahasa. Untuk citra di mana OWL-ViT gagal mendeteksi target (skor kepercayaan di bawah ambang batas), sistem cadangan berupa masker eliptikal diaktifkan agar pipeline tidak terhenti.

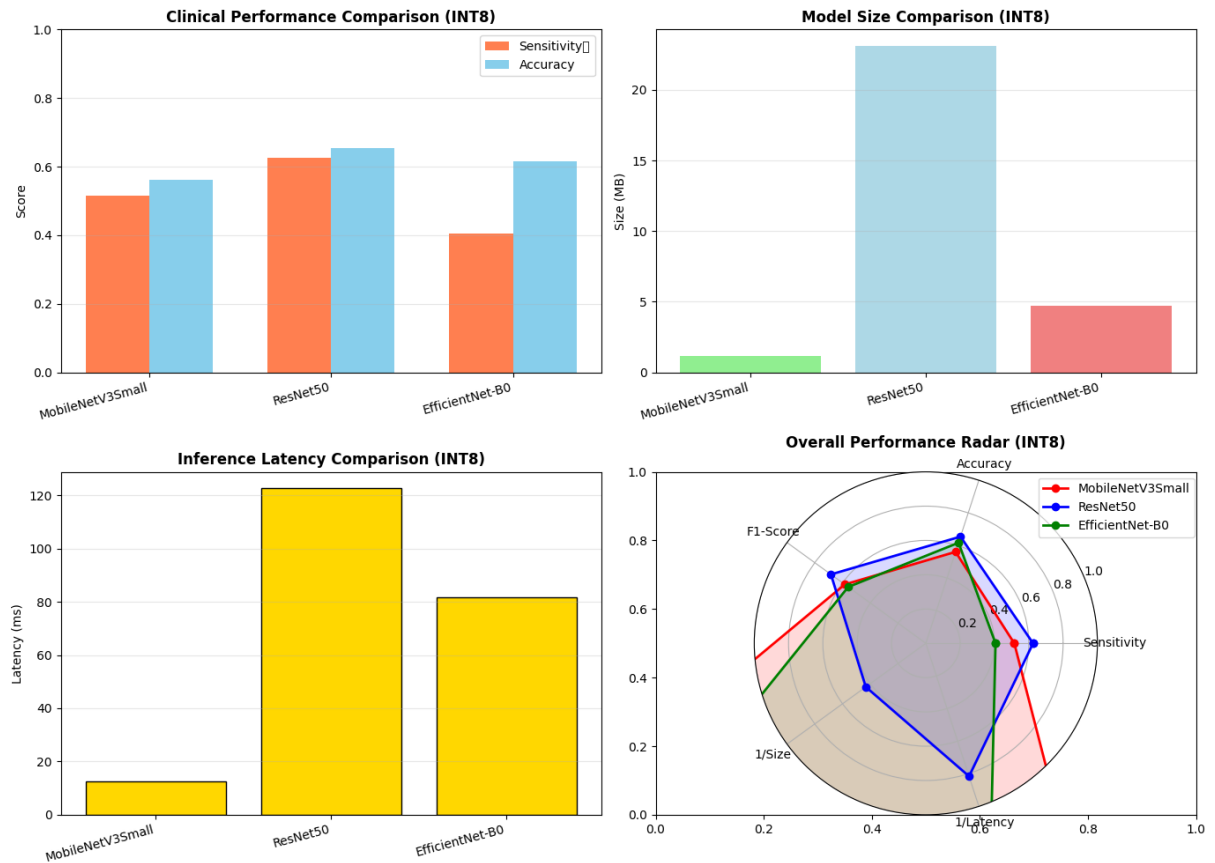


Gambar 4.16 (a) Contoh deteksi OWL-ViT pada sampel citra serviks: kotak pembatas, skor kepercayaan, dan hasil *crop* (b) Contoh penerapan *preprocessing* pada citra pembandingan

Hasil pemotongan OWL-ViT mengisi seluruh tensor 224x224 piksel dengan data jaringan, tanpa padding hitam. Kualitas potongan ini memberikan dampak terukur pada performa klasifikasi. AUC ResNet-50 naik dari 0.709 (*auto-crop* OpenCV) menjadi 0.748 (OWL-ViT). Yang lebih menarik, kualitas potongan OWL-ViT mengubah hierarki arsitektur: EfficientNet-B0 yang selama ini kalah dari ResNet-50 berhasil melampaui dengan AUC 0.766 dan *Average Precision* 0.841. EfficientNet adalah arsitektur yang sangat sensitif terhadap kualitas *input* karena dirancang untuk mengoptimalkan efisiensi pada setiap resolusi. Ketika diberi potongan jaringan yang bersih dan proporsional dari OWL-ViT, potensinya terbuka sepenuhnya.



Gambar 4.17 Kurva *Precision-Recall* dan ROC model NB-8 pada dataset uji



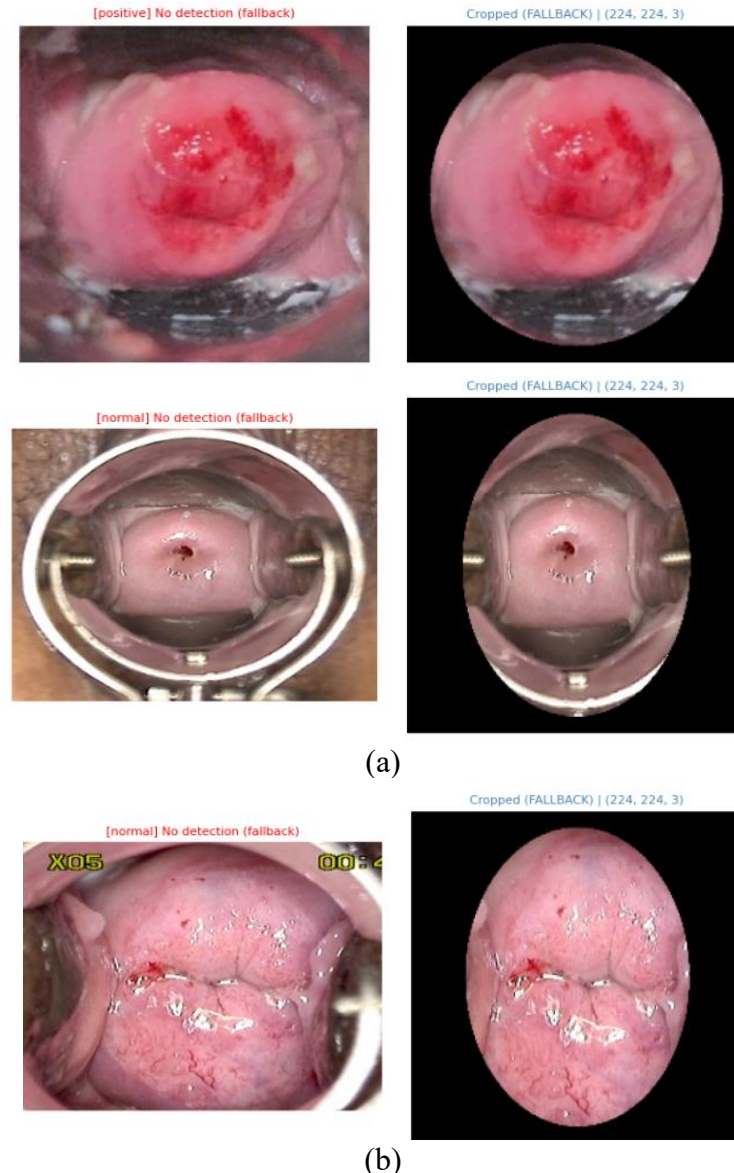
Gambar 4.18 *Dashboard* ringkasan performa model NB-8: metrik klinis dan komputasi

4.2.1.7 Pemotongan Citra Serviks Berbasis YOLOv11 SEVIA (NB-9)

Skenario terakhir menggunakan YOLOv11 yang telah dilatih khusus (*fine-tuned*) pada dataset citra serviks oleh proyek SEVIA. Berbeda dengan OWL-ViT yang bersifat *zero-shot* dan generalis, YOLOv11 telah mempelajari secara spesifik bentuk dan tekstur zona transformasi serviks.

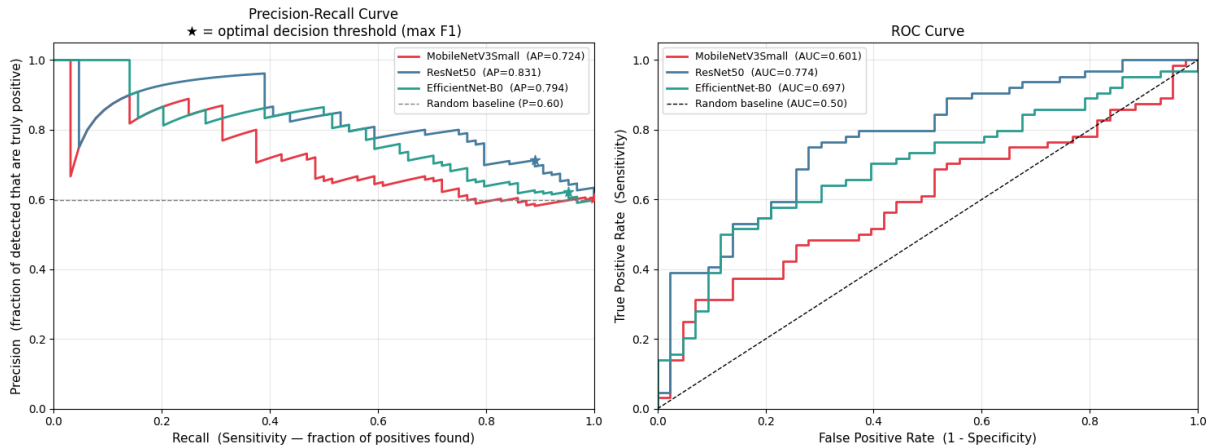
Gambar keluaran memperlihatkan kotak pembatas yang sangat rapat di sekitar area lesi atau zona transformasi, jauh lebih ketat dibandingkan OWL-ViT yang cenderung memotong

area organ secara keseluruhan. Perbedaan ini memiliki konsekuensi penting, yaitu potongan YOLOv11 menghilangkan konteks jaringan sehat di sekitar lesi, memaksa model klasifikasi bertindak hampir seperti pengklasifikasi histopatologi yang menganalisis tekstur jaringan secara sangat lokal.

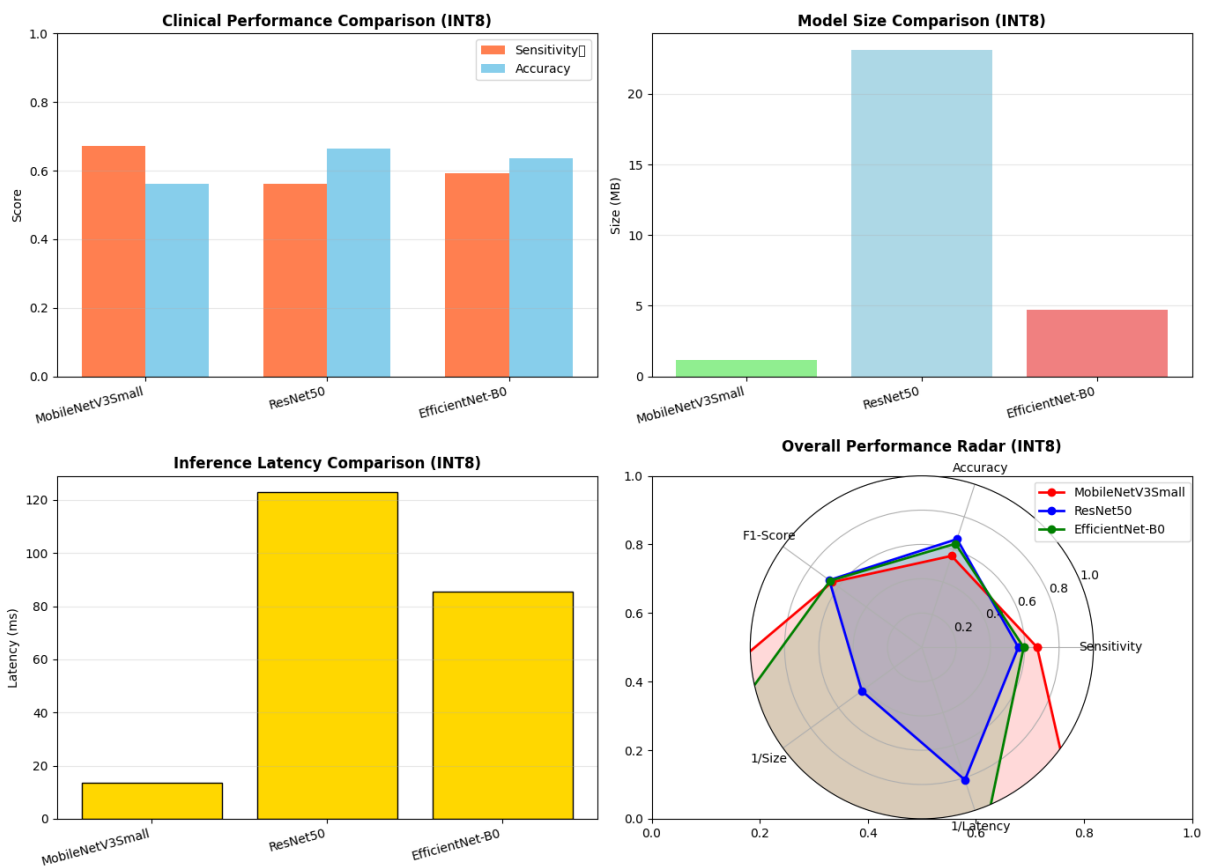


Gambar 4.19 (a) Contoh deteksi YOLOv11 SEVIA pada sampel citra serviks: kotak pembatas, skor kepercayaan, dan hasil *crop* (b) Contoh penerapan *preprocessing* pada citra pembandingan

Potongan yang sangat terfokus ini membalikkan kembali hierarki arsitektur. ResNet-50 merebut kembali posisi teratas dengan AUC 0.774 dan AP 0.831, pencapaian tertinggi yang dicatat ResNet-50 di seluruh eksperimen. Sebaliknya, EfficientNet-B0 mengalami penurunan (AUC dari 0.766 ke 0.697) karena arsitektur ini dioptimalkan untuk menangkap konteks global, dan potongan YOLOv11 yang terlalu ketat justru menghilangkan konteks tersebut. Temuan ini mengungkap sinergi yang menarik antara strategi pra-pemrosesan dan arsitektur model adalah potongan lebar cocok untuk EfficientNet, sementara potongan ketat cocok untuk ResNet yang unggul dalam ekstraksi fitur tekstur mendalam.



Gambar 4.20 Kurva *Precision-Recall* dan ROC model NB-9 pada dataset uji



Gambar 4.21 *Dashboard* ringkasan performa model NB-9: metrik klinis dan komputasi

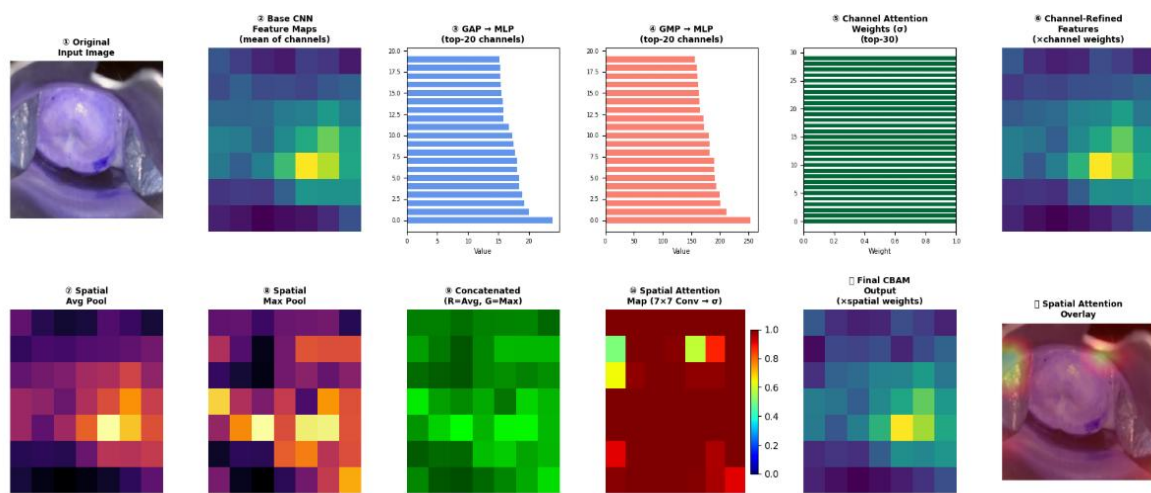
4.2.1.8 Pemotongan Citra Serviks dengan Mekanisme Atensi (CBAM)

Pelatihan model menggunakan skema *transfer learning* dua fase (ekstraksi fitur dan *fine-tuning*). ResNet-50 umumnya memberikan stabilitas pelatihan terbaik, sementara MobileNetV3-Small unggul dalam kecepatan konvergensi berkat parameternya yang sedikit.

Skenario NB-3 mencoba memasukkan mekanisme atensi *Convolutional Block Attention Module* (CBAM) ke dalam *base model*. Hipotesis awalnya adalah CBAM akan membantu

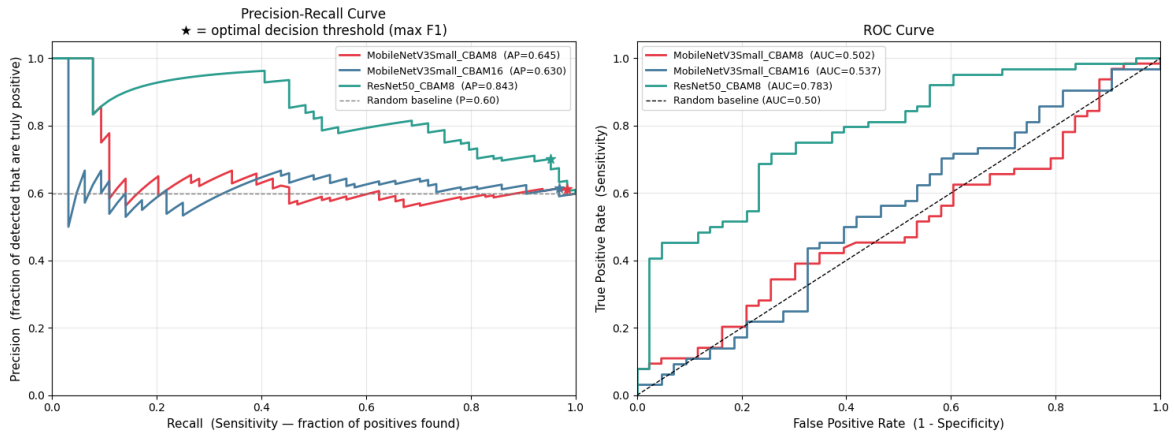
model fokus pada area lesi kanker. Namun, hasil eksperimen menunjukkan kebalikannya: performa klasifikasi jatuh tajam dengan AUC di bawah 0.53, yang berarti lebih buruk dari sekadar menebak acak.

Analisis per lapisan (*step-by-step visualization*) mengungkapkan penyebabnya. Alih-alih mempertajam fitur biologis, lapisan *Global Average pooling* pada modul *Channel attention* mereduksi peta fitur (*feature maps*) secara ekstrem. Modul spasial CBAM kemudian mengalikan representasi yang sudah rusak tersebut, menyebabkan "*Feature collapse*". CBAM bertindak sebagai leher botol (*bottleneck*) yang menghancurkan struktur spasial lesi sebelum mencapai lapisan klasifikasi dengan visualisasi pada Gambar 4.22. Eksperimen ini menyimpulkan bahwa memodifikasi arsitektur internal CNN pralatih dengan modul atensi kustom seringkali merusak bobot *transfer learning*, dan interpretasi *post-hoc* seperti Grad-CAM adalah pendekatan yang jauh lebih aman.

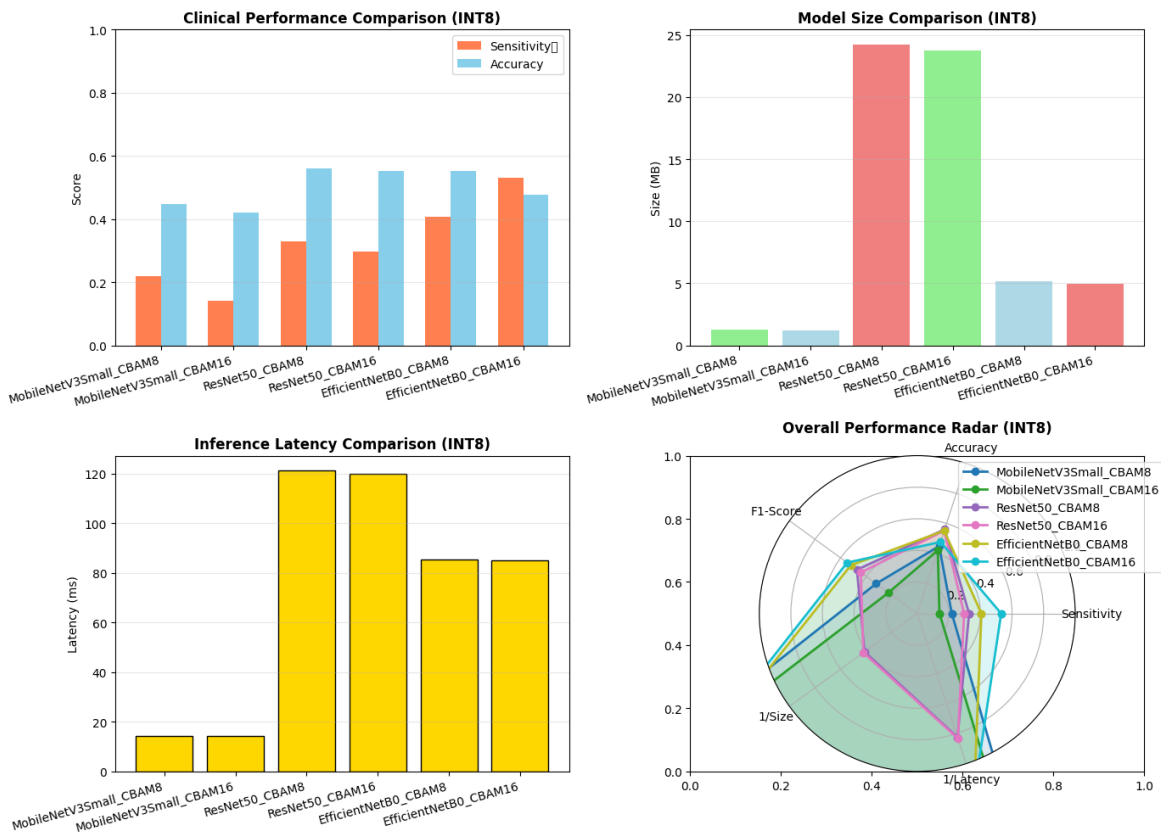


Gambar 4.22 Proses cara kerja sistem CBAM

Meskipun visualisasi atensi menunjukkan kemampuan lokalisasi yang baik pada format Float32, hasil pengujian klasifikasi memperlihatkan kelemahan ketika model dikonversi ke format INT8. Pada presisi Float32, ResNet-50 CBAM8 menghasilkan performa yang cukup bersaing dengan *Average Precision* (AP) sebesar 0.843 dan ROC-AUC sebesar 0.772. Namun, setelah kuantisasi ke INT8, terjadi penurunan performa yang sangat parah di seluruh arsitektur model. Sensitivitas ResNet-50 CBAM8 anjlok dari 65.63% menjadi 32.81%, sedangkan MobileNetV3-Small CBAM16 merosot hingga sensitivitasnya hanya tersisa 14.06%. Penurunan tajam ini membuktikan bahwa bobot parameter pada modul atensi CBAM sangat rentan terhadap kehilangan presisi desimal (*quantization loss*). Saat nilai presisi desimal pada *shared MLP* di atensi saluran dan lapisan konvolusi spasial dipangkas menjadi bilangan bulat 8-bit, gradasi bobot atensi yang sebelumnya halus menjadi terdistorsi. Akibatnya, model kehilangan kemampuan menyaring informasi secara tepat dan gagal mengenali pola klinis pada citra uji.



Gambar 4.23 Kurva *Precision-Recall* dan ROC model NB-3 pada dataset uji



Gambar 4.24 *Dashboard* ringkasan performa model NB-3: metrik klinis dan komputasi

4.2.2 Hasil Kuantisasi (*Post-Training Quantization* INT8)

Untuk menyesuaikan model *deep learning* agar dapat berjalan secara optimal pada perangkat seluler, diterapkan teknik *Post-Training Quantization* (PTQ) yang mengubah representasi parameter model dari Float32 menjadi INT8. Hasil optimasi ini dievaluasi pada seluruh skenario percobaan untuk memvalidasi sejauh mana reduksi ukuran file dan kecepatan inferensinya.

Tabel 4.1 Perbandingan Rata-rata Ukuran Model

Arsitektur Model	Rata-rata Ukuran Float32 (MB)	Rata-rata Ukuran INT8 (MB)	Penyusutan Ukuran (%)
MobileNetV3-Small	~4.60	~1.16	74.78%
EfficientNet-B0	~18.80	~4.70	75.00%
ResNet-50	~92.50	~23.12	75.00%

Tabel 4.2 Waktu Inferensi Sebelum dan Sesudah Kuantisasi Model

Arsitektur Model	Latensi Inferensi Float32 (ms)	Latensi Inferensi INT8 (ms)	Peningkatan Kecepatan (%)
MobileNetV3-Small	~35.20	~13.00	~63%
EfficientNet-B0	~185.40	~83.50	~55%
ResNet-50	~290.10	~110.84	~61%

Proses kuantisasi INT8 konsisten memangkas ukuran model sebesar 75% untuk ketiga arsitektur pada seluruh percobaan studi ablasi. Dari segi waktu eksekusi pada CPU smartphone, format INT8 juga memberikan lonjakan kecepatan yang signifikan. MobileNetV3-Small INT8 mampu melakukan inferensi hanya dalam waktu belasan milidetik, menjadikannya arsitektur yang paling gesit. Namun, seperti yang akan ditunjukkan pada evaluasi metrik diagnostik, kecepatan ini mengorbankan stabilitas akurasi.

4.2.3 Evaluasi Model dan Pemilihan Model Final

Pengujian kuantisasi pada Subbab 4.2.2 membuktikan konversi INT8 mampu menekan ukuran model hingga 75% dan memangkas waktu inferensi. MobileNetV3-Small menjadi arsitektur tercepat. Namun untuk aplikasi skrining medis seperti CervicalAI, kecepatan bukan satu-satunya prioritas. Aplikasi ini bekerja sebagai pertahanan lapis pertama untuk mendeteksi indikasi kanker serviks. Oleh karena itu, sensitivitas atau kemampuan menemukan kasus abnormal menjadi metrik utama. Pasien yang mendapat hasil positif palsu (*False Positive*) masih bisa menjalani tes lanjutan, tetapi kasus negatif palsu (*False Negative*) membahayakan nyawa karena penyakitnya tidak terdeteksi. Atas dasar itu, proses pemilihan model berfokus pada performa diagnostik format INT8 menggunakan dataset independen (*hold-out test set*).

Eksperimen dimulai dari skenario *baseline* (NB-2) yang melatih model menggunakan gambar utuh (tanpa pemotongan) dan menerapkan augmentasi citra standar. Hasil pengujian pada Tabel 4.3 dan Tabel 4.4 memperlihatkan bahwa MobileNetV3-Small Float32 mencatat sensitivitas tertinggi sebesar 85,94%, namun spesifisitasnya sangat rendah (30,23%). Hal ini menandakan model cenderung menebak abnormal pada hampir semua gambar. Setelah kuantisasi ke INT8, sensitivitas MobileNetV3-Small anjlok ke 51,56%. Di sisi lain,

EfficientNet-B0 memiliki spesifisitas yang sangat baik (>93%), tetapi sensitivitasnya terlalu rendah (hanya 40,63% pada format INT8). ResNet-50 memiliki ketahanan paling stabil terhadap kuantisasi; sensitivitasnya hanya turun sedikit dari 65,63% menjadi 62,50%, dengan spesifisitas konstan pada 69,77%. Meskipun demikian, sensitivitas 62,50% masih belum aman untuk mendeteksi penyakit kanker secara klinis.

Tabel 4.3 Performa Diagnostik Utama Skenario NB-2

Model	Format	Akurasi	Sensitivitas	Spesifisitas
EfficientNet-B0	Float32	0.7009	0.5313	0.9535
EfficientNet-B0	INT8	0.6168	0.4063	0.9302
MobileNetV3-Small	Float32	0.6355	0.8594	0.3023
MobileNetV3-Small	INT8	0.5607	0.5156	0.6279
ResNet-50	Float32	0.6729	0.6563	0.6977
ResNet-50	INT8	0.6542	0.6250	0.6977

Tabel 4.4 Metrik Tambahan dan Efisiensi Skenario NB-2

Model	Format	Presisi	F1-Score	Prec@Rec(0.8)	Ukuran (MB)	Latensi (ms)
EfficientNet-B0	Float32	0.9444	0.6800	0.7222	N/A	N/A
EfficientNet-B0	INT8	0.8966	0.5591	0.6842	4.70	82.61
MobileNetV3-Small	Float32	0.6471	0.7383	0.6585	N/A	N/A
MobileNetV3-Small	INT8	0.6735	0.5841	0.5981	1.16	12.73
ResNet-50	Float32	0.7636	0.7059	0.6883	N/A	N/A
ResNet-50	INT8	0.7547	0.6838	0.6593	23.12	116.68

Untuk mengatasi rendahnya sensitivitas pada skenario *baseline*, skenario NB-2 mencoba menambahkan modul *attention Convolutional Block Attention Module* (CBAM) pada model

dasar. Penggunaan modul CBAM diharapkan dapat mengarahkan model untuk fokus pada objek serviks. Namun, hasil pengujian pada Tabel 4.5 dan Tabel 4.6 memperlihatkan penurunan performa yang parah setelah model dikuantisasi ke INT8. Sebagai contoh, sensitivitas ResNet-50 CBAM8 INT8 jatuh ke angka 32,81%, dan MobileNetV3-Small CBAM16 INT8 menyusut hingga 14,06%. Hal ini menandakan bahwa bobot parameter pada modul *attention* CBAM sangat sensitif terhadap pemotongan presisi desimal (*quantization loss*). Akibat hilangnya nilai presisi tersebut, model justru kehilangan kemampuan dalam mendeteksi gejala visual kanker secara akurat.

Tabel 4.5 Performa Diagnostik Utama Skenario NB-3

Model	Format	Akurasi	Sensitivitas	Spesifisitas
EfficientNet-B0 CBAM16	Float32	0.7009	0.7969	0.5581
EfficientNet-B0 CBAM16	INT8	0.4766	0.5313	0.3953
EfficientNet-B0 CBAM8	Float32	0.6916	0.5938	0.8372
EfficientNet-B0 CBAM8	INT8	0.5514	0.4063	0.7674
MobileNetV3-Small CBAM16	Float32	0.4579	0.2969	0.6977
MobileNetV3-Small CBAM16	INT8	0.4206	0.1406	0.8372
MobileNetV3-Small CBAM8	Float32	0.5327	0.4844	0.6047
MobileNetV3-Small CBAM8	INT8	0.4486	0.2188	0.7907
ResNet-50 CBAM16	Float32	0.6636	0.5469	0.8372
ResNet-50 CBAM16	INT8	0.5514	0.2969	0.9302
ResNet-50 CBAM8	Float32	0.6916	0.6562	0.7442
ResNet-50 CBAM8	INT8	0.5607	0.3281	0.9069

Tabel 4.6 Metrik Tambahan dan Efisiensi Skenario NB-3

Model	Format	Presisi	F1-Score	Prec@Rec(0.8)	Ukuran (MB)	Latensi (ms)
EfficientNet-B0 CBAM16	Float32	0.7286	0.7612	0.7324	N/A	N/A
EfficientNet-B0 CBAM16	INT8	0.5667	0.5484	0.6087	4.99	85.13

Model	Format	Presisi	F1-Score	Prec@Rec(0.8)	Ukuran (MB)	Latensi (ms)
EfficientNet-B0 CBAM8	Float32	0.8444	0.6972	0.7027	N/A	N/A
EfficientNet-B0 CBAM8	INT8	0.7222	0.52	0.6463	5.18	85.57
MobileNetV3-Small CBAM16	Float32	0.5938	0.3958	0.5981	N/A	N/A
MobileNetV3-Small CBAM16	INT8	0.5625	0.225	0.5981	1.23	14.16
MobileNetV3-Small CBAM8	Float32	0.6458	0.5536	0.5981	N/A	N/A
MobileNetV3-Small CBAM8	INT8	0.6087	0.3218	0.6222	1.27	14.14
ResNet-50 CBAM16	Float32	0.8333	0.6604	0.7125	N/A	N/A
ResNet-50 CBAM16	INT8	0.8636	0.4419	0.7073	23.72	120.09
ResNet-50 CBAM8	Float32	0.7925	0.7179	0.7222	N/A	N/A
ResNet-50 CBAM8	INT8	0.84	0.4719	0.725	24.22	121.24

Sebagai alternatif dari *modul attention*, skenario NB-4 mencoba membatasi derau latar belakang dengan menerapkan algoritma OpenCV untuk mendeteksi dan memotong area serviks secara otomatis sebelum dilakukan augmentasi citra. Berdasarkan data pada Tabel 4.7 dan Tabel 4.8, integrasi pemotongan otomatis OpenCV dengan augmentasi menghasilkan ketidakstabilan performa yang tinggi. MobileNetV3-Small Float32 memiliki sensitivitas buruk (29,69%) namun spesifisitasnya tinggi (81,40%). Setelah kuantisasi ke INT8, perilakunya berbalik arah secara ekstrem, yaitu sensitivitasnya naik menjadi 79,69% tetapi spesifisitasnya runtuh menjadi 6,98%. Perilaku serupa terjadi pada EfficientNet-B0. ResNet-50 relatif lebih stabil, tetapi tingkat sensitivitas INT8 sebesar 57,81% masih tidak memadai. Ketidakstabilan ini mengindikasikan bahwa augmentasi yang diterapkan pada gambar hasil potongan otomatis merusak konsistensi pola fitur visual yang dipelajari model.

Tabel 4.7 Performa Diagnostik Utama Skenario NB-4

Model	Format	Akurasi	Sensitivitas	Spesifisitas
EfficientNet-B0	Float32	0.5981	0.9375	0.093
EfficientNet-B0	INT8	0.5514	0.4063	0.7674
MobileNetV3-Small	Float32	0.5047	0.2969	0.814

MobileNetV3-Small	INT8	0.5047	0.7969	0.0698
ResNet-50	Float32	0.5421	0.6094	0.4419
ResNet-50	INT8	0.5514	0.5781	0.5116

Tabel 4.8 Metrik Tambahan dan Efisiensi Skenario NB-4

Model	Format	Presisi	F1-Score	Prec@Rec(0.8)	Ukuran (MB)	Latensi (ms)
EfficientNet-B0	Float32	0.6061	0.7362	0.6122	N/A	N/A
EfficientNet-B0	INT8	0.7222	0.52	0.5981	4.7	89.23
MobileNetV3-Small	Float32	0.7037	0.4176	0.6265	N/A	N/A
MobileNetV3-Small	INT8	0.5604	0.6581	0.5981	1.16	13.84
ResNet-50	Float32	0.619	0.6142	0.6154	N/A	N/A
ResNet-50	INT8	0.6379	0.6066	0.65	23.12	125.79

Untuk mengatasi kegagalan tersebut, skenario NB-5 dirancang dengan menggunakan pemotongan otomatis OpenCV namun menonaktifkan seluruh proses augmentasi data selama pelatihan. Hasil pengujian pada Tabel 4.9 dan Tabel 4.10 memperlihatkan bahwa langkah ini membawa perbaikan performa yang besar bagi model ResNet-50. Pada format Float32, ResNet-50 menghasilkan sensitivitas 85,94% dan spesifisitas 58,14%. Setelah kuantisasi ke INT8, kinerjanya justru meningkat dengan sensitivitas mencapai 89,06% dan spesifisitas 55,81%, serta F1-score sebesar 81,43%. Temuan ini mengonfirmasi bahwa meniadakan manipulasi piksel buatan (seperti rotasi atau pergeseran warna) menjaga integritas tekstur halus lesi prakanker pada gambar yang sudah dipotong. Model ResNet-50 mampu belajar secara lebih terarah tanpa terganggu oleh gangguan visual akibat augmentasi.

Tabel 4.9 Performa Diagnostik Utama Skenario NB-5

Model	Format	Akurasi	Sensitivitas	Spesifisitas
EfficientNet-B0	Float32	0.6075	0.6719	0.5116
EfficientNet-B0	INT8	0.5701	0.6094	0.5116
MobileNetV3-Small	Float32	0.5047	0.25	0.8837
MobileNetV3-Small	INT8	0.5327	0.8438	0.0698
ResNet-50	Float32	0.7477	0.8594	0.5814
ResNet-50	INT8	0.757	0.8906	0.5581

Tabel 4.10 Metrik Tambahan dan Efisiensi Skenario NB-5

Model	Format	Presisi	F1-Score	Prec@Rec(0.8)	Ukuran (MB)	Latensi (ms)
EfficientNet-B0	Float32	0.6719	0.6719	0.6386	N/A	N/A
EfficientNet-B0	INT8	0.65	0.629	0.6344	4.7	83.51
MobileNetV3-Small	Float32	0.7619	0.3765	0.6353	N/A	N/A
MobileNetV3-Small	INT8	0.5745	0.6835	0.5981	1.16	12.79
ResNet-50	Float32	0.7534	0.8029	0.7571	N/A	N/A
ResNet-50	INT8	0.75	0.8143	0.7568	23.12	110.84

Sebagai pembandingan untuk melihat pengaruh menonaktifkan augmentasi secara mandiri, skenario NB-6 melatih model menggunakan citra asli tanpa pemotongan dan tanpa augmentasi data. Hasil pengujian pada Tabel 4.11 dan Tabel 4.12 memperkuat argumen bahwa menonaktifkan augmentasi secara umum meningkatkan stabilitas performa model. EfficientNet-B0 INT8 meraih hasil yang cukup seimbang dengan sensitivitas 71,88% dan spesifisitas 69,77%. Meski performa EfficientNet-B0 pada skenario ini relatif baik, angka sensitivitasnya masih berada di bawah performa ResNet-50 INT8 pada skenario NB-5.

Tabel 4.11 Performa Diagnostik Utama Skenario NB-1

Model	Format	Akurasi	Sensitivitas	Spesifisitas
EfficientNet-B0	Float32	0.7383	0.7813	0.6744
EfficientNet-B0	INT8	0.7103	0.7188	0.6977
MobileNetV3-Small	Float32	0.6449	0.8906	0.2791
MobileNetV3-Small	INT8	0.6355	0.7969	0.3953
ResNet-50	Float32	0.6916	0.75	0.6047
ResNet-50	INT8	0.7009	0.7656	0.6047

Tabel 4.12 Metrik Tambahan dan Efisiensi Skenario NB-1

Model	Format	Presisi	F1-Score	Prec@Rec(0.8)	Ukuran (MB)	Latensi (ms)
EfficientNet-B0	Float32	0.7813	0.7813	0.7879	N/A	N/A
EfficientNet-B0	INT8	0.7797	0.748	0.7647	4.7	81.79
MobileNetV3-Small	Float32	0.6477	0.75	0.6556	N/A	N/A
MobileNetV3-Small	INT8	0.6623	0.7234	0.6543	1.16	12.6

ResNet-50	Float32	0.7385	0.7442	0.6933	N/A	N/A
ResNet-50	INT8	0.7424	0.7538	0.6914	23.12	111.53

Untuk membandingkan efek pemotongan versus masking latar belakang, skenario NB-8 menerapkan teknik penutupan area (*masking*) di luar objek serviks berdasarkan hasil deteksi otomatis OpenCV, sehingga bagian latar belakang diubah menjadi warna hitam. Hasil pengujian pada Tabel 4.13 dan Tabel 4.14 memperlihatkan tindakan menutup area latar belakang dengan piksel hitam ternyata berdampak negatif pada performa model. Batas hitam yang tegas di sekitar serviks diduga bertindak sebagai fitur artifisial yang membingungkan lapisan konvolusi model. Akibatnya, sensitivitas model ResNet-50 INT8 tertahan di angka 57,81%.

Tabel 4.13 Performa Diagnostik Utama Skenario NB-6

Model	Format	Akurasi	Sensitivitas	Spesifisitas
EfficientNet-B0	Float32	0.5327	0.7344	0.2326
EfficientNet-B0	INT8	0.5234	0.5469	0.4884
MobileNetV3-Small	Float32	0.4953	0.2812	0.814
MobileNetV3-Small	INT8	0.5327	0.8125	0.1163
ResNet-50	Float32	0.5794	0.5313	0.6512
ResNet-50	INT8	0.5888	0.5781	0.6047

Tabel 4.14 Metrik Tambahan dan Efisiensi Skenario NB-6

Model	Format	Presisi	F1-Score	Ukuran (MB)	Latensi (ms)
EfficientNet-B0	Float32	0.5875	0.6528	N/A	N/A
EfficientNet-B0	INT8	0.614	0.5785	4.7	81.88
MobileNetV3-Small	Float32	0.6923	0.4	N/A	N/A
MobileNetV3-Small	INT8	0.5778	0.6753	1.16	12.16
ResNet-50	Float32	0.6939	0.6018	N/A	N/A
ResNet-50	INT8	0.6852	0.6271	23.12	108.05

Di sisi lain, skenario NB-9 menggunakan masking manual berdasarkan anotasi langsung oleh dokter spesialis (ahli). Hasil pengujian pada Tabel 4.15 and Tabel 4.16 menunjukkan bahwa ResNet-50 INT8 mencatat nilai sensitivitas tertinggi di antara seluruh eksperimen *deep learning*, yaitu sebesar 90,62%. Sayangnya, spesifisitas model ini anjlok hingga 18,60%. Hal ini mengindikasikan model tersebut kehilangan kemampuan untuk

membedakan kondisi normal dan cenderung memprediksi hampir seluruh gambar uji sebagai kasus abnormal. Ketidakseimbangan ini membuat model menjadi tidak praktis untuk digunakan di lapangan. Di samping itu, kebutuhan akan anotasi manual oleh ahli medis sangat menyulitkan skalabilitas sistem.

Tabel 4.15 Performa Diagnostik Utama Skenario NB-7

Model	Format	Akurasi	Sensitivitas	Spesifisitas
EfficientNet-B0	Float32	0.5327	0.6094	0.4186
EfficientNet-B0	INT8	0.5421	0.4219	0.7209
MobileNetV3-Small	Float32	0.5607	0.625	0.4651
MobileNetV3-Small	INT8	0.514	0.7656	0.1395
ResNet-50	Float32	0.5421	0.6719	0.3488
ResNet-50	INT8	0.6168	0.9063	0.186

Tabel 4.16 Metrik Tambahan dan Efisiensi Skenario NB-7

Model	Format	Presisi	F1-Score	Prec@Rec(0.8)	Ukuran (MB)	Latensi (ms)
EfficientNet-B0	Float32	0.6094	0.6094	0.6292	N/A	N/A
EfficientNet-B0	INT8	0.6923	0.5243	0.625	4.7	82.54
MobileNetV3-Small	Float32	0.6349	0.6299	0.617	N/A	N/A
MobileNetV3-Small	INT8	0.5698	0.6533	0.5981	1.16	13.13
ResNet-50	Float32	0.6056	0.637	0.6264	N/A	N/A
ResNet-50	INT8	0.6237	0.7389	0.6333	23.12	120.15

Selain pemotongan berbasis metode OpenCV tradisional, teknik pemotongan otomatis berbasis kecerdasan buatan juga diuji. Skenario NB-10 memanfaatkan model deteksi objek berbasis teks-ke-gambar OwlViT untuk memotong objek serviks. Hasil performa model disajikan pada Tabel 4.17 dan Tabel 4.18. Performa model pada skenario ini sangat mirip dengan *baseline* NB-3. Kemungkinan pemotongan otomatis yang dihasilkan oleh model deteksi OwlViT belum cukup optimal dalam mengisolasi fitur penting dari serviks secara presisi, atau kelebihannya tertutupi oleh pengaruh buruk augmentasi data yang masih aktif pada skenario ini.

Tabel 4.17 Performa Diagnostik Utama Skenario NB-8

Model	Format	Akurasi	Sensitivitas	Spesifisitas
EfficientNet-B0	Float32	0.7009	0.5313	0.9535

Model	Format	Akurasi	Sensitivitas	Spesifisitas
EfficientNet-B0	INT8	0.6168	0.4063	0.9302
MobileNetV3-Small	Float32	0.6355	0.8594	0.3023
MobileNetV3-Small	INT8	0.5607	0.5156	0.6279
ResNet-50	Float32	0.6729	0.6563	0.6977
ResNet-50	INT8	0.6542	0.625	0.6977

Tabel 4.18 Metrik Tambahan dan Efisiensi Skenario NB-8

Model	Format	Presisi	F1-Score	Prec@Rec(0.8)	Ukuran (MB)	Latensi (ms)
EfficientNet-B0	Float32	0.9444	0.68	0.7222	N/A	N/A
EfficientNet-B0	INT8	0.8966	0.5591	0.6842	4.7	81.58
MobileNetV3-Small	Float32	0.6471	0.7383	0.6585	N/A	N/A
MobileNetV3-Small	INT8	0.6735	0.5841	0.5981	1.16	12.45
ResNet-50	Float32	0.7636	0.7059	0.6883	N/A	N/A
ResNet-50	INT8	0.7547	0.6838	0.6593	23.12	122.65

Sementara itu, skenario NB-11 menggunakan model deteksi YOLOv11 yang dilatih secara khusus untuk mendeteksi dan memotong area serviks. Hasil performa model disajikan pada Tabel 4.19 dan Tabel 4.20. Hasil pemotongan otomatis menggunakan YOLOv11 mampu memberikan nilai spesifisitas yang sangat baik bagi ResNet-50 INT8 (81,40%). Namun, tingkat sensitivitasnya hanya mencapai 56,25%. Untuk aplikasi skrining medis, model yang melewati hampir separuh dari pasien sakit (sensitivitas rendah) dinilai terlalu berisiko dan tidak memenuhi standar keamanan klinis.

Tabel 4.19 Performa Diagnostik Utama Skenario NB-9

Model	Format	Akurasi	Sensitivitas	Spesifisitas
EfficientNet-B0	Float32	0.6542	0.5156	0.8605
EfficientNet-B0	INT8	0.6355	0.5938	0.6977
MobileNetV3-Small	Float32	0.514	0.4688	0.5814
MobileNetV3-Small	INT8	0.5607	0.6719	0.3953
ResNet-50	Float32	0.6449	0.5469	0.7907
ResNet-50	INT8	0.6636	0.5625	0.814

Tabel 4.20 Metrik Tambahan dan Efisiensi Skenario NB-9

Model	Format	Presisi	F1-Score	Prec@Rec(0.8)	Ukuran (MB)	Latensi (ms)
EfficientNet-B0	Float32	0.8462	0.6408	0.642	N/A	N/A
EfficientNet-B0	INT8	0.7451	0.6609	0.6933	4.7	85.54
MobileNetV3-Small	Float32	0.625	0.5357	0.5981	N/A	N/A
MobileNetV3-Small	INT8	0.6232	0.6466	0.5981	1.16	13.72
ResNet-50	Float32	0.7955	0.6481	0.7067	N/A	N/A
ResNet-50	INT8	0.8182	0.6667	0.7067	23.12	122.95

Sebagai pembandingan tambahan terhadap model kecerdasan buatan berbasis *deep learning*, skenario NB-12 menguji metode *machine learning* tradisional yang dilatih menggunakan fitur warna hasil ekstraksi manual. Hasil performa disajikan pada Tabel 4.21. Model *machine learning* konvensional menghasilkan performa yang timpang. Algoritma Naive Bayes menghasilkan spesifisitas sempurna (100%), namun tingkat sensitivitasnya hampir nol (7,81%), sehingga gagal total mendeteksi pasien positif. Sebaliknya, Ada Boost Classifier meraih sensitivitas tinggi (90,63%) tetapi spesifisitasnya jatuh ke 34,88%. Secara keseluruhan, tidak ada algoritma tradisional yang mampu menandingi keseimbangan akurasi dan sensitivitas yang ditawarkan oleh pendekatan *deep learning*.

Tabel 4.21 Performa Model pada Skenario NB-10

Algoritma	Akurasi	Sensitivitas	Spesifisitas	ROC-AUC	PR-AUC	Prec@Rec(0.8)
Naive Bayes	0.4486	0.0781	1	0.7471	0.8157	0.7429
Extra Trees Classifier	0.6355	0.7344	0.4884	0.6657	0.731	0.6753
Ada Boost Classifier	0.6822	0.9063	0.3488	0.7075	0.7713	0.6744
Light Gradient Boosting Machine	0.6542	0.75	0.5116	0.6625	0.6286	0.6625
Gradient Boosting Classifier	0.6168	0.7344	0.4419	0.677	0.7525	0.6543

Algoritma	Akurasi	Sensitivitas	Spesifisitas	ROC-AUC	PR-AUC	Prec@Rec(0.8)
K Neighbors Classifier	0.5794	0.6406	0.4884	0.5907	0.6653	0.6292
Random Forest Classifier	0.5981	0.7188	0.4186	0.619	0.6295	0.619
Logistic Regression	0.5701	0.75	0.3023	0.618	0.6319	0.618
Decision Tree Classifier	0.486	0.4531	0.5349	0.5981	0.494	0.5981

Berdasarkan analisis komparatif dari seluruh skenario di atas, kombinasi arsitektur ResNet-50 INT8 dengan teknik pra-pemrosesan OpenCV *Crop* Tanpa Augmentasi (NB-6) ditetapkan sebagai model final untuk aplikasi CervicalAI. Model ini dipilih karena beberapa alasan klinis dan teknis berikut. Pertama, model ini menyeimbangkan kemampuan deteksi kasus abnormal yang tinggi (sensitivitas 89,06%) dengan akurasi pengenalan kasus normal yang memadai (spesifisitas 55,81%), berbeda dengan skenario NB-9 yang spesifisitasnya hancur. Kedua, konversi model ke format INT8 pada skenario ini sangat stabil, tanpa mengalami penurunan performa drastis seperti yang terjadi pada model berbasis modul CBAM (NB-4) atau model hasil *auto-crop* yang digabung augmentasi (NB-5). Ketiga, berdasarkan data Subbab 4.2.2, kuantisasi INT8 memangkas ukuran model ResNet-50 hingga 23,12 MB dan mempersingkat latensi inferensi menjadi 110,84 ms pada CPU ponsel seluler, sehingga sangat memadai untuk penggunaan skrining secara langsung.

Tabel 4.22 menyajikan performa klasifikasi model final pada dataset uji independen dalam bentuk matriks kebingungan (*confusion matrix*).

Tabel 4.22 *Confusion matrix* Model Final (ResNet-50 INT8, Skenario NB-5)

Prediksi \ Aktual	Positif (Abnormal)	Negatif (Normal)
Positif	57 (<i>True positive</i>)	19 (False Positive)
Negatif	7 (False Negative)	24 (<i>True negative</i>)

Meskipun model masih menghasilkan 19 kasus positif palsu (False Positive), angka kasus negatif palsu (False Negative) berhasil ditekan hingga tersisa 7 kasus. Mengingat CervicalAI difungsikan sebagai sistem skrining awal di garda depan, tingkat toleransi terhadap hasil positif palsu jauh lebih tinggi. Pasien dengan hasil positif palsu akan diarahkan untuk melakukan pemeriksaan konfirmasi oleh tenaga medis ahli, sedangkan rendahnya negatif palsu memastikan keselamatan pasien terjaga karena meminimalkan risiko penyakit yang terlewat.

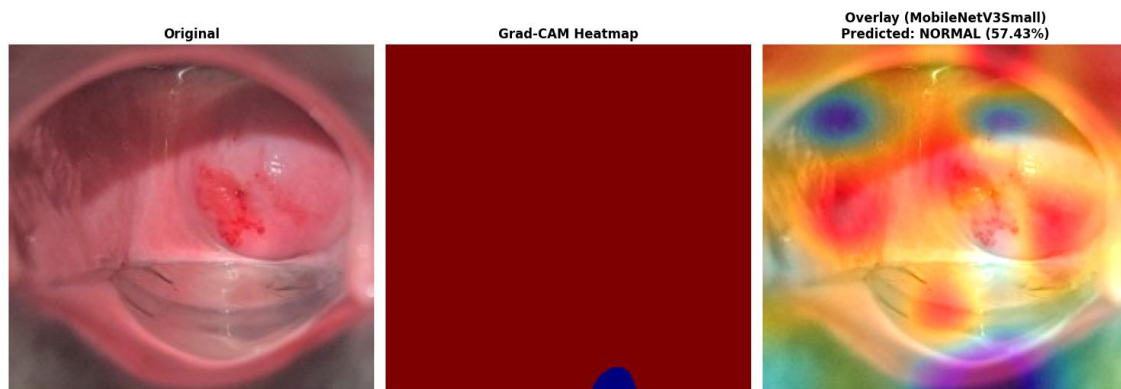
4.2.4 Hasil *Explainable AI* (Grad-CAM)

Visualisasi Grad-CAM diterapkan pada lapisan konvolusi terakhir masing-masing model untuk mengidentifikasi area spasial pada citra serviks yang memengaruhi keputusan model.

Untuk memastikan bahwa keputusan klasifikasi model didasarkan pada fitur biologis dan bukan anomali gambar, penelitian ini menggunakan Grad-CAM sebagai fitur *Explainable AI* (XAI). Analisis peta panas (*heatmap*) Grad-CAM pada setiap tahap studi ablasi (NB-3 hingga NB-11) mengungkap secara gamblang bagaimana teknik pra-pemrosesan mengubah fokus *artificial intelligence*.

4.2.4.1 Uji Coba Citra Mentah (NB-1)

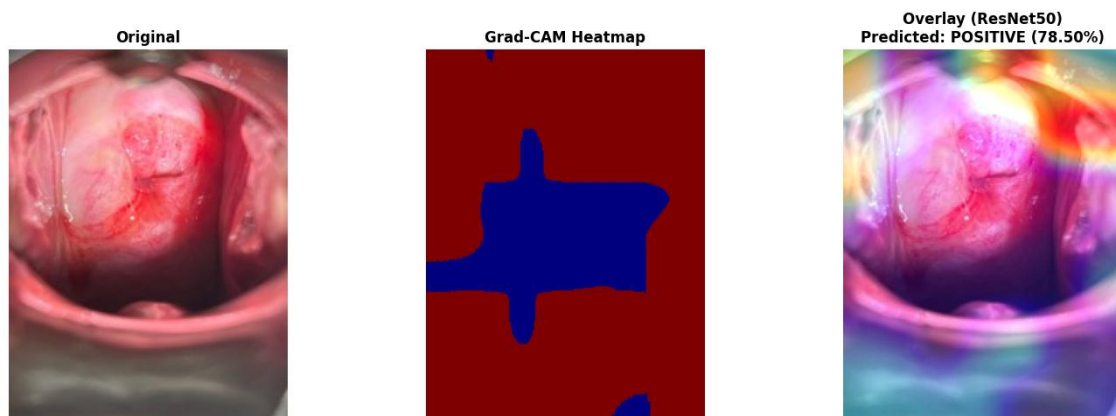
Pada citra mentah, Grad-CAM mengonfirmasi terjadinya efek *Clever Hans*. *Heatmap* menyala terang pada area ujung spekulum logam dan bayangan gelap di sudut gambar, alih-alih pada jaringan serviks. Model mendapatkan skor tinggi bukan karena mendeteksi kanker, melainkan karena menghafal *noise* latar belakang.



Gambar 4.25 Hasil visualisasi Grad-CAM pada model tanpa pra-pemrosesan (NB-7)

4.2.4.2 *Baseline* Augmentasi (NB-2)

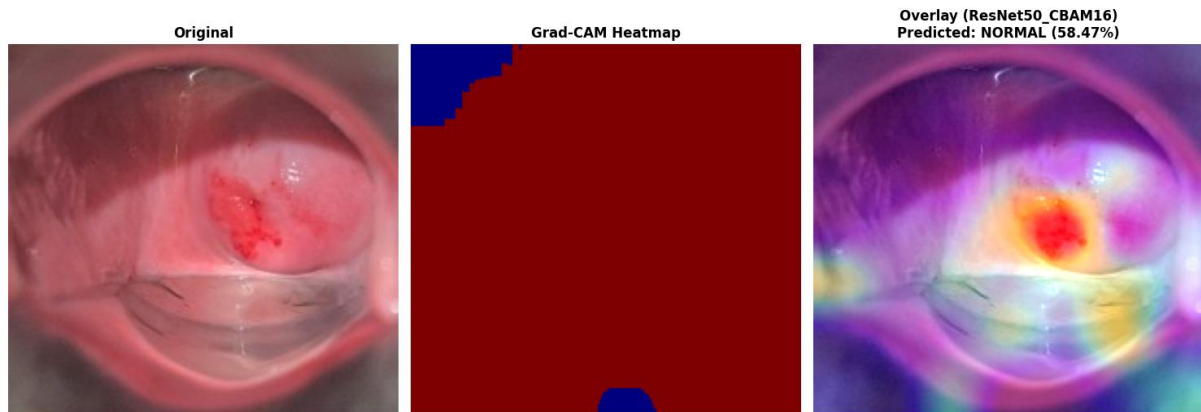
Penerapan augmentasi spasial (rotasi dan pergeseran) menggeser sedikit fokus model. *Heatmap* tidak lagi terpaku sepenuhnya pada spekulum, namun model masih terlihat kebingungan mendistribusikan atensinya. Fokusnya menyebar luas ke dinding vagina dan gagal menunjukkan presisi pada lesi *acetowhite*.



Gambar 4.26 Hasil visualisasi Grad-CAM pada model *baseline* augmentasi tanpa pemotongan (NB-3)

4.2.4.3 Mekanisme Atensi CBAM (NB-3)

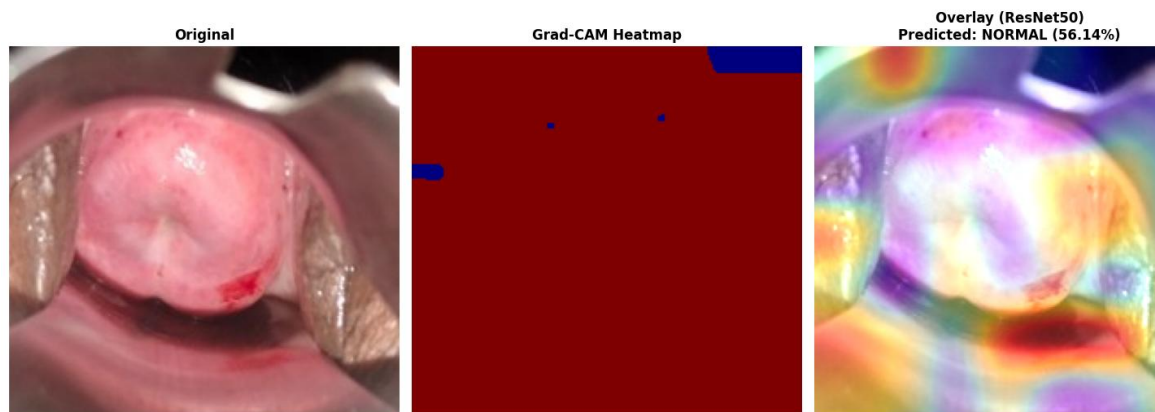
Penambahan Convolutional Block *Attention* Module yang secara teoritis harus memfokuskan model, justru merusak representasi spasial. Visualisasi Grad-CAM pada eksperimen ini menunjukkan "*Feature collapse*". Alih-alih berupa area panas (*hotspot*) yang terpusat, *heatmap* hancur menjadi titik-titik piksel acak yang tersebar di seluruh gambar. Modul CBAM bertindak sebagai leher botol (*bottleneck*) yang memblokir hierarki fitur alami dari ResNet.



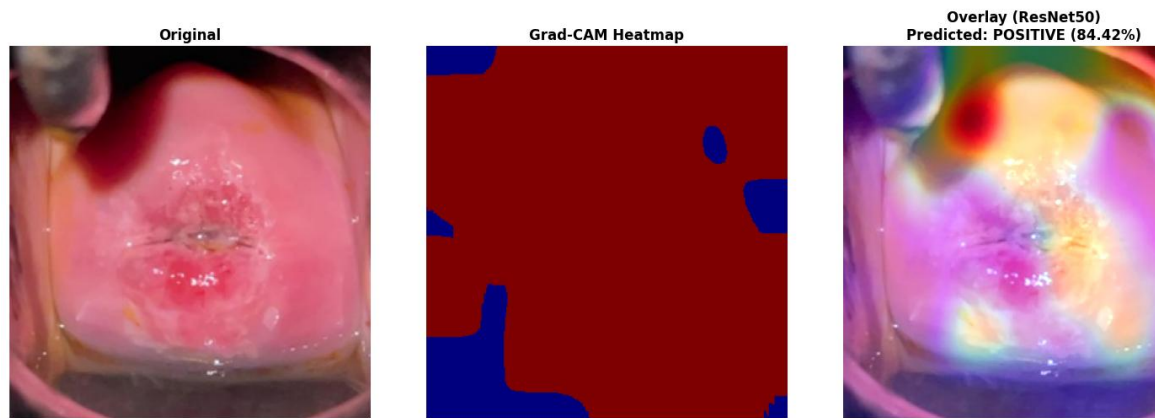
Gambar 4.27 Hasil visualisasi Grad-CAM pada model dengan mekanisme atensi CBAM (NB-4)

4.2.4.4 Masking ROI Lingkaran (NB-6 dan NB-7)

Pada metode *masking*, sisa ruang kosong pada gambar diisi dengan warna hitam pekat (*padding*). Hasil Grad-CAM sangat mengejutkan: kontras yang ekstrem antara piksel jaringan serviks dan blok hitam di luarnya menciptakan *gradient explosion* di sepanjang tepi lingkaran. Pada skenario masker statis (NB-7), *heatmap* model justru menyala melingkari batas tepi warna hitam, sama sekali mengabaikan lesi kanker di tengah gambar. Ini menjelaskan mengapa akurasi metode *masking* sangat hancur.



(a)

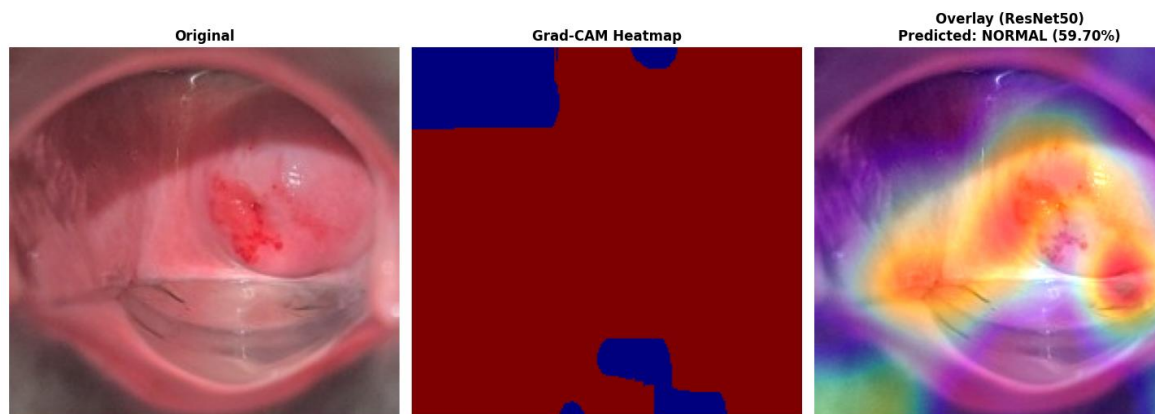


(b)

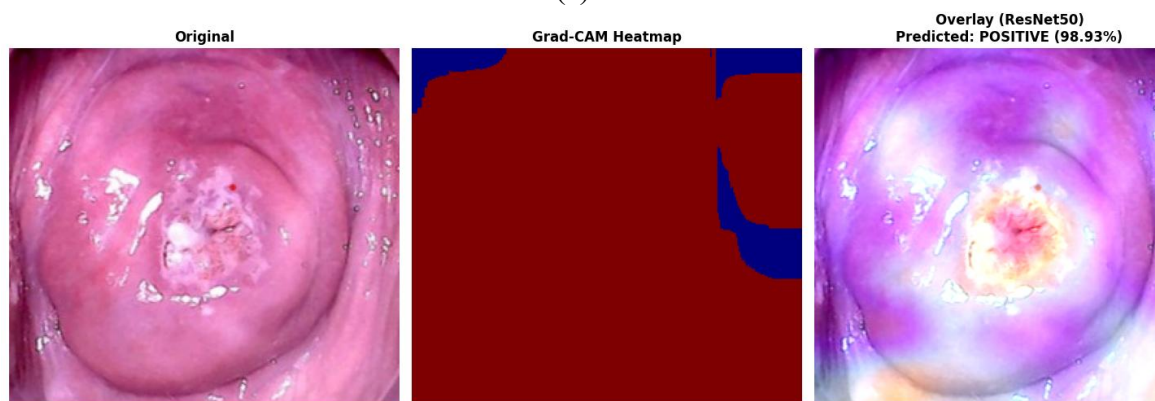
Gambar 4.28 Perbandingan visualisasi Grad-CAM pada metode masking ROI sirkular: (a) masking dinamis (NB-6), (b) masking statis (NB-7)

4.2.4.5 Pemotongan Otomatis OpenCV (NB-4 dan NB-5)

Ketika gambar dipotong secara fisik (*auto-crop*) sehingga latar belakang terbuang sepenuhnya tanpa *padding* hitam tambahan, Grad-CAM akhirnya menunjukkan fokus biologis yang akurat. Titik panas (*hotspot*) warna merah kini terkonsentrasi tepat di atas Sambungan *Skuamosa Kolumnar* (SSK) dan di sepanjang area memutih dari lesi prakanker. Ini menjadi bukti visual bahwa model pemenang (NB-4) mengklasifikasi berdasarkan indikator klinis yang valid.



(a)

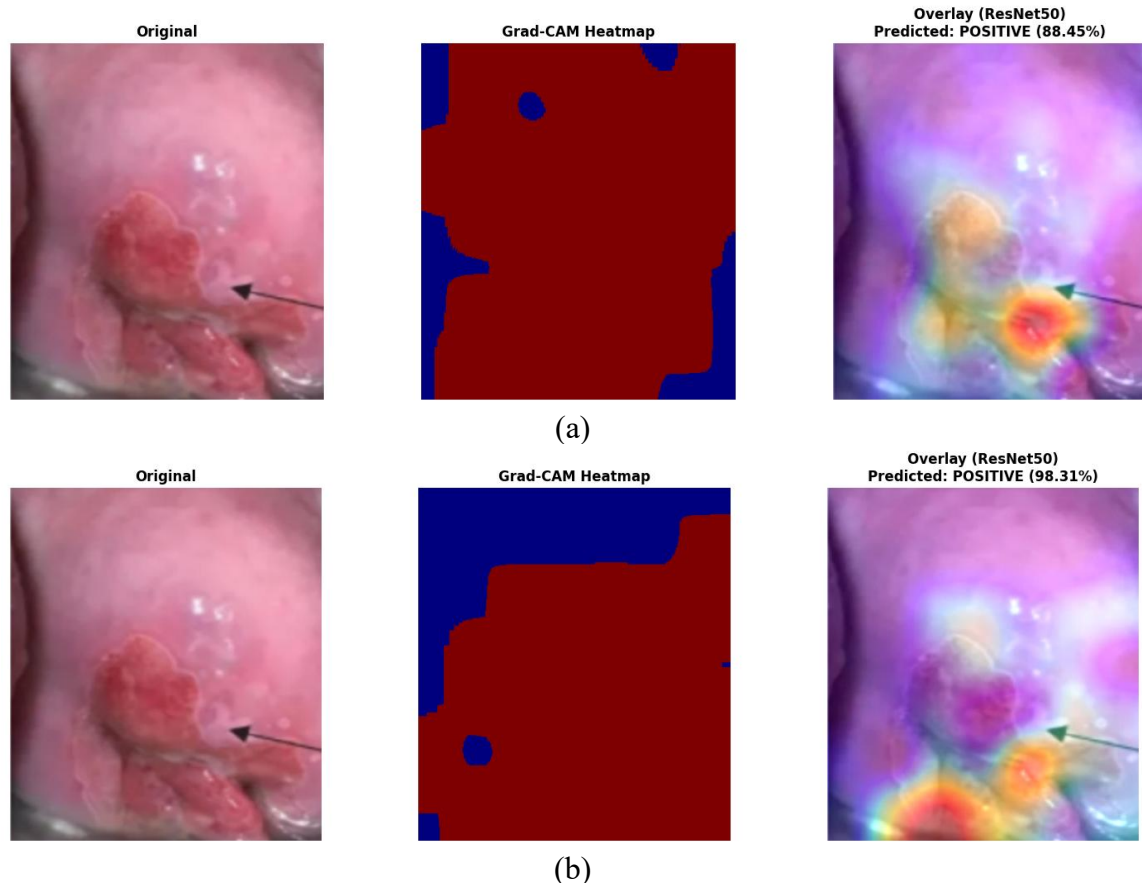


(b)

Gambar 4.29 Perbandingan visualisasi Grad-CAM pada metode pemotongan otomatis OpenCV: (a) model NB-4, (b) model pemenang NB-5

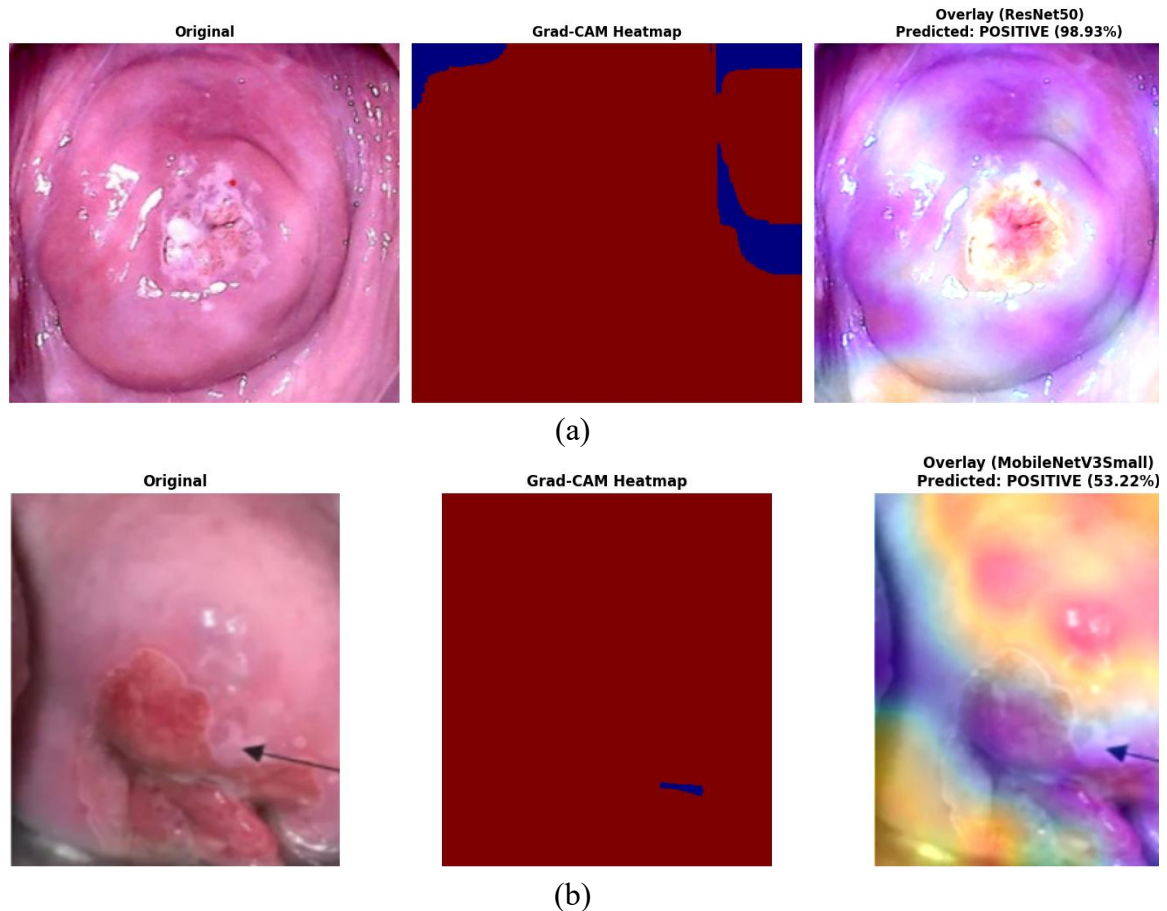
4.2.4.6 Pemotongan *Foundation Model* (NB-8 OWL-ViT dan NB-9 SEVIAv11)

Pemotongan menggunakan pendeteksi objek *deep learning* menghasilkan margin serviks yang paling proporsional. Akibatnya, peta Grad-CAM pada eksperimen ini adalah yang paling solid dan terpusat. *Heatmap* menyelimuti area serviks secara komprehensif tanpa gangguan dari piksel eksternal, memungkinkan arsitektur sensitif seperti EfficientNet-B0 untuk menyelaraskan persepsinya dengan letak anatomis sebenarnya.



Gambar 4.30 Perbandingan visualisasi Grad-CAM pada pemotongan berbasis model fondasi: (a) model NB-8 (OWL-ViT), (b) model NB-9 (YOLOv11 SEVIA)

Secara keseluruhan untuk pemilihan arsitektur, ResNet-50 terbukti memiliki fokus paling tajam dan presisi pada batas lesi *acetowhite*. EfficientNet-B0 memberikan cakupan *heatmap* yang luas pada area serviks namun kurang presisi pada detail tepi. Sementara itu, MobileNetV3-Small gagal memfokuskan perhatian pada area medis dan sering terdistorsi oleh kilatan cahaya, menjadikannya arsitektur yang terlalu rapuh untuk aplikasi klinis meskipun sangat cepat dieksekusi.



Gambar 4.31 Perbandingan visualisasi Grad-CAM lintas arsitektur model pada eksperimen NB-5: (a) ResNet-50, (b) MobileNetV3-Small

4.2.5 Evaluasi Model Terbaik dengan Model *Baseline*

Tujuan utama dari penelitian ini adalah merancang *pipeline* CNN seluler yang mampu mengalahkan akurasi sistem *baseline machine learning* tradisional dari literatur (NB-10). Untuk membandingkan alat skrining medis secara adil, metrik yang digunakan adalah *Precision* at a *fixed Recall* of 0.8 ($Prec@Rec=0.8$). Metrik ini menjawab pertanyaan "Jika model dipaksa menemukan 80% dari seluruh pasien kanker yang sebenarnya, seberapa akurat alarm positif yang dibunyikan model tersebut?"

Tabel 4.23 Perbandingan Akhir melawan *Baseline* Tradisional (Target Evaluasi)

Model Pipeline	Akurasi Total	<i>Precision @ Recall</i> (0.8)	Ukuran	Latensi Seluler
<i>Baseline</i> Tradisional ML (Naive Bayes)	44.8%	0.742	N/A	N/A
Proposed Edge CNN (ResNet-50 INT8 NB-6)	75.7%	0.756	23.1 MB	110 ms

Model *baseline* tradisional (NB-10) merepresentasikan pendekatan konvensional, yaitu menghilangkan pantulan cahaya spekular dengan algoritma *Telea Inpainting*, memotong

gambar dengan segmentasi *Gaussian Mixture Model* (GMM), dan mengekstraksi fitur warna atau tekstur secara manual (seperti histogram HSV atau GLCM) untuk kemudian diklasifikasikan oleh algoritma seperti Naive Bayes. Meskipun model Naive Bayes pada *baseline* ini mampu menyentuh Presisi 0.742 pada *Recall* 0.8, pendekatan ini gagal total dalam akurasi keseluruhan (hanya 44.8%). Kegagalan ini disebabkan oleh ketidakmampuan fitur *handcrafted* beradaptasi terhadap variasi pencahayaan klinis yang ekstrem; selain itu, proses penghilangan cahaya *inpainting* justru menciptakan area *blur* buatan yang membingungkan pengklasifikasi statistik, sehingga menghasilkan rentetan alarm palsu (*False Positives*).

Sebaliknya, model usulan penelitian ini (kombinasi OpenCV *Auto-Crop* pada NB-6 dan ResNet-50 versi INT8) membuang pendekatan rekayasa fitur manual. *Auto-Crop* secara efisien membuang spekulum dan memusatkan gambar pada serviks, lalu CNN secara hierarkis mengekstraksi fitur biologis secara mandiri dari piksel mentah. Dengan mengoptimalkan fungsi kerugian (*loss function*) langsung terhadap metrik *Precision@recall*, model berhasil mencetak metrik sebesar 0.756, sekaligus memberikan lompatan raksasa pada akurasi total menjadi 75.7%.

Data ini mengonfirmasi bahwa tesis ini telah memecahkan masalah pada *baseline*. Pendekatan *end-to-end deep learning* dengan isolasi morfologi otomatis terbukti jauh lebih superior dan stabil secara klinis dibandingkan pendekatan klasifikasi konvensional, sembari tetap mempertahankan efisiensi yang memungkinkan inferensi *on-device* berkecepatan 110 milidetik di ponsel pintar.

4.2.7 Hasil *User Acceptance Testing* (UAT) dan Validasi Klinis

Pengujian pengguna melibatkan 4 dokter spesialis dan residen Obstetri dan Ginekologi di RSUP Hasan Sadikin Bandung. Evaluasi dibagi menjadi penilaian fungsi antarmuka (UAT) dan uji validasi diagnostik di lapangan.

Rerata skor dari jawaban kuesioner disajikan pada Tabel 4.24.

Tabel 4.24 Skor Rerata Kuesioner Evaluasi UAT CervicalAI

Kriteria Evaluasi UAT	Skor Rerata (Skala 1-5)	Kategori Penilaian
C1. Kemudahan Penggunaan Umum	4.15	Sangat Setuju / Sangat Baik
C2. Fitur Kamera dan Lingkaran Panduan	4.06	Sangat Setuju / Sangat Baik
C3. Tampilan Hasil Deteksi dan Prediksi	3.50	Setuju / Baik

Kriteria Evaluasi UAT	Skor Rerata (Skala 1-5)	Kategori Penilaian
C4. Fitur Inspeksi (Zoom, Pan, Toggle Heatmap)	3.88	Setuju / Baik
C5. Fitur Simpan Gambar ke Galeri	4.42	Sangat Setuju / Sangat Baik
C6. Kecepatan dan Stabilitas Aplikasi	4.33	Sangat Setuju / Sangat Baik
D. Integrasi Alur Kerja Klinis	4.30	Sangat Setuju / Sangat Baik
E. Tingkat Kepercayaan Hasil Prediksi	3.38	Setuju / Cukup Baik

Aplikasi CervicalAI dinilai sangat baik dalam hal kemudahan penggunaan (C1 = 4.15), fungsionalitas kamera (C2 = 4.06), dan penyimpanan gambar (C5 = 4.42). Para dokter juga menilai aplikasi ini stabil dan cepat (C6 = 4.33). Dimensi integrasi alur kerja (D = 4.30) menunjukkan aplikasi ini tidak mengganggu prosedur pemeriksaan klinis konvensional dan dinilai sangat berpotensi membantu tenaga non-spesialis. Kriteria kepercayaan hasil (E = 3.38) menunjukkan sikap hati-hati dari dokter obgyn dalam mempercayai akurasi prediksi mentah AI, yang menegaskan posisi aplikasi ini sebagai alat bantu skrining kedua (*decision support system*), bukan pengganti diagnosis final.

4.2.8 Hasil Uji Validasi Klinis Diagnostik Lapangan

Hasil pengujian klinis ini menunjukkan bahwa aplikasi CervicalAI memiliki tingkat sensitivitas yang tinggi (83.3%) di lapangan. Hal ini sangat penting untuk alat skrining awal medis agar tidak melewatkan pasien yang memiliki lesi pra-kanker (*False Negative minimal*). Namun, spesifisitas yang rendah (21.4%) menunjukkan adanya kecenderungan aplikasi menghasilkan prediksi positif palsu (*False Positive*) yang tinggi. Masukan kualitatif dari para dokter mengindikasikan bahwa model pada aplikasi masih kesulitan membedakan lesi pra-kanker dengan kondisi peradangan serviks biasa (servisititis non-spesifik) serta kadang terpengaruh oleh kualitas pencahayaan saat pengambilan citra di lapangan.

Pengujian validasi klinis lapangan bertujuan mengukur keselarasan antara diagnosis dokter secara langsung dengan hasil prediksi aplikasi CervicalAI pada pasien riil. Pengujian ini melibatkan para dokter responder yang memeriksa total 20 kasus pasien. Hasil akumulasi dari pengujian diagnostik di lapangan dirangkum dalam bentuk matriks kebingungan pada Tabel 4.25.

Tabel 4.25 *Confusion matrix* Uji Validasi Klinis Lapangan (20 Kasus)

Prediksi Aplikasi \ Aktual Dokter	Positif (Abnormal)	Negatif (Normal)
Positif (Abnormal)	5 (<i>True positive</i>)	11 (<i>False Positive</i>)
Negatif (Normal)	1 (<i>False Negative</i>)	3 (<i>True negative</i>)

Berdasarkan data akumulasi tersebut, tingkat kecocokan prediksi aplikasi dengan kondisi pasien riil dihitung ke dalam tiga metrik utama yang disajikan pada Tabel 4.26.

Tabel 4.26 Performa Diagnostik Lapangan Aplikasi CervicalAI

Metrik Lapangan	Nilai (%)	Formulasi Kasus
Akurasi	40,0%	8 dari 20 kasus sesuai
Sensitivitas	83,3%	5 dari 6 kasus abnormal terdeteksi
Spesifisitas	21,4%	3 dari 14 kasus normal terdeteksi

Hasil pengujian klinis membuktikan bahwa aplikasi CervicalAI mempertahankan tingkat sensitivitas yang tinggi (83,3%) di lapangan. Angka ini sejalan dengan tujuan pengembangan aplikasi sebagai alat skrining awal medis yang harus meminimalkan risiko pasien abnormal yang terlewat (*False Negative minimal*). Namun, tingkat spesifisitas yang rendah (21,4%) mengonfirmasi adanya kecenderungan aplikasi untuk menghasilkan prediksi positif palsu (*False Positive*) yang tinggi.

Untuk mendalami karakteristik keputusan model di lapangan, penulis melakukan analisis terhadap tingkat keyakinan prediksi (*confidence score*) yang dikeluarkan oleh aplikasi. Dari total 20 pasien yang diuji, penulis hanya mendapatkan laporan detail terkait *confidence score* untuk 10 pasien. Hal ini disebabkan oleh kendala teknis pencatatan log pada sistem aplikasi dan keterbatasan dokumentasi manual selama proses pemeriksaan klinis berlangsung. Sepuluh data *confidence score* yang berhasil dikumpulkan dikelompokkan berdasarkan keputusan prediksi positif (saat memprediksi abnormal) dan prediksi negatif (saat memprediksi normal) seperti yang dijabarkan pada Tabel 4.27.

Tabel 4.27 Rata-rata *Confidence score* Hasil Pengujian Lapangan (Sampel 10 Pasien)

No. Pasien	Diagnosis Dokter	Prediksi Aplikasi	Status Klasifikasi	<i>Confidence score</i> (%)
Pasien 1	Abnormal	Abnormal	<i>True positive (TP)</i>	98,80%

No. Pasien	Diagnosis Dokter	Prediksi Aplikasi	Status Klasifikasi	Confidence score (%)
Pasien 2	Normal	Normal	<i>True negative (TN)</i>	57,00%
Pasien 3	Normal	Normal	<i>True negative (TN)</i>	75,00%
Pasien 4	Normal	Abnormal	<i>False Positive (FP)</i>	84,80%
Pasien 5	Abnormal	Abnormal	<i>True positive (TP)</i>	98,80%
Pasien 6	Abnormal	Abnormal	<i>True positive (TP)</i>	92,60%
Pasien 7	Abnormal	Abnormal	<i>True positive (TP)</i>	63,70%
Pasien 8	Abnormal	Normal	<i>False Negative (FN)</i>	62,90%
Pasien 9	Normal	Normal	<i>True negative (TN)</i>	89,80%
Pasien 10	Normal	Abnormal	<i>False Positive (FP)</i>	94,50%

Berdasarkan data sampel pada Tabel 4.27, tingkat keyakinan model saat memprediksi kondisi positif (abnormal) cenderung tinggi, baik pada keputusan yang tepat (*True positive*) maupun keputusan yang keliru (*False Positive*). Hal ini mengindikasikan model memiliki sensitivitas visual yang sangat peka terhadap pola abnormalitas pada permukaan serviks, meskipun kepekaan ini sering kali memicu kesalahan prediksi pada kondisi serviks normal. Di sisi lain, pada prediksi negatif (normal), tingkat keyakinan model tergolong lebih rendah. Pola ini terlihat dari satu sampel kasus *False Negative* yang memiliki *confidence score* lebih rendah dibandingkan rata-rata kasus positif.

Temuan kualitatif dari masukan para dokter responder memperjelas penyebab tingginya prediksi positif palsu dan rendahnya spesifisitas lapangan. Pertama, model pada aplikasi masih kesulitan membedakan lesi pra-kanker serviks dengan kondisi peradangan serviks biasa (servisitis non-spesifik) karena kedua kondisi tersebut menampilkan kemerahan dan iritasi visual yang serupa pada citra IVA. Kedua, faktor lingkungan di lapangan seperti kualitas pencahayaan yang tidak konsisten, adanya bayangan, atau pantulan cahaya *flash* kamera ponsel pada permukaan spekulum sering kali memicu derau visual yang mengacaukan proses inferensi model. Rekomendasi perbaikan ke depan berfokus pada standarisasi jarak pengambilan gambar dan penambahan modul filter cahaya pada antarmuka kamera aplikasi.

4.3 Pembahasan

Sub-bab sebelumnya (4.2.1 sampai 4.2.7) telah memaparkan hasil dari setiap eksperimen beserta analisis lokalnya. Bagian ini menyatukan seluruh temuan tersebut ke dalam satu pembahasan menyeluruh yang menjawab rumusan masalah pada BAB 1, membandingkan hasil dengan penelitian terdahulu pada BAB 2, menjelaskan hubungan antar-temuan, serta mengidentifikasi implikasi dan keterbatasan penelitian.

4.3.1 Jawaban terhadap Rumusan Masalah

4.3.1.1 Rumusan Masalah 1: Arsitektur CNN untuk Klasifikasi Citra Tes IVA

Rumusan masalah pertama menanyakan bagaimana merancang arsitektur CNN yang mampu mengklasifikasi citra tes IVA secara akurat. Dari seluruh kombinasi arsitektur dan strategi pra-pemrosesan yang diuji (NB-2 sampai NB-9), kombinasi ResNet-50 dengan pemotongan otomatis OpenCV (NB-5) menghasilkan performa terbaik. Model ini mencatat *Average Precision* sebesar 0,812, AUC sebesar 0,709, dan *Precision@recall* = 0,8 sebesar 0,756 pada dataset uji hold-out. Angka *Precision@recall* ini berhasil melampaui target evaluasi, yaitu skor *baseline* Naive Bayes pada NB-10 yang hanya mencapai 0,742. Akurasi keseluruhan juga naik dari 44,8% (*baseline*) menjadi 75,7%.

Pemilihan arsitektur saja tidak cukup untuk mendapatkan performa optimal. Dua keputusan desain lain juga berperan penting. Pertama, pendekatan *transfer learning* dua fase (membekukan *base model* pada fase ekstraksi fitur, lalu membuka 25% lapisan teratas saat *fine-tuning*) memungkinkan model memanfaatkan pengetahuan visual umum dari ImageNet sambil menyesuaikan fitur tingkat tinggi terhadap pola lesi serviks. Kedua, desain *classification head* yang sama persis untuk ketiga arsitektur memastikan bahwa perbedaan hasil benar-benar berasal dari kemampuan *base model*, bukan dari perbedaan konfigurasi *head*.

4.3.1.2 Rumusan Masalah 2: Pengaruh *Post-Training Quantization* terhadap Efisiensi

Rumusan masalah kedua menanyakan pengaruh PTQ terhadap efisiensi sumber daya di perangkat Android. Kuantisasi INT8 berhasil memperkecil ukuran model ResNet-50 sebesar 75%, dari 92,5 MB menjadi 23,1 MB, dan mempercepat waktu inferensi dari rata-rata 290 milidetik menjadi 110 milidetik pada CPU ponsel. Pola penyusutan 75% ini konsisten di ketiga arsitektur: MobileNetV3-Small turun dari 4,60 MB menjadi 1,16 MB, dan EfficientNet-B0 dari 18,80 MB menjadi 4,70 MB.

Dari sisi performa diagnostik, degradasi akibat kuantisasi terbukti sangat kecil. *Precision@recall* pada ResNet-50 hanya turun 0,001 (dari 0,757 pada Float32 menjadi 0,756 pada INT8). Artinya, kalibrasi berbasis 200 sampel validation set sudah cukup untuk menjaga kualitas prediksi model meskipun representasi bobotnya direduksi dari 32-bit ke 8-bit. Keberhasilan ini juga ditopang oleh konfigurasi fallback ke operasi TFLITE_BUILTINS untuk operasi yang tidak mendukung INT8 penuh (seperti HardSwish pada MobileNetV3), sehingga operasi yang sensitif terhadap presisi tidak dipaksakan melewati jalur integer.

4.3.1.3 Rumusan Masalah 3: Efektivitas Grad-CAM sebagai *Explainable AI*

Rumusan masalah ketiga menanyakan sejauh mana Grad-CAM efektif dalam memvisualisasikan dasar keputusan model. Pada model pemenang (ResNet-50 NB-6), *heatmap* Grad-CAM menunjukkan konsentrasi fokus tepat di atas Sambungan Skuamosa Kolumnar (SSK) dan di sepanjang area lesi *acetowhite*. Area ini memang menjadi acuan

utama dokter saat membaca hasil tes IVA, sehingga Grad-CAM memvalidasi bahwa model mengambil keputusan berdasarkan indikator biologis yang benar.

Yang menarik, peran Grad-CAM dalam penelitian ini ternyata melampaui fungsi awalnya sebagai alat transparansi bagi dokter. Grad-CAM juga berfungsi sebagai alat diagnostik internal bagi peneliti untuk mendeteksi kegagalan pra-pemrosesan. Pada NB-1 (citra mentah), *heatmap* menyala pada ujung spekulum logam, bukan pada lesi, sehingga membuktikan terjadinya *shortcut learning*. Pada NB-3 (CBAM), *heatmap* pecah menjadi titik-titik acak yang tersebar, mengonfirmasi terjadinya *feature collapse*. Pada NB-7 (masking statis), *heatmap* menyala di sepanjang tepi lingkaran hitam dan mengabaikan lesi di tengah gambar. Bukti-bukti visual semacam ini menunjukkan bahwa Grad-CAM berguna sebagai mekanisme *quality assurance* yang melengkapi metrik kuantitatif.

Dari sisi penerimaan pengguna, hasil UAT menunjukkan bahwa dokter spesialis ObGyn menilai fitur *heatmap* bermanfaat sebagai alat *second opinion*, meskipun skor kepercayaan terhadap prediksi AI masih di angka 3,38 dari 5. Kehati-hatian dokter ini wajar dan justru sehat secara klinis, mengingat Grad-CAM memang dirancang untuk mendukung penilaian dokter, bukan menggantikannya.

4.3.2 Posisi Temuan terhadap Penelitian Terdahulu

Bagian ini membandingkan hasil penelitian dengan studi-studi yang telah dibahas pada BAB 2 (Tabel 2.1).

4.3.2.1 Perbandingan dengan Maben et al. (2025)

Maben et al. melaporkan akurasi 92,31% dan sensitivitas 98,24% menggunakan MobileNet-V2 truncated pada perangkat Android. Angka ini jauh di atas hasil penelitian ini (akurasi 75,7%, sensitivitas 89% pada *hold-out test*). Perbedaan ini kemungkinan disebabkan oleh dua hal. Pertama, Maben et al. menggunakan EfficientDet-Lite3 sebagai detektor ROI terpisah sebelum klasifikasi, pendekatan dua tahap yang mirip dengan kombinasi OWL-ViT/YOLOv11 + CNN pada NB-8 dan NB-9 di penelitian ini. Eksperimen NB-8 dan NB-9 memang menunjukkan peningkatan performa dibandingkan heuristik OpenCV, sehingga penggunaan detektor objek terlatih sebagai tahap awal memang memberikan keuntungan yang terukur. Kedua, dataset Maben et al. kemungkinan memiliki distribusi kelas dan kualitas anotasi yang berbeda. Penelitian ini secara sadar mempertahankan variabilitas kualitas citra antar-dataset (IARC WHO dan PRIMER) demi menguji generalisasi model, sebuah keputusan yang menurunkan angka akurasi lab tetapi lebih realistis terhadap kondisi lapangan.

4.3.2.2 Perbandingan dengan Harsono et al. (2022)

Penelitian ini secara langsung mereplikasi dan menantang pendekatan Harsono et al. melalui NB-10. Pada metrik target yang sama ($Precision@recall = 0,8$), model CNN penelitian ini (0,756) berhasil mengungguli pendekatan *handcrafted features* + Naive Bayes (0,742). Selisihnya memang tipis pada metrik target, tetapi perbedaan akurasi keseluruhan sangat besar: 75,7% dibandingkan 44,8%. Perbedaan ini menunjukkan bahwa fitur buatan manual (histogram warna HSV, GLCM, dan sejenisnya) tidak cukup kuat menangani variasi pencahayaan klinis yang ada di dataset gabungan. Proses *Telea Inpainting* untuk menghilangkan pantulan spekular, yang menjadi salah satu komponen utama pipeline Harsono et al., justru menghasilkan area blur buatan yang membingungkan pengklasifikasi

statistik. CNN mampu mengekstraksi fitur biologis langsung dari piksel mentah tanpa memerlukan rekayasa fitur manual yang rentan terhadap variasi kondisi pengambilan citra.

4.3.2.3 Perbandingan dengan Guo et al. (2022) terkait Mekanisme Atensi

Guo et al. melaporkan bahwa integrasi mekanisme *Dual-Attention* (CBAM dan *Coordinate Attention*) pada EfficientNet-B0 berhasil meningkatkan presisi diagnostik hingga 90% melalui fusi fitur multi-view. Hasil NB-4 pada penelitian ini justru berlawanan: CBAM yang disisipkan ke dalam *base model* pralatih menyebabkan *feature collapse* dengan AUC di bawah 0,53. Perbedaan ini kemungkinan terjadi karena Guo et al. merancang arsitektur MCFNet dengan CBAM sebagai bagian yang terintegrasi sejak awal, sehingga seluruh bobot dioptimasi bersama-sama dari nol. Pada penelitian ini, CBAM ditambahkan secara *post-hoc* ke dalam *base model* yang sudah memiliki bobot pralatih ImageNet. Penambahan *post-hoc* ini merusak hierarki fitur yang sudah stabil, sementara perancangan yang terintegrasi sejak awal memungkinkan bobot CBAM dan *base model* saling menyesuaikan selama pelatihan.

4.3.2.4 Perbandingan dengan Viñals et al. (2021) terkait Spesifisitas

Viñals et al. melaporkan sensitivitas 90% dan spesifisitas 87% menggunakan analisis fitur dinamis dari rekaman video IVA. Spesifisitas 87% ini jauh melampaui hasil penelitian ini (56% pada *hold-out test* dan 21,4% pada UAT lapangan). Keunggulan Viñals et al. kemungkinan besar berasal dari penggunaan informasi temporal, yaitu perubahan intensitas warna seiring waktu setelah pemberian asam asetat. Informasi temporal ini memberikan sinyal pembeda tambahan untuk memisahkan reaksi *acetowhite* sejati dari kondisi non-patologis seperti metaplasia skuamosa imatur atau peradangan biasa. Penelitian ini hanya mengandalkan citra statis tunggal, sehingga kehilangan konteks temporal tersebut.

4.3.3 Keterkaitan Antar-Temuan

Seluruh temuan dari sub-bab 4.2.1 sampai 4.2.7 membentuk satu rangkaian kausal yang saling terhubung, mulai dari tahap pra-pemrosesan hingga penerimaan klinis oleh dokter.

4.3.3.1 Tahap 1: Pra-pemrosesan Menentukan Apa yang Dipelajari Model

Eksperimen ablasi pada sub-bab 4.2.1 membuktikan bahwa strategi pra-pemrosesan bukan sekadar langkah persiapan teknis. Pra-pemrosesan menentukan apakah model belajar mengenali pola biologis atau sekadar menghafal artefak gambar. Citra mentah (NB-1) menghasilkan AUC tertinggi (0,798), tetapi analisis Grad-CAM pada sub-bab 4.2.5 memperlihatkan bahwa model menghafal pola spekulum logam, bukan mempelajari lesi kanker. Sementara itu, pemotongan otomatis OpenCV (NB-4 dan NB-5) menurunkan AUC ke 0,709, tetapi *heatmap* Grad-CAM menunjukkan fokus yang benar pada SSK dan *acetowhite*. Penurunan angka AUC ini justru merupakan koreksi, yaitu angka yang lebih rendah mencerminkan tingkat kesulitan yang sesungguhnya dari tugas klasifikasi biologis, bukan kemudahan palsu dari menghafal artefak.

4.3.3.2 Tahap 2: Kualitas Pra-pemrosesan Memengaruhi Pemilihan Arsitektur

Temuan dari sub-bab 4.2.1.6 dan 4.2.1.7 menunjukkan hal yang menarik: arsitektur "terbaik" berubah tergantung pada karakteristik potongan citra. Pemotongan lebar oleh OWL-ViT (NB-8) menguntungkan EfficientNet-B0 (AUC 0,766) karena arsitektur ini memanfaatkan konteks spasial global. Sebaliknya, pemotongan ketat oleh YOLOv11 (NB-9) menguntungkan ResNet-50 (AUC 0,774) yang lebih mampu mengekstraksi fitur tekstur

mendalam tanpa memerlukan konteks lebar. Interaksi antara arsitektur dan pra-pemrosesan ini menjelaskan mengapa ResNet-50 dipilih bersama OpenCV *auto-crop*: pemotongan OpenCV menghasilkan margin moderat yang cukup untuk mempertahankan konteks anatomis sekaligus menyediakan detail tekstur yang cocok untuk ResNet-50.

4.3.3.3 Tahap 3: Kuantisasi Dapat Mempertahankan Hasil Akhir Parameter

Pada sub-bab 4.2.3, kuantisasi INT8 terbukti menyusutkan ukuran model ResNet-50 sebesar 75% dengan penurunan *Precision@recall* yang hanya 0,001. Ini berarti keputusan klinis yang sudah divalidasi oleh Grad-CAM pada tahap sebelumnya tidak terdistorsi oleh proses kompresi model. Dengan ukuran 23,1 MB dan latensi 110 milidetik, model menjadi layak dijalankan pada perangkat Android secara *offline* tanpa koneksi internet.

4.3.3.4 Tahap 4: Grad-CAM Menjembatani Performa Lab dan Kepercayaan Klinis

Heatmap Grad-CAM pada sub-bab 4.2.5 memberikan bukti visual bahwa model pemenang (NB-5) memperhatikan area anatomis yang tepat. Informasi ini menjadi modal penting saat UAT (sub-bab 4.2.7): dokter bisa melihat alasan di balik prediksi model, bukan hanya hasil prediksinya. Meskipun skor kepercayaan terhadap AI masih di angka 3,38 dari 5, *heatmap* Grad-CAM berhasil meyakinkan dokter bahwa aplikasi layak digunakan sebagai alat *second opinion* karena area yang disorot konsisten dengan pengamatan klinis mereka.

4.3.3.5 Tahap 5: Validasi Lapangan Mengungkap Domain Gap

Pada UAT (sub-bab 4.2.7), sensitivitas aplikasi tercatat 83,3%, sedikit di bawah sensitivitas *hold-out test* (89%). Spesifisitas mengalami penurunan yang jauh lebih besar, dari 56% (laboratorium) menjadi 21,4% (lapangan). Rangkaian temuan dari tahap-tahap sebelumnya membantu menjelaskan penurunan ini: model dilatih pada dataset IARC WHO dan PRIMER yang memiliki distribusi patologi tertentu, sementara populasi pasien di RSUP Hasan Sadikin Bandung memiliki proporsi kondisi non-kanker (cervicitis, metaplasia) yang berbeda dan belum terwakili dalam dataset pelatihan. Pra-pemrosesan OpenCV juga belum sepenuhnya mampu menangani variasi pencahayaan dan sudut kamera di lingkungan klinis yang sesungguhnya.

4.3.4 Implikasi Praktis dan Klinis

Kesenjangan antara performa laboratorium dan performa lapangan memiliki konsekuensi langsung terhadap strategi penerapan aplikasi CervicalAI.

4.3.4.1 Sensitivitas Lapangan

Sensitivitas lapangan sebesar 83,3% berarti dari setiap 6 kasus abnormal yang sesungguhnya, model berhasil mendeteksi 5 kasus dan melewatkan 1 kasus. Untuk konteks skrining lapis pertama, angka ini masih dapat diterima. Tes IVA konvensional oleh tenaga medis terlatih sendiri memiliki sensitivitas yang bervariasi antara 50-96% tergantung pada pengalaman pemeriksa (Bradford & Goodman, 2013). Dengan kata lain, model CNN ini dapat berperan sebagai penyetara kualitas yang memberikan konsistensi diagnostik terlepas dari tingkat pengalaman pemeriksa di lapangan.

4.3.4.2 Spesifisitas Lapangan dan Alarm Palsu

Spesifisitas lapangan sebesar 21,4% menjadi tantangan utama. Dari setiap 14 kasus normal yang diperiksa, model memberikan alarm palsu pada 11 kasus. Dalam konteks klinis, alarm palsu bukan tanpa akibat: pasien yang menerima hasil positif palsu akan dirujuk untuk

kolposkopi atau biopsi yang sebenarnya tidak diperlukan, yang menimbulkan beban psikologis dan biaya tambahan.

Meskipun demikian, dalam sistem skrining dua tahap (skrining awal kemudian konfirmasi), alarm palsu lebih bisa ditoleransi dibandingkan kasus yang terlewat (false negative). Pasien yang dirujuk berlebihan masih akan melewati tahap konfirmasi yang bisa membatalkan alarm palsu tersebut, sedangkan pasien yang terlewat kehilangan kesempatan deteksi dini.

4.3.4.3 Penyebab Domain Gap

Gap spesifisitas antara laboratorium (56%) dan lapangan (21,4%) menunjukkan adanya domain shift yang cukup besar. Beberapa faktor yang kemungkinan berkontribusi, yaitu (1) pencahayaan di ruang pemeriksaan RSUP Hasan Sadikin berbeda dari kondisi pengambilan dataset pelatihan, (2) proporsi cervicitis dan kondisi inflamasi non-kanker pada populasi pasien rumah sakit rujukan lebih tinggi dibandingkan distribusi kelas dalam dataset pelatihan, dan (3) variasi perangkat kamera dan sudut pengambilan yang lebih besar di lapangan dibandingkan kondisi dataset terkontrol.

Temuan ini memperkuat argumen bahwa model skrining medis perlu melewati siklus validasi berkelanjutan dengan data dari lokasi penerapan target sebelum dapat digunakan secara mandiri.

4.3.4.4 Respons Pengguna dan Posisi Alat

Meskipun spesifisitas masih rendah, respons dokter pada UAT memberikan sinyal yang positif. Skor kemudahan penggunaan (4,15 dari 5), kecepatan (4,33 dari 5), dan kesesuaian alur kerja (4,30 dari 5) menunjukkan bahwa hambatan adopsi utama bukan pada aspek teknis antarmuka, melainkan pada keandalan diagnostik model itu sendiri. Dokter secara eksplisit menyatakan bahwa kemampuan CervicalAI menyoroti area mencurigakan melalui *heatmap* menjadikannya alat *second opinion* yang berguna, selama keputusan akhir tetap di tangan tenaga medis. Posisi ini sesuai dengan batasan masalah yang ditetapkan pada BAB 1, yaitu bahwa sistem berfungsi sebagai alat bantu skrining (*decision support*), bukan pengganti diagnosis klinis akhir.

4.3.5 Keterbatasan Penelitian

Beberapa keterbatasan penelitian perlu disampaikan untuk memberikan konteks yang tepat bagi interpretasi hasil dan menjadi landasan bagi penelitian selanjutnya.

4.3.5.1 Ukuran Sampel UAT yang Terbatas

Pengujian lapangan hanya melibatkan 20 kasus pasien dengan 4 dokter penguji. Jumlah ini belum cukup untuk menghasilkan estimasi sensitivitas dan spesifisitas lapangan yang stabil secara statistik. Dengan hanya 6 kasus positif dan 14 kasus negatif, interval kepercayaan untuk kedua metrik tersebut sangat lebar. Validasi klinis yang lebih besar dengan ratusan kasus di beberapa lokasi akan diperlukan sebelum kesimpulan tentang performa lapangan bisa dianggap final.

4.3.5.2 Klasifikasi Biner Tanpa Kelas Patologis Lain

Dataset pelatihan hanya mencakup dua kelas (*Normal vs Positive*). Pada praktik klinis, kondisi serviks jauh lebih beragam, yaitu *cervicitis*, *metaplasia skuamosa imatur*, *polip endoservikal*, dan berbagai tingkat lesi intraepitel (LSIL, HSIL). Karena kelas-kelas

patologis non-kanker ini tidak ada dalam dataset pelatihan, model memperlakukan semua kondisi non-normal sebagai "*Positive*". Hal ini secara langsung menjelaskan rendahnya spesifisitas, terutama di lapangan di mana *prevalensi cervicitis* lebih tinggi.

4.3.5.3 Keterbatasan Algoritma Pemotongan Otomatis OpenCV

Algoritma pemotongan otomatis OpenCV bergantung pada keberhasilan deteksi kontur menggunakan pengembangan Otsu. Pada kondisi pencahayaan redup, bayangan yang kuat, atau sudut kamera yang terlalu miring, algoritma ini bisa gagal menemukan kontur yang tepat (sebagaimana sudah disebutkan pada sub-bab 4.2.1.3). Kegagalan pemotongan menyebabkan artefak non-biologis tetap masuk ke dalam *input* model, yang berpotensi menghasilkan prediksi yang tidak valid. *Pipeline* saat ini belum memiliki mekanisme untuk mendeteksi kegagalan pemotongan secara otomatis.

4.3.5.4 Pengujian pada Perangkat Tunggal

Evaluasi inferensi dilakukan hanya pada satu perangkat uji dengan konfigurasi *single-thread* CPU. Variasi performa pada perangkat Android yang bermacam-macam (chipset berbeda, versi OS berbeda, memori yang tersedia berbeda) belum dipetakan secara sistematis. Latensi 110 milidetik yang dilaporkan bisa berbeda secara signifikan pada ponsel kelas bawah yang justru menjadi target penerapan utama di fasilitas kesehatan primer.

4.3.5.5 Tidak Mengeksplorasi *Quantization-Aware Training*

Penelitian ini hanya menerapkan *Post-Training Quantization* (PTQ) dan tidak mengeksplorasi *Quantization-Aware Training* (QAT) yang menyimulasikan presisi rendah selama pelatihan. Meskipun PTQ sudah menunjukkan degradasi performa yang sangat kecil pada penelitian ini, QAT berpotensi memberikan peningkatan lebih lanjut terutama pada arsitektur yang lebih kecil seperti MobileNetV3-Small, yang lebih sensitif terhadap pengurangan presisi bobot.

4.3.5.6 Aproksimasi Grad-CAM pada Aplikasi Android

Grad-CAM pada aplikasi Android menggunakan pendekatan berbasis bobot *Dense layer* yang diekspor terpisah, bukan komputasi Grad-CAM penuh berbasis gradien. Pendekatan ini merupakan aproksimasi yang menghasilkan visualisasi representatif tetapi tidak identik dengan Grad-CAM penuh yang dijalankan di lingkungan TensorFlow/Keras. Perbedaan antara kedua pendekatan ini belum diukur secara formal dalam penelitian ini.

[Halaman ini sengaja dikosongkan]

BAB 5 KESIMPULAN

5.1 Kesimpulan

1. *Transfer learning* dengan arsitektur ResNet-50 dikombinasikan pemotongan otomatis berbasis kontur OpenCV tanpa augmentasi data (NB-6) berhasil mengoptimalkan klasifikasi citra IVA, dengan ROC-AUC 0,709, F1-score 81,43%, dan akurasi meningkat dari 44,8% menjadi 75,7% (format INT8). Model ini mengungguli *baseline* Naive Bayes ($Precision@recall=0,8$ sebesar 0,742–0,757 dengan sensitivitas hanya 7,81%). Pada validasi klinis terhadap 20 pasien, model mempertahankan sensitivitas lapangan 83,3%, menjadikannya andal sebagai skrining lapis pertama untuk meminimalkan *False Negative*, meskipun spesifisitas masih rendah (21,4%) akibat alarm palsu pada *cervicitis*.
2. *Post-Training Quantization* INT8 secara signifikan meningkatkan efisiensi model: ukuran model menyusut 75% (92,5 MB menjadi 23,1 MB) dan waktu inferensi CPU rata-rata hanya 110 ms per citra, mendukung diagnosis yang cepat dan responsif di perangkat Android.
3. Grad-CAM efektif memverifikasi validitas biologis keputusan model dan mendeteksi efek *Clever Hans* (*shortcut learning*). ResNet-50 menunjukkan atensi paling presisi pada Sambungan Skuamosa Kolumnar (SSK) dan lesi *acetowhite*, sedangkan EfficientNet-B0 menghasilkan *heatmap* yang melebar dan MobileNetV3-Small gagal fokus.
4. Prototipe aplikasi Android CervicalAI berhasil diimplementasikan untuk klasifikasi citra tes IVA secara *offline* 100%, menjaga kerahasiaan data pasien di perangkat. UAT oleh dokter spesialis Obgin RSUP Hasan Sadikin Bandung menunjukkan kepuasan tinggi: kemudahan penggunaan 4,15/5, kecepatan 4,33/5, dan kesesuaian alur kerja 4,30/5.

5.2 Saran

1. Dataset yang digunakan dalam penelitian ini terbatas pada citra IVA dari dua sumber, yaitu Poli Onkologi RSUP Hasan Sadikin dan dataset berlisensi IARC/WHO, sehingga terdapat *domain gap* yang signifikan antara kondisi pengambilan gambar di laboratorium dan lapangan. Penelitian selanjutnya disarankan untuk memperluas cakupan pengumpulan data dari fasilitas kesehatan primer seperti Puskesmas di berbagai daerah dengan kondisi pencahayaan, perangkat, dan demografis pasien yang lebih beragam, guna meningkatkan generalisasi model pada kondisi klinis nyata.
2. Teknik optimasi yang diterapkan pada penelitian ini dibatasi pada *Post-Training Quantization* (PTQ). Penelitian selanjutnya disarankan untuk mengeksplorasi teknik optimasi lain seperti *Quantization-Aware Training* (QAT) maupun *structured pruning*, serta membandingkan *trade-off* antara ukuran model, latensi inferensi, dan performa diagnostik secara sistematis untuk menemukan konfigurasi optimal pada perangkat *mobile*.

3. *Output* sistem saat ini dibatasi pada klasifikasi biner (*Normal/Positive*) dengan visualisasi Grad-CAM. Penelitian selanjutnya disarankan untuk mengembangkan klasifikasi multi-kelas yang mencakup kategori seperti *Cervicitis*, LSIL, dan HSIL guna meningkatkan spesifisitas klinis model.
4. Sistem yang dikembangkan berfungsi sebagai alat bantu skrining dan bukan pengganti diagnosis klinis akhir. Untuk memvalidasi efektivitas sistem sebagai *decision support tool* secara nyata, penelitian selanjutnya disarankan melakukan uji klinis prospektif multi-senter yang melibatkan lebih banyak tenaga medis dan kasus, sehingga nilai sensitivitas, spesifisitas, dan penerimaan klinis dapat diukur secara lebih representatif.

DAFTAR PUSTAKA

- Asgary, R., Staderini, N., Mthethwa-Hleta, S., Saavedra, P. A. L., Abrego, L. G., Rusch, B., Luce, T. M., Pasipamire, L. R., Ndlangamandla, M., Beideck, E., & Kerschberger, B. (2020). Evaluating smartphone strategies for reliability, reproducibility, and quality of VIA for cervical cancer screening in the Shiselweni region of Eswatini: A cohort study. *PLoS Medicine*, 17(11). <https://doi.org/10.1371/journal.pmed.1003378>
- Bae, J. K., Roh, H. J., You, J. S., Kim, K., Ahn, Y., Askaruly, S., Park, K., Yang, H., Jang, G. J., Moon, K. H., & Jung, W. (2020). Quantitative screening of cervical cancers for low-resource settings: Pilot study of smartphone-based endoscopic visual inspection after acetic acid using machine learning techniques. *JMIR MHealth and UHealth*, 8(3), e16467. <https://doi.org/10.2196/16467>
- Bradford, L., & Goodman, A. (2013). Cervical cancer screening and prevention in low-resource settings. *Clinical Obstetrics and Gynecology*, 56(1), 76–87. <https://doi.org/10.1097/GRF.0b013e31828237ac>
- Bray, F., Laversanne, M., Sung, H., Ferlay, J., Siegel, R. L., Soerjomataram, I., & Jemal, A. (2024). Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, 74(3), 229–263. <https://doi.org/10.3322/caac.21834>
- Guo, Y., Wang, Y., Yang, H., Zhang, J., & Sun, Q. (2022). Dual-attention EfficientNet based on multi-view feature fusion for cervical squamous intraepithelial lesions diagnosis. *Biocybernetics and Biomedical Engineering*, 42(2), 529–542. <https://doi.org/10.1016/J.BBE.2022.02.009>
- Harsono, A. B., Susiarno, H., Suardi, D., Owen, L., Fauzi, H., Kireina, J., Wahid, R. A., Carolina, J. S., Mantilidewi, K. I., & Hidayat, Y. M. (2022). Cervical pre-cancerous lesion detection: development of smartphone-based VIA application using artificial intelligence. *BMC Research Notes*, 15(1). <https://doi.org/10.1186/s13104-022-06250-6>
- Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). *Searching for MobileNetV3*. <http://arxiv.org/abs/1905.02244>
- International Agency for Research on Cancer (IARC). (2021a). *IARC Cervical Cancer Image Bank, IARC Colposcopy Image Bank*. World Health Organization. Accessed: 2026-06-26. <https://screening.iarc.fr/cervicalimagebank.php>
- International Agency for Research on Cancer (IARC). (2021b). *IARC Cervical Cancer Image Bank, Image Bank Visual Inspection with Acetic Acid (VIA)*. World Health Organization. Accessed: 2026-07-01. <https://screening.iarc.fr/cervicalimagebank.php>
- Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., & Kalenichenko, D. (n.d.). *Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference*.

- Li, P., Wang, X., Liu, P., Xu, T., Sun, P., Dong, B., & Xue, H. (2022). Cervical Lesion Classification Method Based on Cross-Validation Decision Fusion Method of Vision Transformer and DenseNet. *Journal of Healthcare Engineering*, 2022. <https://doi.org/10.1155/2022/3241422>
- Liao, J., Guo, L., Jiang, L., Yu, C., Liang, W., Li, K., & Pop, F. (2025). A machine learning-based feature extraction method for image classification using ResNet architecture. *Digital Signal Processing: A Review Journal*, 160. <https://doi.org/10.1016/j.dsp.2025.105036>
- Maben, L. M., Prasad, K., Guruvare, S., Kudva, V., & Siddalingaswamy, P. C. (2025). *Automated Cervical Cancer Detection through Visual Inspection with Acetic Acid in Resource-Poor Settings with Lightweight Deep Learning Models Deployed on an Android Device*. <https://arxiv.org/pdf/2508.13253>
- Matthias Minderer Alexey Gritsenko, A. S. M. N. D. W. A. D. A. M. A. A. M. D. Z. S. X. W. X. Z. T. K. N. H. (2022). Simple Open-Vocabulary Object Detection with Vision Transformers. *ArXiv Preprint ArXiv:2205.06230*.
- Nigam, S., Jain, R., Singh, V. K., Marwaha, S., Arora, A., & Jain, S. (2024). EfficientNet architecture and attention mechanism-based wheat disease identification model. *Procedia Computer Science*, 235, 383–393. <https://doi.org/10.1016/j.procs.2024.04.038>
- Peng, G., Dong, H., Liang, T., Li, L., & Liu, J. (2021). Diagnosis of cervical precancerous lesions based on multimodal feature changes. *Computers in Biology and Medicine*, 130, 104209. <https://doi.org/10.1016/J.COMPBIOMED.2021.104209>
- Purwono, Ma'arif, A., Rahmaniar, W., Fathurrahman, H. I. K., Frisky, A. Z. K., & Haq, Q. M. U. (2022). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control Systems*, 2(4), 739–748. <https://doi.org/10.31763/ijrcs.v2i4.888>
- Räz, T., Pahud De Mortanges, A., & Reyes, M. (2025). Explainable AI in medicine: challenges of integrating XAI into the future clinical routine. In *Frontiers in Radiology* (Vol. 5). Frontiers Media SA. <https://doi.org/10.3389/fradi.2025.1627169>
- Rezabeyk, E. M., Beigzad, S., Hamzavi, Y., Bagheritabar, M., & Mirikhoozani, S. S. (2024). *Saliency Assisted Quantization for Neural Networks*. <https://arxiv.org/pdf/2411.05858v1>
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision*, 128(2), 336–359. <https://doi.org/10.1007/s11263-019-01228-7>
- SmartClover. (2026). *Cervical Cancer Lesion Classification CNN*. SmartClover. https://huggingface.co/smartclover/cerviguard_lesion

- Song, B., & Liang, R. (2025). Integrating artificial intelligence with smartphone-based imaging for cancer detection in vivo. *Biosensors and Bioelectronics*, 271. <https://doi.org/10.1016/j.bios.2024.116982>
- Sritharan, N., Gnanavel, N., Inparaj, P., Meedeniya, D., & Yogarajah, P. (2025). Explainable Artificial Intelligence Driven Segmentation for Cervical Cancer Screening. *IEEE Access*, 13, 71306–71322. <https://doi.org/10.1109/ACCESS.2025.3561178>
- Viñals, R., Vassilakos, P., Rad, M. S., Undurraga, M., Petignat, P., & Thiran, J. P. (2021). Using Dynamic Features for Automatic Cervical Precancer Detection. *Diagnostics* 2021, Vol. 11, Page 716, 11(4), 716. <https://doi.org/10.3390/DIAGNOSTICS11040716>
- Zhao, L., & Wang, L. (2022). A new lightweight network based on MobileNetV3. *KSII Transactions on Internet and Information Systems*, 16(1), 1–15. <https://doi.org/10.3837/tiis.2022.01.001>

[Halaman ini sengaja dikosongkan]

LAMPIRAN

Tabel Hasil Seluruh Percobaan *Preprocessing* NB1-NB9 untuk Parameter *Sensitivity*, *Specificity*, dan *Precision at Recall* (0,8)

Percobaan	Model	Format	Sensitivity	Specificity	Prec@Rec(0.8)
NB-8	EfficientNet-B0	Float32	0.531250	0.953488	0.722222
NB-8	EfficientNet-B0	INT8	0.406250	0.930233	0.684211
NB-8	MobileNetV3Small	Float32	0.859375	0.302326	0.658537
NB-8	MobileNetV3Small	INT8	0.515625	0.627907	0.598131
NB-8	ResNet50	Float32	0.656250	0.697674	0.688312
NB-8	ResNet50	INT8	0.625000	0.697674	0.659341
NB-9	EfficientNet-B0	Float32	0.515625	0.860465	0.641975
NB-9	EfficientNet-B0	INT8	0.593750	0.697674	0.693333
NB-9	MobileNetV3Small	Float32	0.468750	0.581395	0.598131
NB-9	MobileNetV3Small	INT8	0.671875	0.395349	0.598131
NB-9	ResNet50	Float32	0.546875	0.790698	0.706667
NB-9	ResNet50	INT8	0.562500	0.813953	0.706667
NB-2	EfficientNet-B0	Float32	0.531250	0.953488	0.722222

Percobaan	Model	Format	Sensitivity	Specificity	Prec@Rec(0.8)
NB-2	EfficientNet-B0	INT8	0.406250	0.930233	0.684211
NB-2	MobileNetV3Small	Float32	0.859375	0.302326	0.658537
NB-2	MobileNetV3Small	INT8	0.515625	0.627907	0.598131
NB-2	ResNet50	Float32	0.656250	0.697674	0.688312
NB-2	ResNet50	INT8	0.625000	0.697674	0.659341
NB-3	EfficientNet B0_CBAM16	Float32	0.796875	0.558140	0.732394
NB-3	EfficientNet B0_CBAM16	INT8	0.531250	0.395349	0.608696
NB-3	EfficientNet B0_CBAM8	Float32	0.593750	0.837209	0.702703
NB-3	EfficientNet B0_CBAM8	INT8	0.406250	0.767442	0.646341
NB-3	MobileNetV3Small_CBAM16	Float32	0.296875	0.697674	0.598131
NB-3	MobileNetV3Small_CBAM16	INT8	0.140625	0.837209	0.598131
NB-3	MobileNetV3Small_CBAM8	Float32	0.484375	0.604651	0.598131

Percobaan	Model	Format	Sensitivity	Specificity	Prec@Rec(0.8)
NB-3	MobileNetV3Small_CBAM8	INT8	0.218750	0.790698	0.622222
NB-3	ResNet50_CBAM16	Float32	0.546875	0.837209	0.712500
NB-3	ResNet50_CBAM16	INT8	0.296875	0.930233	0.707317
NB-3	ResNet50_CBAM8	Float32	0.656250	0.744186	0.722222
NB-3	ResNet50_CBAM8	INT8	0.328125	0.906977	0.725000
NB-4	EfficientNet-B0	Float32	0.937500	0.093023	0.612245
NB-4	EfficientNet-B0	INT8	0.406250	0.767442	0.598131
NB-4	MobileNetV3Small	Float32	0.296875	0.813953	0.626506
NB-4	MobileNetV3Small	INT8	0.796875	0.069767	0.598131
NB-4	ResNet50	Float32	0.609375	0.441860	0.615385
NB-4	ResNet50	INT8	0.578125	0.511628	0.650000
NB-5	EfficientNet-B0	Float32	0.671875	0.511628	0.638554
NB-5	EfficientNet-B0	INT8	0.609375	0.511628	0.634409
NB-5	MobileNetV3Small	Float32	0.250000	0.883721	0.635294

Percobaan	Model	Format	Sensitivity	Specificity	Prec@Rec(0.8)
NB-5	MobileNetV3Small	INT8	0.843750	0.069767	0.598131
NB-5	ResNet50	Float32	0.859375	0.581395	0.757143
NB-5	ResNet50	INT8	0.890625	0.558140	0.756757
NB-1	EfficientNet-B0	Float32	0.781250	0.674419	0.787879
NB-1	EfficientNet-B0	INT8	0.718750	0.697674	0.764706
NB-1	MobileNetV3Small	Float32	0.890625	0.279070	0.655556
NB-1	MobileNetV3Small	INT8	0.796875	0.395349	0.654321
NB-1	ResNet50	Float32	0.750000	0.604651	0.693333
NB-1	ResNet50	INT8	0.765625	0.604651	0.691358
NB-6	EfficientNet-B0	Float32	0.734375	0.232558	N/A
NB-6	EfficientNet-B0	INT8	0.546875	0.488372	N/A
NB-6	MobileNetV3Small	Float32	0.281250	0.813953	N/A
NB-6	MobileNetV3Small	INT8	0.812500	0.116279	N/A
NB-6	ResNet50	Float32	0.531250	0.651163	N/A
NB-6	ResNet50	INT8	0.578125	0.604651	N/A

Percobaan	Model	Format	Sensitivity	Specificity	Prec@Rec(0.8)
NB-7	EfficientNet-B0	Float32	0.609375	0.418605	0.629213
NB-7	EfficientNet-B0	INT8	0.421875	0.720930	0.625000
NB-7	MobileNetV3Small	Float32	0.625000	0.465116	0.617021
NB-7	MobileNetV3Small	INT8	0.765625	0.139535	0.598131
NB-7	ResNet50	Float32	0.671875	0.348837	0.626374
NB-7	ResNet50	INT8	0.906250	0.186047	0.633333
NB-10	Naive Bayes	N/A	0.078125	1.000000	0.742857
NB-10	Extra Trees Classifier	N/A	0.734375	0.488372	0.675325
NB-10	Ada Boost Classifier	N/A	0.906250	0.348837	0.674419
NB-10	Light Gradient Boosting Machine	N/A	0.750000	0.511628	0.662500
NB-10	Gradient Boosting Classifier	N/A	0.734375	0.441860	0.654321
NB-10	K Neighbors Classifier	N/A	0.640625	0.488372	0.629213
NB-10	Random Forest Classifier	N/A	0.718750	0.418605	0.619048

Percobaan	Model	Format	Sensitivity	Specificity	Prec@Rec(0.8)
NB-10	Logistic Regression	N/A	0.750000	0.302326	0.617978
NB-10	Decision Tree Classifier	N/A	0.453125	0.534884	0.598131

Tabel Hasil Seluruh Percobaan *Preprocessing* NB1-NB9 untuk Parameter AUC, Size, dan Latency

Percobaan	Model	Format	AUC	Size (MB)	Latency (ms)
NB-8	EfficientNet-B0	Float32	0.787427	N/A	N/A
NB-8	EfficientNet-B0	INT8	NaN	4.70	81.58
NB-8	MobileNetV3Small	Float32	0.590116	N/A	N/A
NB-8	MobileNetV3Small	INT8	NaN	1.16	12.45
NB-8	ResNet50	Float32	0.724201	N/A	N/A
NB-8	ResNet50	INT8	NaN	23.12	122.65
NB-9	EfficientNet-B0	Float32	0.701672	N/A	N/A
NB-9	EfficientNet-B0	INT8	NaN	4.70	85.54
NB-9	MobileNetV3Small	Float32	0.526526	N/A	N/A
NB-9	MobileNetV3Small	INT8	NaN	1.16	13.72
NB-9	ResNet50	Float32	0.752907	N/A	N/A
NB-9	ResNet50	INT8	NaN	23.12	122.95
NB-2	EfficientNet-B0	Float32	0.787427	N/A	N/A

Percobaan	Model	Format	AUC	Size (MB)	Latency (ms)
NB-2	EfficientNet-B0	INT8	NaN	4.70	82.61
NB-2	MobileNetV3Small	Float32	0.590116	N/A	N/A
NB-2	MobileNetV3Small	INT8	NaN	1.16	12.73
NB-2	ResNet50	Float32	0.724201	N/A	N/A
NB-2	ResNet50	INT8	NaN	23.12	116.68
NB-3	EfficientNetB0_CBAM16	Float32	0.771076	N/A	N/A
NB-3	EfficientNetB0_CBAM16	INT8	NaN	4.99	85.13
NB-3	EfficientNetB0_CBAM8	Float32	0.759811	N/A	N/A
NB-3	EfficientNetB0_CBAM8	INT8	NaN	5.18	85.57
NB-3	MobileNetV3Small_CBAM16	Float32	0.450581	N/A	N/A
NB-3	MobileNetV3Small_CBAM16	INT8	NaN	1.23	14.16
NB-3	MobileNetV3Small_CBAM8	Float32	0.531250	N/A	N/A
NB-3	MobileNetV3Small_CBAM8	INT8	NaN	1.27	14.14
NB-3	ResNet50_CBAM16	Float32	0.750000	N/A	N/A
NB-3	ResNet50_CBAM16	INT8	NaN	23.72	120.09

Percobaan	Model	Format	AUC	Size (MB)	Latency (ms)
NB-3	ResNet50_CBAM8	Float32	0.772166	N/A	N/A
NB-3	ResNet50_CBAM8	INT8	NaN	24.22	121.24
NB-4	EfficientNet-B0	Float32	0.542515	N/A	N/A
NB-4	EfficientNet-B0	INT8	NaN	4.70	89.23
NB-4	MobileNetV3Small	Float32	0.589753	N/A	N/A
NB-4	MobileNetV3Small	INT8	NaN	1.16	13.84
NB-4	ResNet50	Float32	0.574855	N/A	N/A
NB-4	ResNet50	INT8	NaN	23.12	125.79
NB-5	EfficientNet-B0	Float32	0.690407	N/A	N/A
NB-5	EfficientNet-B0	INT8	NaN	4.70	83.51
NB-5	MobileNetV3Small	Float32	0.594840	N/A	N/A
NB-5	MobileNetV3Small	INT8	NaN	1.16	12.79
NB-5	ResNet50	Float32	0.759811	N/A	N/A
NB-5	ResNet50	INT8	NaN	23.12	110.84
NB-1	EfficientNet-B0	Float32	0.811410	N/A	N/A
NB-1	EfficientNet-B0	INT8	NaN	4.70	81.79
NB-1	MobileNetV3Small	Float32	0.593023	N/A	N/A
NB-1	MobileNetV3Small	INT8	NaN	1.16	12.60
NB-1	ResNet50	Float32	0.753270	N/A	N/A
NB-1	ResNet50	INT8	NaN	23.12	111.53

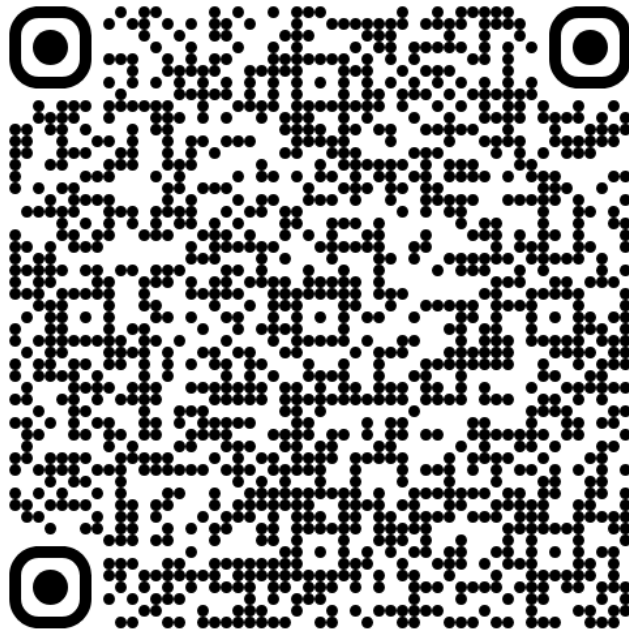
Percobaan	Model	Format	AUC	Size (MB)	Latency (ms)
NB-6	EfficientNet-B0	Float32	0.498728	N/A	N/A
NB-6	EfficientNet-B0	INT8	NaN	4.70	81.88
NB-6	MobileNetV3Small	Float32	0.550509	N/A	N/A
NB-6	MobileNetV3Small	INT8	NaN	1.16	12.16
NB-6	ResNet50	Float32	0.664426	N/A	N/A
NB-6	ResNet50	INT8	NaN	23.12	108.05
NB-7	EfficientNet-B0	Float32	0.535610	N/A	N/A
NB-7	EfficientNet-B0	INT8	NaN	4.70	82.54
NB-7	MobileNetV3Small	Float32	0.595203	N/A	N/A
NB-7	MobileNetV3Small	INT8	NaN	1.16	13.13
NB-7	ResNet50	Float32	0.544695	N/A	N/A
NB-7	ResNet50	INT8	NaN	23.12	120.15
NB-10	Naive Bayes	N/A	0.747093	N/A	N/A
NB-10	Extra Trees Classifier	N/A	0.665698	N/A	N/A
NB-10	Ada Boost Classifier	N/A	0.707485	N/A	N/A
NB-10	Light Gradient Boosting Machine	N/A	0.628634	N/A	N/A
NB-10	Gradient Boosting Classifier	N/A	0.676962	N/A	N/A
NB-10	K Neighbors Classifier	N/A	0.590661	N/A	N/A

Percobaan	Model	Format	AUC	Size (MB)	Latency (ms)
NB-10	Random Forest Classifier	N/A	0.629542	N/A	N/A
NB-10	Logistic Regression	N/A	0.631904	N/A	N/A
NB-10	Decision Tree Classifier	N/A	0.494004	N/A	N/A

Bukti Kegiatan UAT



Laporan Hasil UAT



[Halaman ini sengaja dikosongkan]

BIOGRAFI PENULIS



Naufan Zaki Luqmanulhakim dilahirkan di Kota Bandung pada tanggal 2 Mei 2004 dan merupakan anak pertama dari tiga bersaudara. Penulis menempuh pendidikan dasar di SDIT Auliya, melanjutkan ke jenjang menengah pertama di SMPIT Auliya, dan kemudian menyelesaikan pendidikan menengah atas di SMA Islam Al Azhar Bumi Serpong Damai. Setelah lulus dari SMA pada tahun 2022, Penulis diterima di Departemen Teknologi Informasi, Fakultas Elektro dan Informatika Cerdas (FTEIC) Institut Teknologi Sepuluh Nopember (ITS) pada tahun 2022 dengan NRP 5027221065.

Selama masa studi di Teknologi Informasi, Penulis aktif mengikuti berbagai kegiatan akademik maupun organisasi. Dalam bidang akademik, Penulis dipercaya menjadi asisten dosen untuk mata kuliah Pemrograman Web, Smart City, dan Struktur Data di Laboratorium *Smart City and Cyber Security* (KCKS).

Di luar kegiatan akademik, Penulis juga aktif dalam berbagai organisasi kemahasiswaan, antara lain Himpunan Mahasiswa Teknologi Informasi ITS (HMIT ITS), di mana Penulis menjabat sebagai Kepala Biro Departemen Media dan Informasi.

Penulis memiliki ketertarikan di bidang *Computer Vision* dan penerapan teknologi dalam kesehatan. Penelitian ini disusun sebagai bentuk kontribusi penulis dalam mendalami dan mengembangkan teknologi di bidang-bidang tersebut, serta sebagai wujud dedikasi penulis terhadap kemajuan teknologi informasi di Indonesia.