



Universitas Nusantara PGRI Kediri

UPT. PERPUSTAKAAN, PUBLIKASI DAN INOVASI

Alamat: Kampus 1, Jl. KH. Ahmad Dahlan No.76 Kota Kediri 64112
Telp. (0354) 771576, (0354) 771503, (0354) 771495, Fax. (0354) 771576
Website: <http://ppi.unpkediri.ac.id/> Email: perpustakaan@unpkediri.ac.id

SURAT KETERANGAN BEBAS SIMILARITY

Ketua UPT Perpustakaan, Publikasi dan Inovasi Universitas Nusantara PGRI Kediri menerangkan bahwa mahasiswa dengan identitas berikut:

Nama Mahasiswa : Raya Osgibran
NPM : 2213020139
Program Studi : S1-Teknik Informatika

Judul Karya Ilmiah:

"PENGEMBANGAN SISTEM DETEKSI DINI KANKER KULIT MELANOMA BERBASIS ANDROID
MENGUNAKAN ARSITEKTUR MOBILENETV3"

Dinyatakan sudah memenuhi syarat batas maksimal 30% *similarity* sesuai dengan ketentuan yang berlaku pada setiap subbab naskah Laporan **Tugas Akhir/Skripsi/Tesis** yang disusun.
Demikian Surat Keterangan ini kami berikan untuk dapat dipergunakan seperlunya.

Kediri, 22 Juni 2026
Ka UPT PPI,



Dr. Abdul Aziz Hunaifi, M.A



Turnitin LLCC

Skripsi cek similiarity raya osgibran.docx

TI-06

Document Details

Submission ID

trn:oid::2945:394608897

Submission Date

Jun 22, 2026, 11:28 AM GMT+7

Download Date

Jun 22, 2026, 12:01 PM GMT+7

File Name

Skripsi cek similiarity raya osgibran.docx.pdf

File Size

5.1 MB

77 Pages

19,516 Words

123,495 Characters

15% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Top Sources

- 6%  Internet sources
- 4%  Publications
- 13%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Top Sources

6%	Internet sources
4%	Publications
13%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Student papers	Universitas 17 Agustus 1945 Semarang on 2026-05-25	<1%
2	Student papers	Abdullah Gul University on 2025-11-20	<1%
3	Student papers	Universitas Tarumanagara on 2025-12-24	<1%
4	Internet	kc.umn.ac.id	<1%
5	Student papers	FAKULTAS ILMU KOMPUTER on 2026-01-12	<1%
6	Internet	digilib.unila.ac.id	<1%
7	Internet	www.mdpi.com	<1%
8	Internet	docplayer.info	<1%
9	Student papers	Universitas Tarumanagara on 2025-12-19	<1%
10	Internet	eprints.undip.ac.id	<1%
11	Student papers	UPN Veteran Yogyakarta on 2026-05-29	<1%

12	Student papers	Sriwijaya University on 2025-12-18	<1%
13	Student papers	UPN Veteran Yogyakarta on 2026-02-13	<1%
14	Publication	Sigit Januarto, Aang Alim Murtopo, Zaenul Arif. "Klasifikasi Status Stunting Balita ...	<1%
15	Student papers	Universitas Pancasila on 2026-03-04	<1%
16	Publication	Desi Permata Sari, Bambang Irawan. "Implementasi CNN Mobilenetv2 untuk Klas...	<1%
17	Internet	digilib.uin-suka.ac.id	<1%
18	Student papers	fpptijateng on 2023-09-20	<1%
19	Student papers	Universitas Pamulang on 2026-06-22	<1%
20	Student papers	Universitas Pamulang on 2026-05-04	<1%
21	Student papers	Universitas Bengkulu on 2025-08-28	<1%
22	Internet	widuri.raharja.info	<1%
23	Publication	Andreas Rezeki Zai, Bambang Suhardi, Surya Tri Nowo, Rika Rosnelly, Adil Setiaw...	<1%
24	Student papers	Telkom University on 2026-06-17	<1%
25	Internet	ejournal.ust.ac.id	<1%

26	Internet	www.scribd.com	<1%
27	Student papers	UPN Veteran Yogyakarta on 2025-10-01	<1%
28	Student papers	UPN Veteran Yogyakarta on 2026-05-04	<1%
29	Internet	123dok.com	<1%
30	Publication	Ines Heidiani Ikasari, Perani Rosyani, Resti Amalia. "Klasifikasi Jenis Buah Mengg...	<1%
31	Student papers	Sriwijaya University on 2026-05-04	<1%
32	Student papers	UPN Veteran Yogyakarta on 2026-06-02	<1%
33	Student papers	Universitas Jember on 2025-01-15	<1%
34	Student papers	Universitas Muhammadiyah Sukabumi on 2025-07-18	<1%
35	Student papers	Universitas Pamulang on 2026-05-02	<1%
36	Student papers	Universitas Pancasila on 2026-02-21	<1%
37	Internet	jurnal.pustakagalerimandiri.co.id	<1%
38	Student papers	Sekolah Teknik Elektro & Informatika on 2025-07-18	<1%
39	Student papers	Sultan Agung Islamic University on 2018-09-26	<1%

40	Student papers	University of Hertfordshire on 2025-04-03	<1%
41	Internet	repository.metrouniv.ac.id	<1%
42	Internet	repository.stiki.ac.id	<1%
43	Student papers	UPN Veteran Yogyakarta on 2026-03-02	<1%
44	Internet	ejurnal.seminar-id.com	<1%
45	Internet	repository.uinsaizu.ac.id	<1%
46	Student papers	Sriwijaya University on 2022-06-03	<1%
47	Student papers	Sultan Agung Islamic University on 2026-02-16	<1%
48	Student papers	Universitas Islam Riau on 2025-08-22	<1%
49	Student papers	Universitas Islam Riau on 2025-12-10	<1%
50	Internet	repository.unhas.ac.id	<1%
51	Student papers	Defense University on 2024-07-10	<1%
52	Student papers	Forum Perpustakaan Perguruan Tinggi Indonesia Jawa Timur on 2025-07-08	<1%
53	Student papers	Universitas Tarumanagara on 2025-12-23	<1%

54	Internet	adoc.pub	<1%
55	Internet	docobook.com	<1%
56	Internet	ejurnal.methodist.ac.id	<1%
57	Internet	eprints.unimudasorong.ac.id	<1%
58	Internet	repository.upnjatim.ac.id	<1%
59	Student papers	Universitas Islam Riau on 2026-05-13	<1%
60	Student papers	Universitas Jember on 2026-04-06	<1%
61	Student papers	Wilmington University on 2025-05-04	<1%
62	Internet	text-id.123dok.com	<1%
63	Internet	www.frontiersin.org	<1%
64	Student papers	Sriwijaya University on 2025-11-24	<1%
65	Student papers	UNIVERSITAS BUDI LUHUR on 2026-02-21	<1%
66	Internet	ceur-ws.org	<1%
67	Internet	positori.buddhidharma.ac.id	<1%

68	Internet	trojrobert.github.io	<1%
69	Student papers	Forum Perpustakaan Perguruan Tinggi Indonesia Jawa Timur on 2025-08-14	<1%
70	Student papers	IUBH - Internationale Hochschule Bad Honnef-Bonn on 2025-03-08	<1%
71	Publication	Ririn Anugerah Ikasatya, Cahya Apriliani, Fathir Jannatul Firdaus, Gede Yogi Prata...	<1%
72	Student papers	Universitas Budi Luhur on 2025-07-10	<1%
73	Student papers	Universitas Negeri Semarang - iTh on 2024-07-14	<1%
74	Internet	digilibadmin.unismuh.ac.id	<1%
75	Internet	jurnal.stkipppgritulungagung.ac.id	<1%
76	Student papers	Sriwijaya University on 2022-08-01	<1%
77	Student papers	Universitas Muhammadiyah Sukabumi on 2026-01-13	<1%
78	Internet	rstudio-pubs-static.s3.amazonaws.com	<1%
79	Internet	www.jsisfotek.org	<1%
80	Student papers	Fakultas Teknik on 2025-08-07	<1%
81	Student papers	INTI Universal Holdings SDM BHD on 2025-03-14	<1%

82	Student papers	STT PLN on 2024-08-15	<1%
83	Student papers	UNIVERSITAS BUDI LUHUR on 2026-01-27	<1%
84	Student papers	UPN Veteran Yogyakarta on 2026-05-05	<1%
85	Student papers	Universitas Dian Nuswantoro on 2026-04-27	<1%
86	Internet	doktersehatceria.blogspot.com	<1%
87	Internet	etheses.uin-malang.ac.id	<1%
88	Internet	informatika.unpkediri.ac.id	<1%
89	Internet	www.informatica.si	<1%
90	Student papers	Abdullah Gul University on 2025-11-20	<1%
91	Student papers	Fakultas Teknik on 2025-08-21	<1%
92	Publication	Poonam Rani, Tushar Dahiya, Dhruv Gupta, Prem Yadav, Aditya, Tushar Sachdev...	<1%
93	Student papers	Telkom University on 2026-04-26	<1%
94	Student papers	University of Lancaster on 2022-05-17	<1%
95	Internet	repo.darmajaya.ac.id	<1%

96	Publication	Amelia Arimdani Arifin, Rismayanti Fauziah, Dian Rahmaniar Trisnaputri. "The Fo...	<1%
97	Publication	Muhammad Irvan Maulana, Savana Putra Aditama, Harun Al Rosyid. "Komparasi...	<1%
98	Publication	Nur Alim M. Suma, Lillyan Hadjaratie, Nikmasari Pakaya, Indhitya R. Padiku. "Inte...	<1%
99	Student papers	STT PLN on 2024-08-06	<1%
100	Student papers	Telkom University on 2026-03-27	<1%
101	Student papers	Universitas Bengkulu on 2024-05-08	<1%
102	Student papers	Universitas Maritim Raja Ali Haji on 2023-08-29	<1%
103	Student papers	Universitas Maritim Raja Ali Haji on 2025-06-12	<1%
104	Student papers	Universitas Muslim Indonesia on 2025-08-19	<1%
105	Student papers	Universitas Nasional on 2021-01-11	<1%
106	Internet	ejurnal.stmik-budidarma.ac.id	<1%
107	Internet	eprints.upj.ac.id	<1%
108	Internet	github.com	<1%
109	Internet	jacis.pubmedia.id	<1%

110	Internet	pdfcoffee.com	<1%
111	Internet	qdoc.tips	<1%
112	Student papers	Asia e University on 2024-05-07	<1%
113	Student papers	Institut Teknologi Sumatera on 2026-06-17	<1%
114	Publication	Maya Alfiyana. "PERANCANGAN WEBSITE UNTUK MEDIA PEMBELAJARAN BAHASA ...	<1%
115	Student papers	Politeknik APP on 2026-05-18	<1%
116	Student papers	Sekolah Teknik Elektro & Informatika on 2025-08-04	<1%
117	Student papers	Sultan Agung Islamic University on 2024-08-31	<1%
118	Student papers	Telkom University on 2026-01-23	<1%
119	Student papers	UPN Veteran Yogyakarta on 2026-04-22	<1%
120	Student papers	Universitas Indonesia on 2026-06-11	<1%
121	Student papers	Universitas Islam Indonesia on 2026-04-20	<1%
122	Student papers	Universitas Islam Indonesia on 2026-05-12	<1%
123	Student papers	Universitas Islam Riau on 2026-06-11	<1%

124	Student papers	Universitas Kristen Duta Wacana on 2026-05-07	<1%
125	Student papers	Universitas Negeri Malang on 2025-08-11	<1%
126	Student papers	Universitas Pancasila on 2025-08-10	<1%
127	Student papers	Universitas Tarumanagara on 2025-12-24	<1%
128	Internet	journal.politeknik-pratama.ac.id	<1%
129	Internet	jurnal.risetilmiah.ac.id	<1%
130	Internet	redwerk.com	<1%
131	Internet	theses.hal.science	<1%
132	Internet	5aprincesscengeng.blogspot.com	<1%
133	Publication	Abdul Kohar, Aulia Rizky Muhammad Hendrik Noor Asegaft, Brian Sebastian Sali...	<1%
134	Student papers	FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM on 2026-06-09	<1%
135	Student papers	Institut Teknologi Sumatera on 2025-09-22	<1%
136	Publication	Leny Margaretha Huizen, Roy Rudolf Huizen. "Optimalisasi Keamanan IoT dan Ed...	<1%
137	Publication	Muhamad Rizvi Roshan, Hafiz Irsyad. "Klasifikasi Kerusakan Uang Rupiah Mengg...	<1%

138	Student papers	Perpustakaan on 2025-08-06	<1%
139	Student papers	Telkom University on 2025-07-04	<1%
140	Student papers	Transylvania University on 2024-07-25	<1%
141	Student papers	UIN Walisongo on 2024-09-28	<1%
142	Student papers	UPN Veteran Yogyakarta on 2026-03-31	<1%
143	Student papers	UPN Veteran Yogyakarta on 2026-05-04	<1%
144	Student papers	Universitas Ahmad Dahlan on 2026-01-20	<1%
145	Student papers	Universitas Bengkulu on 2024-07-23	<1%
146	Student papers	Universitas Borneo Tarakan on 2026-06-18	<1%
147	Student papers	Universitas Budi Luhur on 2026-01-31	<1%
148	Student papers	Universitas Gadjah Mada on 2025-07-15	<1%
149	Student papers	Universitas Gunadarma on 2025-09-25	<1%
150	Student papers	Universitas Islam Indonesia on 2026-06-15	<1%
151	Student papers	Universitas Islam Riau on 2025-12-15	<1%

152	Student papers	Universitas Negeri Surabaya The State University of Surabaya on 2026-05-05	<1%
153	Student papers	Universitas Pamulang on 2026-05-04	<1%
154	Student papers	Universitas Siliwangi on 2025-06-05	<1%
155	Student papers	Universitas Sumatera Utara on 2026-06-04	<1%
156	Student papers	Universitas Tarumanagara on 2025-12-22	<1%
157	Internet	dspace.uii.ac.id	<1%
158	Internet	id.123dok.com	<1%
159	Internet	jurnal.iaii.or.id	<1%
160	Internet	jurnal.polibatam.ac.id	<1%
161	Internet	ojs.stmik-banjarbaru.ac.id	<1%
162	Internet	repositori.unsil.ac.id	<1%
163	Internet	repository.its.ac.id	<1%
164	Internet	rumahcoding.id	<1%
165	Internet	www.ngopibareng.id	<1%

166	Student papers	Clayton College & State University on 2021-07-19	<1%
167	Student papers	Fakultas Teknik on 2025-08-21	<1%
168	Student papers	Tarumanagara University on 2026-06-20	<1%
169	Student papers	Universitas Pamulang on 2026-06-17	<1%
170	Student papers	Universitas Tarumanagara on 2025-12-12	<1%
171	Internet	repository.usd.ac.id	<1%
172	Internet	www.kafekepo.com	<1%
173	Student papers	STT PLN on 2024-08-28	<1%
174	Student papers	Telkom University on 2025-07-04	<1%
175	Student papers	Universitas Dian Nuswantoro on 2026-01-04	<1%
176	Student papers	Universitas Diponegoro on 2021-08-17	<1%
177	Student papers	Universitas Islam Indonesia on 2025-09-17	<1%
178	Student papers	Universitas Tarumanagara on 2025-12-23	<1%
179	Student papers	iGroup on 2013-06-20	<1%

BAB I

Pendahuluan

A. Latar Belakang

Kanker kulit merupakan salah satu penyakit kanker dengan angka kejadian tertinggi di dunia. *World Health Organization* (WHO) mencatat sekitar 20 juta kasus kanker baru dan 9,9 juta kematian secara global pada tahun 2020, dan kanker kulit termasuk dalam kelompok yang paling banyak didiagnosis (Sung et al., 2021). Di antara seluruh subtipe kanker kulit, adalah bentuk yang paling mematikan. Meskipun kasusnya hanya mencakup sekitar 5% dari seluruh kasus kanker kulit, bertanggung jawab atas sebagian besar kematian akibat penyakit ini karena kemampuannya *bermetastasis* secara cepat (Alotaibi & AlSaeed, 2025). WHO memperkirakan lebih dari 132.000 penderita baru terdiagnosis setiap tahunnya di seluruh dunia (Alotaibi & AlSaeed, 2025). Deteksi dini menjadi faktor penentu keberhasilan pengobatan. Apabila ditemukan pada tahap awal, tingkat kelangsungan hidup pasien dapat mencapai 99%, namun angka ini turun drastis apabila penyakit telah menyebar ke organ lain (Nguyen et al., 2022).

Secara konvensional, diagnosis *Melanoma* dilakukan melalui pemeriksaan visual *dermoskopi* oleh dokter spesialis, kemudian dikonfirmasi melalui biopsi jaringan. Meskipun biopsi memiliki akurasi mendekati 100%, prosedur ini bersifat invasif, mahal, dan memakan waktu antara satu hari hingga beberapa minggu (Alotaibi & AlSaeed, 2025). Selain itu, pemeriksaan visual sangat bergantung pada keahlian individu dan rentan terhadap ketidakkonsistenan antar pemeriksa, khususnya di fasilitas kesehatan dengan keterbatasan tenaga ahli dermatologi (Kumar Lilhore et al., 2024).

Tantangan fundamental yang secara konsisten menghambat performa model pada *dataset dermoskopi* adalah ketidakseimbangan kelas. Pada *dataset HAM10000*, kelas *melanocytic nevi* mendominasi dengan lebih dari 6.700 sampel, sementara kelas *Melanoma* hanya berjumlah sekitar 1.113 sampel, menciptakan rasio ketidakseimbangan mendekati 1:6 (Al-Asadi & Altun, 2022). Kondisi ini mendorong model untuk memiliki bias terhadap kelas mayoritas, sehingga menghasilkan nilai *Recall* yang rendah pada kelas *Melanoma*. Dalam konteks klinis, kondisi ini berbahaya karena berarti banyak kasus kanker yang luput dari deteksi (Nguyen et al., 2022). Berbagai strategi telah diusulkan untuk mengatasi masalah ini. Al-Asadi & Altun (2022) menggunakan *SMOTE* pada *HAM10000* dan mencapai *Recall* 88%. Javaid et al. (2025) mengintegrasikan *SMOTE-Tomek* dengan arsitektur *CNN* khusus dan memperoleh *Recall* 97%. Nguyen et al. (2022) mengusulkan fungsi *loss* baru yang peka terhadap ketidakseimbangan data dan menemukan bahwa *MobileNetV3Large* mampu berjalan 30 kali lebih cepat dari *InceptionResNetV2* dengan akurasi 86%. Oktoeberza et al. (2025) menggunakan *random oversampling* dengan *DenseNet201* dan mencapai akurasi 97%. Namun, penelitian-penelitian tersebut umumnya menguji satu strategi penanganan *imbalance*, berfokus pada klasifikasi multi kelas, dan menggunakan arsitektur besar yang tidak dirancang untuk

perangkat *mobile*. Belum ada kajian yang secara sistematis membandingkan beberapa strategi penanganan ketidakseimbangan kelas, termasuk

Pada penelitian skripsi ini mengusulkan perbandingan sistematis lima strategi penanganan ketidakseimbangan kelas, yaitu *class weighting*, *SMOTE*, *weighted random sampler*, kombinasi *class weighting* dan *SMOTE*, serta *Focal Loss*, pada arsitektur *MobileNetV3-Small* untuk klasifikasi biner *Melanoma* versus *Nevus* menggunakan dataset *HAM10000*. *MobileNetV3-Small* dipilih karena dirancang khusus untuk efisiensi komputasi pada perangkat *mobile*, dengan jumlah parameter yang jauh lebih kecil dibandingkan arsitektur konvensional tanpa kehilangan akurasi secara signifikan (Mamun et al., 2025). Kumar Lilhore et al. (2024) telah menunjukkan bahwa varian *MobileNet-V3* yang dioptimalkan mampu mencapai presisi 97,84% pada *HAM10000*, sementara Nguyen et al. (2022) membuktikan efisiensi *MobileNetV3* dalam skenario inferensi cepat. Seluruh skenario eksperimen dalam penelitian ini menggunakan kerangka pelatihan yang identik sehingga perbedaan performa yang terukur dapat sepenuhnya dikaitkan dengan strategi penanganan *imbalance* yang divariasikan. Metrik utama yang digunakan adalah *Recall* kelas *Melanoma*, sebagai representasi klinis dari kemampuan model meminimalkan kasus kanker yang tidak terdeteksi.

B. Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka dapat diidentifikasi beberapa permasalahan sebagai berikut:

1. Terdapat kemiripan visual yang tinggi antara lesi *Melanoma* dan *melanocytic nevi*, khususnya pada tahap awal.
2. Distribusi kelas pada dataset *HAM10000* tidak seimbang di mana jumlah citra *Nevus* jauh lebih dominan dibandingkan citra *Melanoma* yang berpotensi menyebabkan bias prediksi terhadap kelas mayoritas.
3. Model klasifikasi dengan performa tinggi umumnya memiliki kompleksitas komputasi yang besar, sehingga kurang optimal untuk diterapkan pada perangkat *mobile* dengan keterbatasan sumber daya.

C. Rumusan Masalah

Berdasarkan latar belakang dan identifikasi masalah yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah:

1. Bagaimana kinerja arsitektur *MobileNetV3* dalam mengklasifikasikan citra biner *Melanoma* dan *Nevus* pada dataset *HAM10000* yang memiliki distribusi kelas tidak seimbang?
2. Bagaimana pengaruh penerapan strategi penanganan ketidakseimbangan data (*data imbalance handling*) terhadap peningkatan nilai sensitivitas (*Recall*) model dalam mendeteksi kelas *Melanoma*?
3. Bagaimana konfigurasi hiperparameter optimal yang dapat meningkatkan nilai *Recall* pada model?

D. Batasan Masalah

Batasan masalah yang akan digunakan dalam penelitian ini adalah sebagai berikut:

1. Penelitian difokuskan pada klasifikasi biner citra *dermoskopi*, yaitu *Melanoma* dan *Nevus*, menggunakan arsitektur *MobileNetV3-Small*.

2. Menggunakan *dataset HAM10000* dari arsip ISIC.
3. Penelitian ini tidak mencakup deteksi atau klasifikasi jenis penyakit kulit lainnya di luar dua kelas dan .
4. Implementasi menggunakan bahasa pemrograman Python dengan *framework PyTorch*.
5. Metrik evaluasi yang digunakan meliputi akurasi, *Recall*, presisi, *F1-Score*, *confusion matrix* dan *AUC (Area Under the Curve)*
6. Penelitian ini hanya menggunakan model *MobileNetV3-Small*

E. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dipaparkan, tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Mengimplementasikan arsitektur *Convolutional Neural Network (CNN)* berbasis *MobileNetV3* untuk melakukan klasifikasi biner pada citra *dermoskopi Melanoma* dan *Nevus* menggunakan *dataset HAM10000*.
2. Menerapkan strategi penanganan ketidakseimbangan data guna meningkatkan nilai *Recall* model dalam mendeteksi kelas *Melanoma*.
3. Melakukan optimasi hiperparameter menggunakan metode pencarian otomatis untuk mendapatkan konfigurasi terbaik yang menghasilkan nilai *Recall* (Sensitivitas) *Melanoma* yang optimal sekaligus menjaga nilai *Precision* pada ambang batas yang dapat diterima secara klinis.

F. Manfaat Penelitian

Manfaat Penelitian yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Manfaat Teoritis: Memberikan kontribusi pada literatur ilmiah mengenai efektivitas arsitektur *lightweight (MobileNetV3)* dalam pengolahan citra medis. Hasil penelitian ini dapat dijadikan referensi untuk pengembangan model *Deep Learning* yang hemat performa diagnostik yang memadai untuk skenario deteksi dini..
2. Manfaat Praktis: Menghasilkan sebuah aplikasi dan *prototipe model* klasifikasi *Melanoma* yang memiliki tingkat sensitivitas tinggi dan efisien secara komputasi. Model ini dapat menjadi fondasi teknis bagi pengembangan sistem deteksi dini berbasis perangkat *mobile*.

BAB II

Landasan Teori

A. Teori dan Penelitian Terdahulu

1. Landasan Teori

a) *Melanoma*

Kanker kulit merupakan kondisi patologis yang ditandai oleh pertumbuhan sel kulit yang tidak terkontrol akibat terganggunya mekanisme regenerasi sel normal (Zulfa et al., 2024). Salah satu bentuk kanker kulit yang memiliki tingkat keganasan paling tinggi adalah *Melanoma*, yang berkembang dari sel melanosit, yaitu sel penghasil pigmen melanin. Walaupun jumlah kasus *Melanoma* relatif lebih sedikit dibandingkan jenis kanker kulit lainnya, penyakit ini menjadi penyebab utama tingginya angka kematian akibat kanker kulit karena sifatnya yang agresif dan kemampuannya untuk menyebar ke organ lain dalam waktu yang relatif cepat (Naseri & Safaei, 2025). Paparan radiasi sinar *ultraviolet* (UV) dalam jangka panjang diketahui sebagai faktor risiko utama yang berperan dalam pembentukan *Melanoma*, karena dapat menyebabkan kerusakan dan mutasi pada DNA sel kulit (Naseri & Safaei, 2025). Dalam praktik klinis, upaya deteksi dini *Melanoma* umumnya dilakukan dengan menggunakan kriteria ABCDE, yang mencakup ketidaksimetrian bentuk lesi *Asymmetry Border Color Evolution* sebagaimana dijelaskan pada Tabel 2. 1.

Tabel 2. 1 Perbedaan *Melanoma* dan *Nevus*

Kriteria (ABCDE)	<i>Nevus</i> (Jinak)	<i>Melanoma</i> (Ganas)
A - <i>Asymmetry</i> (Asimetri)	Biasanya simetris (walaupun <i>Nevus</i> atipikal bisa sedikit asimetris).	Bentuknya asimetris.
B - <i>Border</i> (Batas)	Batas tepi lesi biasanya teratur.	Batas tepi tidak teratur (irregular).
C - <i>Color</i> (Warna)	<i>Pigmentasi</i> warna seragam (uniformly pigmented) pada <i>Nevus</i> biasa.	Memiliki variasi warna yang beragam (color variegation).
D - <i>Diameter</i> (Ukuran)	Ukurannya biasanya di bawah 6 mm.	Ukuran diameternya lebih dari 6 mm.
E - <i>Evolution</i> (Evolusi/Perubahan)	Lesi umumnya stabil atau perlahan menyusut seiring berjalannya waktu.	Lesi mengalami evolusi atau perubahan seiring berjalannya waktu.

Sumber: Diolah dari (Zhang et al., 2024)

Secara singkat, kriteria ABCDE menunjukkan bahwa *Nevus* (tahi lalat jinak) memiliki karakteristik yang seragam dan stabil, ditandai dengan bentuk yang simetris, batas tepi yang teratur, *pigmentasi* warna yang merata, ukuran kurang dari 6 mm, dan tidak mengalami perubahan seiring waktu. Sebaliknya, *Melanoma* menunjukkan indikasi keganasan melalui bentuk yang asimetris, pinggiran yang tidak beraturan, warna yang bervariasi atau belang

dalam satu lesi, diameter yang melebihi 6 mm, serta terus mengalami pertumbuhan atau perubahan (evolusi) yang nyata.

Tantangan utama dalam diagnosis *Melanoma* adalah kemiripan visualnya dengan *Nevus* melanositik atau tahi lalat. Perbedaan mendasar antara keduanya seringkali sulit diidentifikasi secara kasat mata. Selain itu, temuan medis terbaru menunjukkan bahwa lebih dari 70% *Melanoma* muncul secara de novo (baru) dan tidak berkembang dari *Nevus* yang sudah ada sebelumnya, sehingga anggapan bahwa setiap *Nevus* berpotensi menjadi kanker adalah kurang tepat (Naseri & Safaei, 2025). Oleh karena itu, diperlukan metode yang presisi untuk membedakan lesi *Nevus* yang jinak dengan *Melanoma* yang ganas guna menghindari kesalahan diagnosis.

Untuk membantu visualisasi lesi kulit, digunakan teknik dermoskopi, yaitu teknik pencitraan non-invasif yang memungkinkan visualisasi struktur kulit yang lebih dalam dibandingkan dengan mata telanjang (Naseri & Safaei, 2025). Penggunaan dermoskopi terbukti meningkatkan akurasi diagnosis dari sekitar 60% menjadi kisaran 75% hingga 84% dibandingkan pemeriksaan visual standar (Naseri & Safaei, 2025). Citra dermoskopi kini menjadi standar data dalam pengembangan sistem diagnosis berbasis kecerdasan buatan karena kemampuannya menampilkan pola pigmentasi yang detail.

Dalam penelitian ini, data yang digunakan bersumber dari dataset HAM10000 (*Human Against Machine with 10000 training images*). Dataset ini sendiri merupakan kumpulan besar citra dermoskopi yang dikumpulkan dari populasi berbeda dan disimpan dengan berbagai modalitas untuk mengatasi masalah kurangnya keragaman data dalam pelatihan jaringan saraf (Tschandl et al., 2018). Dataset ini memuat 10.015 citra yang mencakup tujuh kategori diagnostik, di mana lebih dari 50% lesi telah dikonfirmasi melalui pemeriksaan patologi, sementara sisanya dikonfirmasi melalui follow-up atau konsensus ahli (Tschandl et al., 2018). Namun, dataset ini memiliki karakteristik distribusi kelas yang sangat tidak seimbang, khususnya antara kelas *Nevus* dan *Melanoma*, yang menjadi fokus penyelesaian masalah dalam penelitian ini.

b) *Machine Learning* dan *Deep Learning*

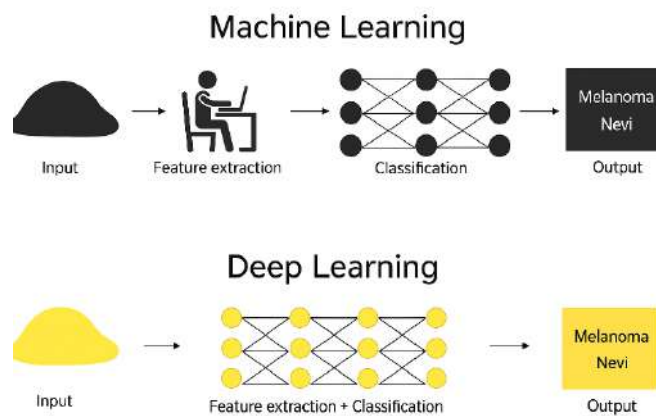
1. *Machine Learning*

Machine Learning (ML) merupakan cabang kecerdasan buatan yang memungkinkan sistem untuk belajar dari data dan meningkatkan kinerjanya dari waktu ke waktu tanpa diprogram secara eksplisit untuk setiap aturan (Naseri & Safaei, 2025). Dalam konteks penelitian ini *Machine Learning* digunakan untuk memproses data citra dermoskopi dan mengekstraksi fitur relevan seperti asimetri, ketidakteraturan tepi, dan variasi warna untuk membedakan lesi jinak dan ganas.

Berdasarkan cara pembelajarannya, penelitian ini menerapkan pendekatan *Supervised Learning* (Pembelajaran Terawasi). *Supervised Learning* merupakan Jenis pembelajaran yang mana algoritma mesin dilatih menggunakan dataset berlabel. Tujuannya adalah untuk mempelajari pola antara input dan output untuk menghasilkan prediksi yang akurat pada data baru.

2. *Deep Learning*

Deep Learning merupakan pendekatan lanjutan dalam *Machine Learning* yang memanfaatkan jaringan saraf tiruan berlapis untuk mempelajari pola kompleks secara langsung dari data mentah. Dalam konteks penelitian ini, *Deep Learning* dipilih karena kemampuannya mengolah citra dermoskopi secara *end-to-end* tanpa memerlukan proses ekstraksi fitur manual yang bergantung pada subjektivitas ahli (Naseri & Safaei, 2025). Berbeda dengan pendekatan *Machine Learning* tradisional yang mengharuskan perancangan fitur secara eksplisit, *Deep Learning* mampu mengekstraksi representasi fitur secara otomatis dan bertahap, mulai dari pola visual sederhana hingga karakteristik *morfologi* yang lebih kompleks. (Naseri & Safaei, 2025). Perbedaan alur kerja kedua pendekatan ini dapat dilihat pada Gambar 2.2.

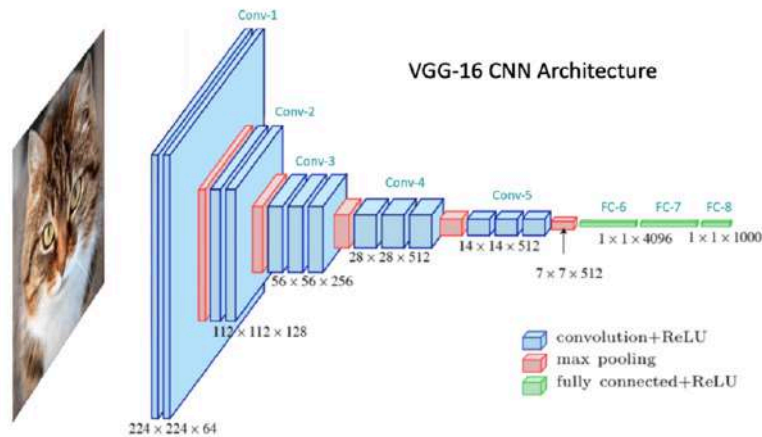


Gambar 2. 1 Perbedaan alur kerja Machine Learning dan Deep Learning

c) *Convolutional Neural Network*

Dalam konteks penelitian ini, *CNN* dimanfaatkan untuk memproses data visual seperti citra dermoskopi karena *CNN* memiliki kemampuan untuk mempelajari representasi fitur secara hierarkis dari data piksel mentah tanpa memerlukan ekstraksi fitur manual. Struktur umum *CNN* terdiri dari dua tahapan utama: ekstraksi fitur dan klasifikasi (Naseri & Safaei, 2025).

Sebagaimana diilustrasikan pada Gambar 2.2 arsitektur *CNN* tersusun atas tumpukan lapisan (layers) yang bekerja secara berurutan:



Gambar 2. 2 Arsitektur umum CNN (Bill Kromydas, 2023)

Komponen utama penyusun CNN meliputi:

1. Lapisan Konvolusi (*Convolutional Layer*): Ditunjukkan oleh blok berwarna biru pada gambar. Lapisan ini merupakan fondasi utama CNN yang menerapkan filter (kernel) pada citra input untuk mengekstraksi pola visual seperti garis, tepi, dan tekstur. Setiap operasi konvolusi biasanya diikuti oleh fungsi aktivasi non-linear seperti ReLU (*Rectified Linear Unit*) untuk memungkinkan model mempelajari pola yang kompleks.
2. Lapisan Pooling (*Max Pooling Layer*): Ditunjukkan oleh blok berwarna merah. Lapisan ini berfungsi untuk mereduksi dimensi spasial dari *feature maps*. Teknik yang umum digunakan adalah *Max Pooling*, yang mengambil nilai piksel terbesar dalam area tertentu. Tujuannya adalah mengurangi beban komputasi dan memori, serta membuat model lebih tahan (*robust*) terhadap pergeseran kecil pada posisi objek dalam citra.
3. Lapisan Terhubung Penuh (*Fully Connected Layer*): Ditunjukkan oleh blok berwarna hijau di akhir arsitektur. Setelah fitur diekstraksi dan direduksi dimensinya, data diratakan (*flatten*) menjadi vektor satu dimensi dan dimasukkan ke dalam jaringan saraf biasa untuk melakukan klasifikasi akhir (misalnya: menentukan probabilitas antara *Melanoma* atau *Melanositik Nevi*). Kombinasi dari lapisan-lapisan ini memungkinkan CNN untuk menangkap detail visual dari level rendah hingga tinggi secara otomatis, menjadikannya metode standar dalam diagnosis medis berbasis citra.

d) Mekanisme Pelatihan Model

Proses pembelajaran pada *Deep Learning* melibatkan dua tahapan utama yang dilakukan secara berulang (iteratif), yaitu *Forward Propagation* dan *Backward Propagation*. Proses pelatihan jaringan saraf tiruan bertujuan untuk menemukan parameter bobot (W) yang meminimalkan fungsi kerugian ($J(W)$). Proses ini dilakukan melalui dua tahapan iteratif:

1. Forward Propagation

Pada tahap ini, data input diproses melewati lapisan-lapisan jaringan untuk menghasilkan prediksi. Mekanisme ini

direpresentasikan secara matematis sebagai komposisi fungsi dari setiap lapisan jaringan (*Deep Feedforward Networks*). Berdasarkan (Vakalopoulou et al., 2023), hubungan antara input dan output prediksi melalui K lapisan didefinisikan sebagai berikut:

$$\hat{y} = f(x; W^1, \dots, W^K) = z_k(z_{k-1}(\dots(z_1(x))\dots)) \quad (2.1)$$

Di mana z_k merepresentasikan operasi linear pada lapisan ke- k (didefinisikan sebagai $z = Wx + b$) yang diikuti oleh fungsi aktivasi.

Di mana $\hat{y}$ adalah hasil prediksi akhir, x adalah data input (citra), dan z_k adalah fungsi transformasi pada setiap lapisan ke- k .

Jika prediksi $\hat{y}$ telah diperoleh, maka tujuan pelatihan adalah meminimalkan selisih antara prediksi tersebut dengan label asli y . Fungsi kerugian (*Loss Function*) yang umum digunakan untuk klasifikasi adalah *Categorical Cross-Entropy Loss*, yang didefinisikan sebagai (Vakalopoulou et al., 2023):

$$J(W) = - \sum_{i=1}^n \log(P(y = y^{(i)} | x = x^{(i)}; W)) \quad (2.2)$$

Penjelasan Variabel:

- $J(W)$ = Nilai total kerugian (loss) yang ingin diminimalkan.
- n = Jumlah sampel data dalam satu batch.
- $P(x^{(i)}; W)$ = Probabilitas prediksi kelas benar $y^{\{i\}}$ untuk sampel ke- i , dihitung melalui softmax atas logits y .
- $y^{(i)}$ = Label kelas yang sebenarnya dari data ke- i
- $x^{(i)}$ = Data input (citra) ke- i .
- W = Parameter bobot seluruh jaringan (termasuk weights dan biases).

2. Backward Propagation

Setelah nilai *loss* dihitung, algoritma *Backpropagation* digunakan untuk menghitung gradien kerugian terhadap setiap bobot dalam jaringan. Proses ini menerapkan aturan rantai (*chain rule*) untuk merambatkan kesalahan dari output kembali ke input, guna mengetahui seberapa besar kontribusi setiap bobot terhadap kesalahan tersebut. Secara matematis, *gradien* bobot untuk lapisan tertentu dapat dihitung sebagai berikut (Vakalopoulou et al., 2023):

$$\frac{\partial J(W)}{\partial w} = \frac{\partial J(W)}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial z} \times \frac{\partial z}{\partial w} \quad (2.3)$$

Penjelasan Variabel:

- $\frac{\partial J(W)}{\partial w}$ = Gradien (tingkat perubahan) loss terhadap bobot w . Nilai inilah yang dicari untuk memperbaiki bobot.

- $\frac{\partial J(W)}{\partial \hat{y}} =$ Seberapa besar perubahan prediksi $\hat{y}$ mempengaruhi $loss$.
- $\frac{\partial \hat{y}}{\partial z} =$ Seberapa besar perubahan output lapisan z mempengaruhi prediksi akhir
- $\frac{\partial z}{\partial w} =$ Seberapa besar perubahan bobot w mempengaruhi output lapisan z .

3. Fungsi Aktivasi (*Activation Functions*)

Fungsi aktivasi adalah komponen krusial dalam jaringan saraf yang bertugas memperkenalkan non-linearitas ke dalam model, memungkinkan jaringan untuk mempelajari pola data yang kompleks. Tanpa fungsi aktivasi, jaringan saraf sedalam apa pun hanya akan berperilaku seperti model *regresi linear* biasa. Dalam penelitian ini, digunakan dua jenis fungsi aktivasi utama:

- a. *Rectified Linear Unit (ReLU)*, adalah fungsi aktivasi yang paling umum digunakan pada lapisan tersembunyi (*hidden layer*) karena efisiensi komputasinya dan kemampuannya mengurangi masalah *vanishing gradient*. Secara matematis, *ReLU* didefinisikan sebagai fungsi yang mengembalikan nilai input jika positif, dan 0 jika negatif (Vakalopoulou et al., 2023):

$$g(x) = (0, x) \# \dots (2.4)$$

Penjelasan Variabel:

- $g(x)$ = Output dari fungsi aktivasi ReLU
- = Nilai input dari neuron (hasil penjumlahan bobot dan bias).
- $(0, x)$ = Operasi matematika yang mengambil nilai terbesar antara 0 dan x . Jika x negatif, hasilnya 0. Jika x positif, hasilnya x itu sendiri.

Turunan dari ReLU sangat sederhana, yaitu bernilai 1 untuk $x > 0$ dan 0 untuk $x \leq 0$, yang mempercepat proses komputasi backward propagation.

b. *Swish dan Hard-Swish (h-swish)*

Fungsi *Swish* pertama kali diperkenalkan sebagai pengganti *ReLU* karena mampu meningkatkan akurasi model secara signifikan. Rumus swish adalah:

$$swish = x \cdot \sigma(x) \# \dots (2.5)$$

Penjelasan Variabel:

- x = nilai input ke fungsi aktivasi (output dari operasi konvolusi atau linear sebelumnya)
- $\sigma(x) = \frac{1}{1+e^{-x}}$ = Fungsi sigmoid

Meskipun Swish memberikan peningkatan akurasi, fungsi *sigmoid* di dalamnya mahal secara komputasi pada perangkat mobile (karena melibatkan operasi eksponensial) dan rentan terhadap kehilangan presisi pada mode kuantisasi (int8). Untuk

mengatasi masalah tersebut, *MobileNetV3* memperkenalkan hard-swish (h-swish) sebagai aproksimasi piecewise linear dari *Swish*. Rumusnya adalah:

$$h - swish(x) = x \cdot \frac{ReLU6(x+3)}{6} \#.. (2.6)$$

Penjelasan Variabel:

- $ReLU6(z) = ((0, z), 6)$: fungsi ReLU yang dibatasi maksimum 6 (untuk stabilitas dan kuantisasi),
- x : nilai input (dapat negatif atau positif),
- Pembagian dengan 6 dan penambahan +3 dipilih karena memberikan kecocokan terbaik dengan kurva *Swish* asli secara sederhana.

c. *Hard-Sigmoid* (h-sigmoid)

Fungsi sigmoid konvensional juga digunakan pada modul *Squeeze-and-Excitation* (SE) untuk menghitung skala per channel. Namun, karena alasan yang sama (komputasi mahal dan masalah kuantisasi), sigmoid diganti dengan *hard-sigmoid*. Rumusnya adalah:

$$h - sigmoid(x) = \frac{ReLU6(x+3)}{6} \#.... (2.7)$$

Penjelasan Variabel:

- x : nilai input (biasanya output dari fully-connected layer pada SE module setelah global average pooling),
- Output berada pada rentang [0, 1] seperti sigmoid, tetapi berbentuk piecewise linear.

4. Algoritma Optimasi (*Optimization Algorithms*)

Algoritma optimasi bertugas memperbarui parameter bobot (W) jaringan berdasarkan gradien kesalahan yang dihitung selama backward propagation.

- a. *Gradient Descent*, Ini adalah algoritma optimasi dasar di mana bobot diperbarui dengan mengambil langkah berlawanan arah dengan gradien fungsi kerugian. Rumus pembaruannya adalah (Vakalopoulou et al., 2023):

$$W^{t+1} \leftarrow W^t - \eta \cdot \nabla J(W^t) \# \# \# ... (2.8) \# \# \#$$

Penjelasan Variabel:

- W^{t+1} = Nilai bobot baru yang sudah diperbarui.
- W^t = Nilai bobot lama (saat ini).
- η = *Learning rate*, yaitu parameter yang menentukan seberapa besar langkah perubahan bobot yang diambil.
- $\nabla J(W^t)$ = Gradien total yang menunjukkan arah kesalahan. Tanda minus (–) berarti bobot digeser ke arah yang berlawanan dengan kesalahan untuk mengurangi *loss*.

148

- b. *Adaptive Moment Estimation (Adam)*, adalah algoritma optimasi berbasis gradien stokastik yang diusulkan oleh Kingma & Ba (2015). Algoritma ini dirancang untuk menghitung *learning rate* adaptif secara individual bagi setiap parameter model, sehingga cocok digunakan pada masalah dengan parameter berdimensi tinggi dan data dalam skala besar. Nama "Adam" berasal dari kemampuannya memperkirakan momen pertama dan momen kedua dari gradien secara adaptif (adaptive moment estimation).

Adam mengintegrasikan keunggulan dua metode optimasi sebelumnya: AdaGrad, yang efektif untuk gradien sparse, dan *RMSProp*, yang bekerja baik pada fungsi objektif yang tidak stasioner (Kingma & Ba, 2017). Berbeda dengan *Stochastic Gradient Descent* (SGD) yang menggunakan satu nilai *learning rate* global untuk semua parameter, Adam menyesuaikan besar langkah pembaruan secara otomatis berdasarkan histori gradien masing-masing parameter.

24

Pada setiap langkah pelatihan, Adam mempertahankan dua variabel akumulasi untuk setiap parameter: momen pertama m_t (rata-rata bergerak dari gradien, merepresentasikan arah update) dan momen kedua v_t (rata-rata bergerak dari kuadrat gradien, merepresentasikan besarnya variasi gradien). Kedua variabel ini diinisialisasi dengan nilai nol pada awal pelatihan.

24

Secara formal, proses pembaruan parameter θ pada iterasi ke- t mengikuti langkah-langkah berikut (Kingma & Ba, 2017):

Langkah 1 Hitung gradien:

$$g_t = \nabla f(\theta_{t-1}) \quad (2.9)$$

Langkah 2 Update momen pertama (mean gradien):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2.10)$$

Langkah 3 Update momen kedua (varians gradien):

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2.11)$$

Langkah 4 Koreksi bias momen pertama:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2.12)$$

Langkah 5 Koreksi bias momen kedua:

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \# \dots (2.13)$$

Langkah 6 Pembaruan parameter:

$$\theta_t = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \# \dots (2.14)$$

Penjelasan variabel:

- α = *learning rate*.
- β_1 = koefisien peluruhan eksponensial untuk momen pertama (default 0,9).
- β_2 = koefisien peluruhan eksponensial untuk momen kedua (default 0,999).
- ϵ = konstanta kecil untuk mencegah pembagian dengan nol (default 10^{-8}).
- g_t = gradien fungsi loss terhadap parameter pada iterasi ke- t .
- m_t = estimasi momen pertama.
- v_t = estimasi momen kedua.
- $\hat{m}_t$ = estimasi momen pertama setelah koreksi bias.

Karena m_t dan v_t diinisialisasi dengan nol, estimasi keduanya cenderung bias mendekati nol pada iterasi-iterasi awal pelatihan, terutama ketika β_1 dan β_2 mendekati nilai 1. Koreksi bias pada Langkah 4 dan 5 menggunakan pembagi $(1 - \beta_1^t)$ dan $(1 - \beta_2^t)$ yang nilainya mendekati nol di awal iterasi dan mendekati 1 seiring bertambahnya t (Kingma & Ba, 2015). Dengan demikian, pembaruan parameter lebih besar di awal pelatihan (ketika estimasi masih belum stabil) dan semakin kecil saat model mendekati konvergensi, menghasilkan perilaku anil otomatis (*automatic annealing*).

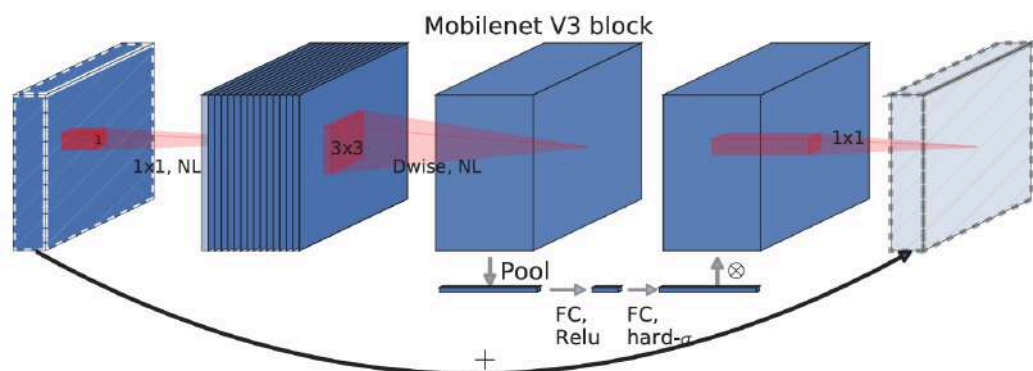
Nilai hiperparameter default yang direkomendasikan oleh Kingma & Ba (2015) adalah $\alpha = 0,001$, $\beta_1 = 0,9$, $\beta_2 = 0,999$, dan $\epsilon = 10^{-8}$. Pada penelitian ini, Adam digunakan sebagai optimizer pada seluruh skenario eksperimen (S1–S4) dengan *learning rate* $\alpha = 1 \times 10^{-4}$ (S1–S3) dan *differential learning rate* pada S4 (backbone: 1×10^{-5} , classifier: 1×10^{-4}).

e) *MobileNetV3*

Merupakan pengembangan terbaru dalam keluarga arsitektur *MobileNet* yang dirancang oleh Google dengan fokus utama pada efisiensi komputasi untuk perangkat seluler dan sistem dengan keterbatasan sumber daya (Howard et al., 2019). Arsitektur ini dikembangkan sebagai penyempurnaan dari *MobileNetV2* dengan tujuan mencapai keseimbangan yang lebih optimal antara kecepatan inferensi dan tingkat akurasi model. Untuk mencapai tujuan tersebut, *MobileNetV3* dirancang melalui pendekatan otomatis berbasis *Network Architecture Search (NAS)* yang dikombinasikan dengan algoritma

NetAdapt, sehingga konfigurasi jaringan yang dihasilkan dapat diadaptasi secara efektif dengan karakteristik perangkat keras mobile (Howard et al., 2019).

Secara konseptual, *MobileNetV3* masih mengadopsi prinsip *inverted residual blocks* dan *linear bottleneck* sebagaimana pada *MobileNetV2*, namun dengan sejumlah peningkatan pada struktur bloknya. Salah satu pembaruan penting adalah penyematan modul *Squeeze-and-Excitation* (SE) di dalam bottleneck, yang memungkinkan jaringan melakukan penyesuaian bobot kanal secara dinamis. Mekanisme ini membantu model untuk menekankan fitur-fitur yang paling informatif sekaligus mengurangi kontribusi fitur yang kurang relevan, sehingga meningkatkan efisiensi representasi tanpa menambah beban komputasi secara signifikan (Howard et al., 2019). Ilustrasi struktur blok *MobileNetV3* dengan modul SE ditampilkan pada Gambar 2.4.

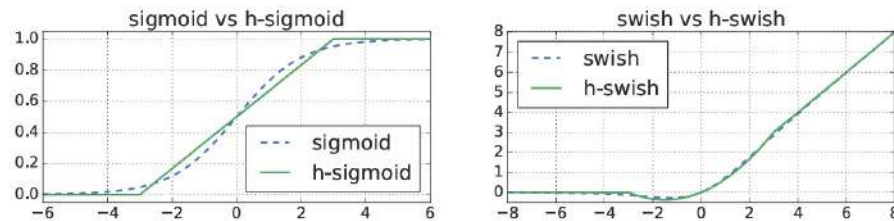


Gambar 2.3 Blok Arsitektur *MobileNetV3* yang mengintegrasikan modul *Squeeze-and-Excitation* (SE) pada layer residual (Howard et al., 2019)

Selain perubahan struktural, *MobileNetV3* juga memperkenalkan fungsi aktivasi baru yang disebut *hard-swish* (h-swish). Fungsi ini merupakan modifikasi dari fungsi aktivasi *swish* standar. (Howard et al., 2019) menemukan bahwa meskipun fungsi *swish* dapat meningkatkan akurasi, fungsi tersebut tidak efisien untuk dihitung pada perangkat *mobile* karena mengandung operasi sigmoid yang mahal secara komputasi. Oleh karena itu, *h-swish* dirumuskan sebagai pendekatan linier dengan rumus:

$$h - swish[x] = x \frac{ReLU(x+3)}{6} \# \dots (2.15)$$

Fungsi *h-swish* ini terbukti memberikan akurasi yang setara dengan *swish* asli namun dengan biaya komputasi yang jauh lebih rendah dan lebih ramah terhadap kuantisasi. Perbandingan visual antara fungsi *sigmoid/swish* asli dengan versi *hard* (keras) dapat dilihat pada Gambar 2.5.



Gambar 2. 4 Perbandingan fungsi aktivasi Sigmoid dan Swish dengan versi efisiennya (Howard et al., 2019)

Terdapat dua varian model yang dirilis, yaitu *MobileNetV3-Large* dan *MobileNetV3-Small*. Dalam penelitian ini, digunakan varian *MobileNetV3-Small* karena target penelitian berfokus pada efisiensi maksimal dan kecepatan inferensi yang tinggi. Berdasarkan eksperimen (Howard et al., 2019), *MobileNetV3-Small* menunjukkan peningkatan akurasi sebesar 6,6% dibandingkan *MobileNetV2* dengan latensi yang setara, menjadikannya pilihan ideal untuk klasifikasi lesi kulit pada perangkat dengan daya komputasi rendah.

Alasan memilih model ini dan bukan model *MobileNetV4* adalah karena tujuan spesifik penelitian ini untuk mendapatkan model yang paling ringan mengingat selisih beban komputasinya mencapai hampir 3 kali lipat dari *MobileNetV3-Small* 0,066 MACs (Howard et al., 2019) vs *MobileNetV4-Conv-Small* 0,2 MACs (Qin et al., 2024). MACs sendiri merupakan jumlah operasi perkalian dan akumulasi yang dilakukan model selama proses inferensi. peningkatan akurasi minor dari V4 tidak sebanding dengan hilangnya efisiensi dan peningkatan konsumsi daya yang terjadi.

f) *Confusion Matrix*

Untuk mengukur kinerja model klasifikasi yang dikembangkan, penelitian ini menggunakan *Confusion Matrix* sebagai alat evaluasi utama. *Confusion Matrix* adalah tabel representasi yang membandingkan hasil prediksi model dengan label kebenaran dasar (*ground truth*). Komponen dasar dari matriks ini terdiri dari empat parameter utama (Kumar Lilhore et al., 2024):

1. *True Positive* (TP): Jumlah data positif (*Melanoma*) yang diprediksi dengan benar sebagai positif.
2. *True Negative* (TN): Jumlah data negatif (*Nevus*) yang diprediksi dengan benar sebagai negatif.
3. *False Positive* (FP): Jumlah data negatif yang salah diprediksi sebagai positif (Kesalahan Tipe I).
4. *False Negative* (FN): Jumlah data positif yang salah diprediksi sebagai negatif (Kesalahan Tipe II).

Berdasarkan komponen tersebut, kinerja model diukur menggunakan empat metrik evaluasi berikut:

1. Akurasi (*Accuracy*) Akurasi mengukur proporsi total prediksi yang benar (baik positif maupun negatif) terhadap keseluruhan data. Dalam konteks formal, akurasi didefinisikan sebagai rasio prediksi yang tepat dari model (Kumar Lilhore et al., 2024):

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \#... (2.16)$$

2. Presisi (*Precision*) Presisi menggambarkan proporsi individu yang didiagnosis positif secara benar oleh model terhadap total hasil prediksi positif. Metrik ini menunjukkan seberapa handal model saat menyatakan suatu lesi sebagai *Melanoma* (Kumar Lilhore et al., 2024):

$$Presisi = \frac{TP}{TP+FP} \# ... (2.17)$$

3. Sensitivitas atau *Recall* Sensitivitas, yang juga dikenal sebagai *Recall* atau *True Positive Rate*, mengukur kemampuan model dalam mengenali seluruh kasus positif yang sebenarnya. Metrik ini sangat krusial dalam diagnosis kanker untuk meminimalkan kasus yang terlewat (Kumar Lilhore et al., 2024):

$$Sensitivitas = \frac{TP}{TP+FN} \#... (2.18)$$

4. *F1-Score* (*F-Measure*) *F1-Score* adalah rata-rata harmonik dari Presisi dan *Recall*. Metrik ini memberikan penilaian tunggal yang menyeimbangkan antara presisi dan sensitivitas, sehingga sangat efektif untuk mengevaluasi model pada dataset yang tidak seimbang (Kumar Lilhore et al., 2024):

$$F1 \text{ score} = 2 \times \frac{Presisi \times Sensitivitas}{Presisi + Sensitivitas} \#... (2.19)$$

5. *Area Under the Curve* (*AUC*) merupakan area di bawah kurva ROC (Receiver Operating Characteristic). Kurva ROC adalah grafik yang menggambarkan hubungan antara True Positive Rate (TPR) dan False Positive Rate (FPR) pada berbagai nilai threshold keputusan. TPR identik dengan Sensitivitas (*Recall*), mengukur proporsi kasus *Melanoma* yang berhasil dideteksi. FPR mengukur proporsi kasus *Nevus* yang salah diklasifikasikan sebagai *Melanoma*. Nilai AUC berkisar antara 0 hingga 1: AUC = 0,5 menunjukkan performa setara pengklasifikasi acak, sedangkan AUC = 1,0 menunjukkan performa sempurna. AUC bersifat robust terhadap ketidakseimbangan kelas karena tidak bergantung pada satu nilai threshold. Persamaan matematisnya dinyatakan sebagai berikut (Kumar Lilhore et al., 2024).

$$TPR = \frac{TP}{TP+FN} \#... (2.20)$$

$$FPR = \frac{FP}{FP+TN} \#... (2.21)$$

$$AUC = \int_{i=1}^{n-1} (FPR_{i+1} - FPR_i) \left(\frac{TPR_i + TPR_{i+1}}{2} \right) \#... (2.22)$$

- g) Penanganan ketidakseimbangan data
1. *Class Weight* (*Pembobotan Kelas*)

Class Weight adalah metode berbasis fungsi loss yang memberikan bobot lebih besar kepada kelas minoritas selama pelatihan (Ghosh et al., 2024). Bobot dihitung secara invers proporsional terhadap frekuensi kelas dalam data latih sehingga kesalahan pada kelas minoritas dihukum lebih berat. Persamaan perhitungan bobot kelas adalah:

$$w(c) = \frac{N}{K \times n(c)} \# \dots (2.23)$$

di mana N adalah total sampel data latih, K adalah jumlah kelas, dan $n(c)$ adalah jumlah sampel pada kelas c . Bobot ini kemudian diterapkan sebagai parameter weight pada fungsi CrossEntropyLoss.

2. SMOTE (Synthetic Minority Over-sampling Technique)

SMOTE adalah teknik oversampling yang menghasilkan sampel sintesis baru pada kelas minoritas dengan cara interpolasi linier antara sampel yang ada dan tetangga terdekatnya (Chawla et al., 2002). Sampel sintesis dibangkitkan menggunakan persamaan:

$$x_{\text{baru}} = x_i + \lambda \cdot (x_{nn} - x_i) \# \dots (2.24)$$

di mana x_i adalah sampel minoritas, x_{nn} adalah salah satu tetangga terdekat (k-NN) dari x_i , dan λ adalah bilangan acak dalam rentang $[0, 1]$. Dalam penelitian ini SMOTE diterapkan pada indeks data latih menggunakan library imbalanced-learn.

3. Weighted Random Sampling (WRS)

Weighted Random Sampling adalah teknik berbasis sampler yang mengatur probabilitas pengambilan sampel per batch agar kelas minoritas lebih sering muncul dalam setiap iterasi pelatihan (Ghosh et al., 2022). Berbeda dengan SMOTE, WRS tidak menambah data secara fisik. Setiap sampel diberi bobot sampling sebesar invers dari jumlah sampel kelasnya, kemudian PyTorch WeightedRandomSampler menggunakan bobot ini untuk memilih sampel dengan replacement pada setiap batch.

4. Kombinasi (SMOTE + Class Weight)

Metode Kombinasi menerapkan SMOTE dan Class Weight secara bersamaan (Ghosh et al., 2022). SMOTE menyeimbangkan distribusi data secara fisik pada level dataset, sementara Class Weight memberikan penalti tambahan pada fungsi loss. Pendekatan ganda ini bertujuan memaksimalkan perhatian model terhadap kelas minoritas dari dua sisi sekaligus, namun berpotensi menghasilkan double-penalization yang terlalu agresif.

5. Focal Loss

Focal Loss adalah fungsi loss dinamis yang secara otomatis menurunkan bobot sampel yang mudah diklasifikasikan sehingga model terfokus pada sampel yang sulit (Lin et al., 2017). Persamaan Focal Loss adalah:

$$FL(p_t) = -\alpha_t \cdot (1 - p_t)^\gamma \cdot \log(p_t) \# \dots (2.25)$$

di mana p_t adalah probabilitas kelas yang benar, α_t adalah faktor penyeimbang berbasis distribusi kelas ($\alpha = 0,8576$ untuk penelitian ini), dan γ adalah focusing parameter ($\gamma = 2$). Ketika $\gamma = 0$, Focal Loss identik dengan Cross-Entropy standar. Semakin besar γ , semakin kuat penekanan pada sampel sulit.

h) *Transfer Learning dan Fine-Tuning*

Transfer learning adalah teknik dalam deep learning yang memanfaatkan pengetahuan (knowledge) yang telah diperoleh model dari satu domain sumber (source domain) untuk diterapkan pada domain target yang berbeda (Tan et al., 2018). Pendekatan ini sangat relevan dalam konteks dataset medis yang berukuran terbatas, karena pengumpulan dan anotasi data klinis membutuhkan biaya tinggi dan keahlian khusus (Tan et al., 2018).

Dalam transfer learning berbasis CNN, model yang telah dilatih pada dataset berskala besar seperti *ImageNet* digunakan sebagai titik awal. Lapisan-lapisan awal jaringan (yang berfungsi sebagai ekstraktor fitur umum seperti tepi, tekstur, dan pola) dipertahankan, sementara lapisan klasifikasi akhir diganti dan disesuaikan dengan tugas baru (Tan et al., 2018). Strategi ini terbagi menjadi dua pendekatan utama: feature extraction (seluruh backbone dibekukan dan hanya classifier yang dilatih) dan fine-tuning (sebagian atau seluruh lapisan backbone dibuka kembali dan dilatih dengan *learning rate* yang lebih kecil).

Pada penelitian ini, *MobileNetV3-Small* yang telah dilatih pada *ImageNet* digunakan sebagai backbone. Pada tahap awal (S1 dan S2), backbone dibekukan sepenuhnya (frozen) agar fitur *ImageNet* tidak berubah dan hanya lapisan classifier yang disesuaikan. Pada tahap akhir (S4), seluruh lapisan backbone dibuka (unfreeze all) dengan *learning rate* yang dibedakan antara backbone dan classifier (differential *learning rate*) untuk memungkinkan adaptasi fitur secara menyeluruh terhadap domain citra dermoskopi.

i) *Augmentasi Data*

Augmentasi data adalah teknik untuk memperluas ukuran dataset pelatihan secara artifisial melalui penerapan berbagai transformasi pada citra asli (Shorten & Khoshgoftaar, 2019). Teknik ini bertujuan meningkatkan kemampuan generalisasi model dan mengurangi risiko overfitting, terutama pada kondisi data pelatihan yang terbatas (Shorten & Khoshgoftaar, 2019).

Transformasi geometris merupakan kategori augmentasi yang paling umum digunakan, mencakup operasi seperti random cropping, horizontal flip, dan rotasi. Shorten & Khoshgoftaar (2019) menyatakan bahwa transformasi-transformatasi tersebut terbukti efektif dalam berbagai tugas klasifikasi citra karena menghasilkan variasi posisi dan orientasi objek tanpa mengubah label kelas.

Pada penelitian ini, augmentasi hanya diterapkan pada data pelatihan (training set), sedangkan data validasi dan pengujian hanya melalui operasi standar tanpa augmentasi untuk memastikan evaluasi yang tidak bias. Transformasi yang digunakan meliputi *RandomResizedCrop* (224×224

piksel), *RandomHorizontalFlip*, dan rotasi acak hingga 15° , diikuti normalisasi menggunakan mean dan standar deviasi *ImageNet*.

j) *Early Stopping*

Early stopping adalah salah satu teknik regularisasi dalam pelatihan jaringan saraf tiruan yang bertujuan mencegah overfitting dengan menghentikan proses pelatihan secara otomatis ketika performa model pada data validasi tidak lagi menunjukkan peningkatan selama sejumlah epoch tertentu (Prechelt, 1998). Jumlah epoch toleransi sebelum pelatihan dihentikan ini disebut patience.

Goodfellow et al. (2016) menjelaskan bahwa early stopping bekerja berdasarkan prinsip bahwa validation loss yang mulai meningkat meskipun training loss terus menurun merupakan indikasi bahwa model telah memasuki fase overfitting, yaitu menghafal data pelatihan daripada belajar pola yang dapat digeneralisasi. Dengan menyimpan bobot model terbaik berdasarkan validation loss terendah (best model checkpoint), metode ini memastikan model yang digunakan pada tahap pengujian adalah model dengan kemampuan generalisasi optimal.

Pada penelitian ini, early stopping diterapkan pada semua skenario dengan nilai patience = 7, artinya pelatihan dihentikan apabila validation loss tidak mengalami penurunan selama 7 epoch berturut-turut. Nilai ini dipilih sebagai keseimbangan antara memberikan waktu yang cukup bagi model untuk melewati fluktuasi sementara pada loss curve dan mencegah komputasi yang tidak perlu.

k) *Stratified Split*

Pembagian dataset (dataset splitting) adalah proses memisahkan data menjadi subset pelatihan (training set), validasi (validation set), dan pengujian (test set) yang digunakan pada fase berbeda dalam siklus pengembangan model. Raschka (2018) mendefinisikan stratified splitting sebagai teknik pembagian yang mempertahankan proporsi kelas asli pada setiap subset yang dihasilkan, sehingga distribusi kelas pada training set, validation set, dan test set mencerminkan distribusi pada dataset keseluruhan.

Stratifikasi ini menjadi sangat penting ketika dataset bersifat tidak seimbang (imbalanced). Tanpa stratifikasi, pembagian acak berisiko menghasilkan subset di mana kelas minoritas kurang terwakili atau bahkan tidak muncul sama sekali, yang dapat menyebabkan estimasi performa model menjadi bias (Raschka, 2020).

Pada penelitian ini, dataset *HAM10000* (mel=1.113 vs. nv=6.705) dibagi secara stratifikasi dengan rasio 70:20:10 untuk masing-masing training set, validation set, dan test set, dengan random seed = 42 untuk menjamin reproduktibilitas eksperimen. Stratifikasi memastikan proporsi *Melanoma* (~14,2%) terjaga konsisten di ketiga subset, sehingga evaluasi model pada test set merepresentasikan kondisi data yang sesungguhnya.

2. Kajian Pustaka

Al-Asadi & Altun (2022) meneliti penanganan ketidakseimbangan kelas dan kelangkaan data pada klasifikasi lesi kulit berbasis citra dermoskopi. Mereka mengusulkan penggunaan SMOTE untuk menghasilkan data sintetis pada dataset

69 69

HAM10000 yang mencakup tujuh kelas lesi, kemudian melatih model Deep Learning Convolutional Neural Network (DLCNN). Hasilnya menunjukkan akurasi 85,99%, presisi 90%, *Recall* 88%, dan F1-score 88%. Keterbatasan penelitian ini terletak pada pengujian yang hanya dilakukan terhadap satu strategi penyeimbangan data tanpa perbandingan langsung terhadap pendekatan lain seperti class weighting atau Focal Loss, sehingga tidak memberikan gambaran komparatif tentang efektivitas relatif setiap strategi.

Nguyen et al. (2022) meneliti klasifikasi lesi kulit pada data tidak seimbang menggunakan deep learning dengan mekanisme soft-attention. Mereka mengkombinasikan beberapa arsitektur *CNN* dengan soft-attention, mengusulkan fungsi loss baru yang memperhitungkan ketidakseimbangan data, serta memanfaatkan informasi personal pasien seperti usia dan jenis kelamin. Eksperimen pada *HAM10000* menunjukkan bahwa MobileNetV3Large dengan soft-attention dan fungsi loss baru mencapai akurasi 86% dengan kecepatan inferensi 30 kali lebih cepat dari InceptionResNetV2. Keterbatasan penelitian ini adalah tidak adanya perbandingan terisolasi antar strategi penanganan imbalance pada arsitektur yang sama, sehingga kontribusi spesifik fungsi loss versus mekanisme soft-attention sulit dipisahkan.

Javaid et al. (2025) meneliti klasifikasi kanker kulit multi-kelas dengan fokus pada penanganan ketidakseimbangan dataset dermoskopi. Mereka mengusulkan DSSCC-Net, arsitektur *CNN* yang diintegrasikan dengan teknik SMOTE-Tomek secara pasca-pembagian data untuk mencegah kebocoran data. Model diuji pada tiga dataset (*HAM10000*, ISIC 2018, PH2) dan dievaluasi menggunakan Grad-CAM untuk interpretabilitas. Hasilnya menunjukkan akurasi rata-rata 97,82%, presisi 97%, *Recall* 97%, dan AUC 99,43%. Keterbatasan penelitian ini adalah penggunaan resolusi input yang sangat kecil (28x28 piksel) dan arsitektur yang tidak dirancang untuk penerapan pada perangkat mobile, berbeda dengan fokus penelitian ini yang menggunakan *MobileNetV3-Small* pada resolusi 224x224 untuk mendukung deployment *Android*.

Kumar Lilhore et al. (2024) meneliti diagnosis kanker kulit menggunakan kombinasi U-Net dan MobileNet-V3 yang ditingkatkan dengan optimasi hiperparameter Bayesian. Mereka mengganti fungsi loss cross-entropy standar MobileNet-V3 dengan bias loss function, mengganti komponen squeeze-and-excitation dengan practical channel attention untuk mengurangi jumlah parameter, serta mengintegrasikan cross-layer connections dan dilated convolutions. Model diuji pada *HAM10000* dan menghasilkan presisi 97,84%, sensitivitas 96,35%, akurasi 98,86%, dan spesifisitas 97,32%. Keterbatasan penelitian ini terletak pada kompleksitas arsitektur hibrida yang menggabungkan segmentasi dan klasifikasi multi-kelas, sehingga kontribusi spesifik setiap komponen sulit diisolasi dan tidak dapat langsung dibandingkan dengan pendekatan berbasis strategi imbalance tunggal.

Mateen et al. (2024) meneliti diagnosis *Melanoma* menggunakan kerangka hybrid deep learning berbasis citra dermoskopi. Mereka mengusulkan kombinasi U-Net untuk segmentasi, InceptionResNet-V2 untuk ekstraksi fitur, dan Vision Transformer dengan mekanisme self-attention untuk penyempurnaan fitur, dilengkapi dengan hiperparameter tuning. Dataset ISIC2020 digunakan sebagai dataset utama, sementara *HAM10000* digunakan untuk studi ablasi. Hasilnya mencapai akurasi 98,65%, sensitivitas 99,20%, dan spesifisitas 98,03%

20

pada ISIC2020. Keterbatasan penelitian ini adalah kompleksitas arsitektur yang sangat tinggi sehingga tidak praktis diterapkan pada perangkat mobile dengan sumber daya komputasi terbatas.

Alotaibi & AlSaeed (2025) meneliti deteksi kanker kulit menggunakan transfer learning dengan berbagai mekanisme attention yang diintegrasikan ke dalam arsitektur Xception. Mereka membandingkan empat model pada *HAM10000*: Xception standar serta Xception yang dikombinasikan dengan self-attention, hard attention, dan soft attention. Hasilnya menunjukkan bahwa self-attention memberikan akurasi tertinggi sebesar 94,11%, meningkat dari 91,05% pada Xception standar, dengan nilai *Recall* yang lebih unggul dibandingkan penelitian sebelumnya. Keterbatasan penelitian ini adalah fokusnya pada mekanisme attention sebagai variabel utama tanpa membahas strategi penanganan ketidakseimbangan kelas secara eksplisit, padahal dataset *HAM10000* memiliki ketidakseimbangan yang signifikan.

Ismail et al. (2025) meneliti klasifikasi citra kanker kulit berpigmen menggunakan kombinasi MobileNetV2 dengan modul Convolutional Block Attention Module (CBAM). Mereka membandingkan tiga pendekatan, yaitu VGG16+CBAM, MobileNetV2+CBAM, dan hibrid VGG16-MobileNetV2, pada subset *HAM10000* yang telah diseimbangkan menjadi 2.400 citra dari tiga kelas (AKIEC, BCC, MEL). Hasilnya menunjukkan bahwa MobileNetV2+CBAM mencapai akurasi terbaik sebesar 86%, presisi 86,53%, dan *Recall* 86,46%. Keterbatasan penelitian ini adalah penggunaan undersampling berat dari 10.015 menjadi 2.400 citra yang berpotensi membuang informasi penting, serta cakupan yang dibatasi hanya pada tiga kelas.

Oktoeberza et al. (2025) meneliti perbandingan model deep learning untuk klasifikasi kanker kulit multi-kelas pada data tidak seimbang menggunakan *HAM10000*. Mereka membandingkan tiga arsitektur transfer learning, yaitu DenseNet201, ResNet50, dan VGG16, dengan teknik random oversampling. DenseNet201 menunjukkan performa terbaik dengan akurasi 97% pasca-penyeimbangan dan akurasi uji 96,52%, melampaui laporan sebelumnya yang berkisar 90-92%. Keterbatasan penelitian ini adalah pengujian hanya pada satu strategi penyeimbangan tanpa perbandingan dengan alternatif lain, serta penggunaan arsitektur besar yang tidak dioptimalkan untuk perangkat mobile.

Mamun et al. (2025) meneliti optimasi deep learning untuk klasifikasi kanker kulit dengan prioritas efisiensi komputasi. Mereka membandingkan model *CNN* ringan khusus dengan ResNet50 berbasis transfer learning menggunakan dataset *HAM10000*, dengan FLOPs sebagai metrik perbandingan utama. Hasilnya menunjukkan bahwa model *CNN* khusus dengan hanya 30,04 juta FLOPs mencapai akurasi yang hanya berbeda 0,022% dari ResNet50 (4 miliar FLOPs) sambil mengurangi parameter sebesar 96,7%. Keterbatasan penelitian ini adalah tidak adanya pembahasan tentang penanganan ketidakseimbangan kelas dan optimasi *Recall* untuk kelas minoritas yang relevan secara klinis.

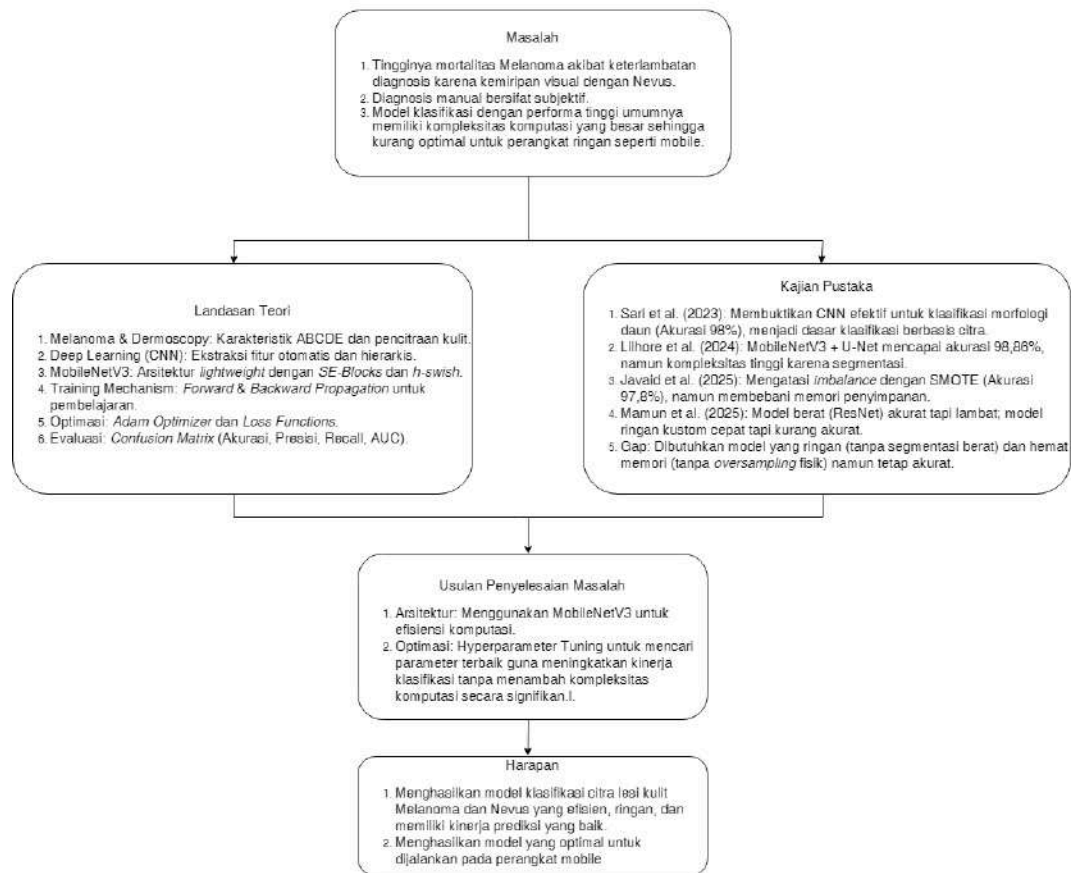
4

Tabel 2. 2 Tabel sintesis penelitian terdahulu

No	Penulis & Tahun	Metode Utama	Dataset	Hasil Utama	Keterbatasan
----	-----------------	--------------	---------	-------------	--------------

1	Al-Asadi & Altun (2022)	SMOTE + DLCNN	<i>HAM10000</i> (7 kelas)	Akurasi 85,99%, <i>Recall</i> 88%	Hanya uji SMOTE, tidak komparatif
2	Nguyen et al. (2022)	MobileNetV3Large + Soft-Attention + Loss baru	<i>HAM10000</i> (7 kelas)	Akurasi 86%, 30× lebih cepat	Tidak isolasi kontribusi setiap komponen
3	Javaid et al. (2025)	DSSCC-Net + SMOTE-Tomek	<i>HAM10000</i> , ISIC2018, PH2	Akurasi 97,82%, <i>Recall</i> 97%	Resolusi 28×28, tidak cocok untuk mobile
4	Kumar Lilhore et al. (2024)	U-Net + MobileNet-V3 + Bayesian Opt	<i>HAM10000</i>	Akurasi 98,86%, Presisi 97,84%	Arsitektur kompleks, multi-kelas
5	Mateen et al. (2024)	U-Net + InceptionResNet-V2 + ViT	ISIC2020 + <i>HAM10000</i>	Akurasi 98,65%, <i>Recall</i> 99,20%	Sangat kompleks, tidak <i>mobile-friendly</i>
6	Alotaibi & AlSaeed (2025)	Xception + Attention Mechanisms	<i>HAM10000</i> (biner)	Akurasi 94,11% (self-attention)	Tidak bahas class imbalance
7	Ismail et al. (2025)	MobileNetV2 + CBAM	<i>HAM10000</i> (3 kelas)	Akurasi 86%, <i>Recall</i> 86,46%	Undersampling berat, 3 kelas saja
8	Oktoberza et al. (2025)	DenseNet201 + Random Oversampling	<i>HAM10000</i> (7 kelas)	Akurasi 97%, Akurasi uji 96,52%	Hanya 1 strategi imbalance, bukan <i>mobile</i>
9	Mamun et al. (2025)	Custom <i>CNN</i> vs ResNet50	<i>HAM10000</i>	Model 34x lebih ringan dari ResNet50 dengan akurasi yang hampir identik	Tidak bahas <i>Recall</i> minoritas / imbalance

B. Kerangka berpikir



Gambar 2. 5 Kerangka Berpikir

Kerangka berpikir penelitian ini dibangun berdasarkan alur permasalahan, landasan teori, tinjauan pustaka, hingga solusi yang diusulkan, sebagaimana diilustrasikan pada Gambar 2.5 Kerangka Berpikir. Penelitian bermula dari masalah tingginya urgensi diagnosis dini kanker kulit *Melanoma*. Tantangan utama yang dihadapi adalah kemiripan visual yang tinggi antara *Melanoma* dan *Nevus*, serta karakteristik dataset medis (*HAM10000*) yang sangat tidak seimbang.

Pada blok Landasan Teori, penelitian ini berpijak pada konsep *Deep Learning* khususnya *Convolutional Neural Network (CNN)* sebagai metode ekstraksi fitur otomatis. Arsitektur *MobileNetV3* dipilih sebagai fondasi utama karena desainnya yang efisien untuk perangkat terbatas, didukung oleh mekanisme pelatihan menggunakan *Forward-Backward Propagation* dan evaluasi menggunakan *Confusion Matrix*.

Blok Kajian Pustaka merangkum evolusi dan kesenjangan penelitian terdahulu. Penelitian (Kumar Lilhore et al., 2024) dan (Javaid et al., 2025) menunjukkan potensi akurasi tinggi namun menyisakan celah (*gap*) berupa kompleksitas komputasi akibat segmentasi dan inefisiensi memori akibat penggunaan teknik *oversampling (SMOTE)*.

Berdasarkan analisis tersebut, Usulan Penyelesaian Masalah dalam penelitian ini berfokus pada efisiensi sistem. Penelitian ini mengusulkan penggunaan arsitektur *MobileNetV3-Small* untuk menjaga beban komputasi tetap rendah. Selain itu, untuk

mengatasi ketidakseimbangan data tanpa membebani memori, diterapkan Strategi Penanganan *Imbalance Data* (seperti *Class Weighted Loss*) yang dikombinasikan dengan Hyperparameter Tuning.

Melalui pendekatan ini, Harapan akhirnya adalah dihasilkannya model klasifikasi yang tidak hanya memiliki akurasi dan sensitivitas tinggi dalam mendeteksi *Melanoma*, tetapi juga efisien dari segi komputasi sehingga layak diimplementasikan pada sistem diagnosis berbantuan komputer yang praktis.

BAB III

Metodologi Penelitian

A. Desain Penelitian

1. Jenis Penelitian

Penelitian ini menggunakan pendekatan hybrid eksperimental dan pengembangan., yang menggabungkan proses perancangan dan implementasi model klasifikasi citra lesi kulit berbasis arsitektur *MobileNetV3* dengan pengujian eksperimental kuantitatif secara terkontrol. Tahap pengembangan berfokus pada pembangunan dan optimalisasi model untuk lingkungan perangkat bergerak, sedangkan tahap eksperimental bertujuan menganalisis pengaruh *hyperparameter tuning* terhadap kinerja klasifikasi dan efisiensi komputasi model dalam membedakan lesi kulit *Melanoma* dan *melanositik nevi*. Variabel independen dalam penelitian ini adalah konfigurasi hiperparameter model, sementara variabel dependen meliputi metrik performa model, yaitu akurasi, presisi, *Recall*, *F1-score*, dan *AUC*, serta indikator efisiensi komputasi seperti waktu inferensi dan kompleksitas model.

2. Variabel Penelitian

a). Variabel Independen (Bebas)

Variabel independen adalah variabel yang dimanipulasi oleh peneliti untuk mengamati pengaruhnya terhadap hasil klasifikasi.

Dalam penelitian ini, variabel independen meliputi:

1. Strategi Penanganan Imbalance Data (5 metode dibandingkan): Class Weight, SMOTE, Weighted Random Sampling (WRS), Kombinasi (SMOTE + Class Weight), dan Focal Loss
2. Konfigurasi Hiperparameter Pelatihan: Nilai parameter pelatihan model yang dioptimasi secara otomatis menggunakan *framework* Optuna. Ruang pencarian parameter ini mencakup *learning rate*, *weight decay*, *dropout rate*, strategi *fine-tuning* (*freeze all*, *unfreeze last*, *unfreeze all*), sudut rotasi augmentasi, dan penggunaan *horizontal flip*.
3. Ambang batas keputusan (Threshold): Batas nilai probabilitas yang digunakan untuk menentukan kelas prediksi. Nilai ini dimanipulasi dari *threshold default* 0,5 untuk menemukan titik keseimbangan optimal antara metrik *Recall* dan *Precision* pada evaluasi akhir.

b). Variabel Dependen (Terikat)

Variabel dependen adalah hasil atau keluaran dari model yang diukur sebagai akibat dari perubahan pada variabel independen.

Dalam penelitian ini, variabel dependen mencakup dua kategori pengukuran:

1. Kinerja Fungsional Model: Tingkat keberhasilan dan keandalan model dalam mengklasifikasikan citra yang diukur menggunakan metrik evaluasi *Confusion Matrix*. Indikator pengukurannya meliputi *Recall* (dengan fokus pada sensitivitas deteksi kelas *Melanoma*), *Precision*, *F1-Score*, *Area Under the Curve* (AUC), *Accuracy*, serta jumlah *False Negative* (FN) dan *False Positive* (FP).
2. Kinerja Non-Fungsional (komputasi): Tingkat efisiensi model ketika diekspor dan diimplementasikan ke dalam sistem. Indikator ini diukur

berdasarkan ukuran penyimpanan model (dalam ukuran MB), total parameter, estimasi kebutuhan RAM, serta waktu inferensi rata-rata pada perangkat CPU, GPU, maupun perangkat *Android* nyata.

c). Variable Kontrol

Variabel kontrol adalah variabel yang dikendalikan atau dibuat konstan sehingga pengaruh variabel independen terhadap variabel dependen tidak dipengaruhi oleh faktor luar yang tidak diteliti.

Dalam penelitian ini, variabel kontrol meliputi:

Tabel 3. 1 Variabel Kontrol

Variabel Kontrol	Nilai Tetap / Keterangan
Dataset	<i>HAM10000</i> (subset: <i>Melanoma</i> dan <i>melanocytic nevi</i>)
Arsitektur Model	<i>MobileNetV3-Small</i>
Ukuran gambar	224 × 224 piksel
Train/Val/Test split	70% / 20% / 10%
Sistem Operasi	Windows
Bahasa pemrograman	<i>Python 3.10</i>
<i>Framework</i>	<i>PyTorch</i>
<i>Optimizer</i>	<i>Adam</i>
<i>Batas Epoch</i>	100 (<i>early stop 7</i>)
<i>Random Seed</i>	42

B. Instrumen Penelitian

Instrumen penelitian merupakan seperangkat alat dan perangkat lunak yang digunakan oleh peneliti untuk melakukan pengumpulan data, pelatihan model, evaluasi, serta analisis hasil eksperimen. Instrumen ini digunakan untuk menjamin bahwa proses penelitian dapat dilakukan secara sistematis, terukur, dan dapat direplikasi oleh peneliti lain.

1. Perangkat Keras

Penelitian ini dilakukan menggunakan perangkat komputer dengan spesifikasi sebagai berikut:

Tabel 3. 2 Spesifikasi Perangkat

Komponen	Spesifikasi
Laptop	MSI GF63 THIN 10SC
Prosesor	Intel CoreI i5-10500H CPU @ 2.50GHz (2.50 GHz)
GPU	GTX 1650

RAM	16 GB
Storage	SSD 512 GB
Sistem Operasi	Windows

Spesifikasi perangkat keras yang dijelaskan [pada tabel 3.2 ini digunakan untuk melakukan proses pelatihan (*training*), validasi, serta pengujian (*testing*) model CNN pada dataset *HAM10000*. GPU digunakan untuk mempercepat proses komputasi saat pelatihan model.

2. Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini terdiri dari sistem operasi, bahasa pemrograman, pustaka pendukung, serta alat bantu untuk pelatihan dan analisis model.

Tabel 3. 3 Software dan Tools yang digunakan

Software dan Tools	Versi
Python	3.11
PyTorch	2.8.0+cu128
TorchVision	0.23.0+cu128
CUDA toolkit	V12.8.93
Pandas, NumPy, scikit-learn, matplotlib	Terbaru
VSCode	1.106.0

3. Dataset

Dataset yang digunakan adalah *HAM10000* (Human Against Machine with 10000 training images), yang diperoleh dari sumber publik *ISIC 2018 Challenge Dataset*. Dataset ini berisi 10.015 citra lesi kulit dengan resolusi 600×450 piksel yang mencakup tujuh kategori, yaitu:

Tabel 3. 4 Deskripsi Lesi Kulit

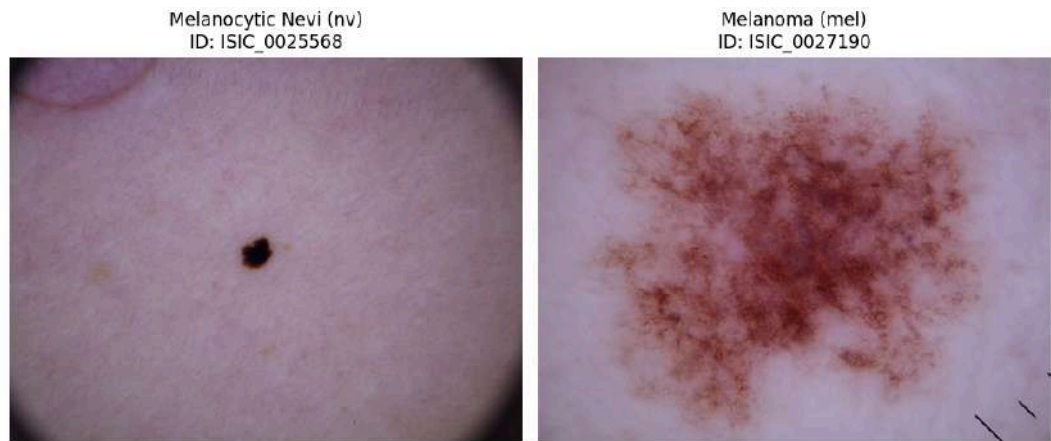
Kode Label	Jenis Lesi Kulit	Keterangan
akiec	Actinic keratoses	Lesi prakanker akibat paparan sinar UV
bcc	Basal cell carcinoma	Kanker kulit non- <i>Melanoma</i>
bkl	Benign keratosis-like lesions	Lesi jinak mirip keratosis
df	Dermatofibroma	Tumor kulit jinak
nv	Melanocytic nevi	Tahi lalat atau <i>Nevus</i> pigmen
mel		Jenis kanker kulit berbahaya

	<i>Melanoma</i>	
vasc	Vascular lesions	Lesi vaskular seperti hemangioma

Namun, pada penelitian ini hanya digunakan dua kelas utama, yaitu:

1. Melanocytic Nevi (nv) – 6.705 citra
2. *Melanoma* (mel) – 1.113 citra

Sehingga total data yang digunakan adalah 7.818 citra. Distribusi data yang tidak seimbang (rasio sekitar 6:1) menjadi fokus penelitian untuk diterapkan metode penyeimbangan. Contoh citra lesi *Melanoma* dan *Nevus* yang digunakan dalam penelitian ditunjukkan pada Gambar 3.1.



Gambar 3. 1 Citra lesi kulit *Nevus* (kiri) dan *Melanoma* (kanan)

4. Analisis Hasil (Alat Ukur Evaluasi)

a. Pengujian Fungsional (Blackbox)

Dilakukan untuk memvalidasi fungsionalitas inti aplikasi tanpa melihat struktur kode program dengan menjalankan skenario manual yang meliputi: (1) Membuka aplikasi, (2) Mengambil foto via kamera, (3) Konfirmasi gambar sebelum klasifikasi, (4) Mengulang pengambilan gambar, (5) Melakukan klasifikasi, (6) Menampilkan label prediksi dan confidence, (7) Kembali ke layar utama.

b. Pengujian Non-Fungsional (Kinerja):

1. Metrik Evaluasi Klasifikasi (*Confusion Matrix*)

Alat ukur ini digunakan untuk mengevaluasi kemampuan diskriminasi dan tingkat keberhasilan model dalam mengklasifikasikan citra ke dalam kelas target (*Melanoma* dan *Nevus*). Metrik evaluasi non-fungsional yang akan diukur meliputi *Recall* (sensitivitas deteksi kelas minoritas *Melanoma*), *Precision*, *Accuracy*, *F1-Score*, serta nilai *AUC-ROC* (*Area Under the Curve - Receiver Operating Characteristic*)..

2. Pengukuran Ukuran Model dan Waktu Inferensi

- Ukuran Model (*Model Size*): Mengukur kapasitas ruang penyimpanan (dalam satuan MB) serta total jumlah parameter yang dihasilkan oleh file model akhir setelah diekspor ke format TorchScript Lite

- Waktu Inferensi (*Inference Time*): Mengukur durasi waktu rata-rata (dalam milidetik atau detik) yang dibutuhkan model untuk memproses satu citra secara *end-to-end*, yang diuji pada lingkungan PC (CPU/GPU) menggunakan *benchmark notebook* serta pada perangkat *smartphone Android* nyata.
- 3. Analisis Konfigurasi Hiperparameter Optimal
Bagian ini digunakan sebagai parameter evaluasi untuk menilai dampak dari pencarian kombinasi nilai parameter terbaik terhadap peningkatan performa model. Evaluasi difokuskan pada nilai konfigurasi hiperparameter yang ditemukan secara otomatis oleh *framework* Optuna, yang mencakup kombinasi *learning rate*, *weight decay*, *dropout rate*, strategi *fine-tuning* pada *backbone*, serta parameter augmentasi citra (*rotation* dan *horizontal flip*).

C. Jadwal Penelitian

Tabel 3. 5 Jadwal Penelitian

No	Kegiatan Penelitian	Oktober	November	Desember	Januari	Februari	Maret
1	Studi Literatur	■					
2	Akuisisi Data		■				
3	Data Preparasi		■				
4	Modeling			■	■	■	
5	Evaluasi			■	■	■	
6	Deployment					■	■

D. Objek Penelitian/Subjek Penelitian

1. Analisis Kebutuhan Sistem

a. Kebutuhan Fungsional

1. Sistem dapat memuat dan memproses *dataset* citra kulit secara spesifik untuk klasifikasi biner pada kelas *Melanoma* dan *Nevus* menggunakan *dataset HAM10000*.
2. Sistem dapat melakukan pra-pemrosesan citra (*preprocessing*) yang meliputi *resize* dengan mempertahankan *aspect ratio* hingga sisi terpendek 256 piksel, pemotongan area tengah (*center crop*) berukuran 224x224 piksel, dan normalisasi warna menggunakan standar *mean* dan *std ImageNet*.
3. Sistem dapat melatih arsitektur *MobileNetV3-Small* dan membandingkan strategi penanganan ketidakseimbangan data yang meliputi *Class Weight*, *SMOTE*, *Weighted Random Sampler (WRS)*, Kombinasi, serta *Focal Loss*.

4. Sistem dapat mengoptimasi parameter pelatihan menggunakan *framework* Optuna untuk mendapatkan konfigurasi *hyperparameter* terbaik
 5. Sistem dapat melakukan evaluasi model menggunakan metrik AUC, akurasi, *Recall*, *Precision*, dan *F1-score*.
 6. Sistem dapat menampilkan hasil prediksi gambar tunggal berupa label ("Melanoma" atau "Melanositik Nevi").
 7. Sistem dapat mengekspor model akhir dari format *PyTorch state dict* (.pth) ke format *TorchScript Lite* (.ptl) agar kompatibel dengan modul *PyTorch Mobile* di *Android*.
 8. Sistem aplikasi *mobile* dapat menjalankan inferensi dan menampilkan label prediksi ("Melanoma" atau "Nevus") beserta skor probabilitas (*confidence*) dalam bentuk persentase
- b. Kebutuhan Non Fungsional
1. Kinerja: Model harus memiliki waktu inferensi stabil dengan rata-rata kecepatan $\pm 1,2$ detik per citra.
 2. Efisiensi: Ukuran model setelah optimasi tidak melebihi 50 MB agar sesuai untuk aplikasi mobile.
 3. Kegunaan (*Usability*): Sistem mudah dijalankan pada perangkat mobile.
- c. Kebutuhan Pengguna
1. Pengguna dapat menjalankan model melalui aplikasi mobile dengan antarmuka sederhana.
 2. Pengguna dapat mengambil foto lesi kulit secara langsung dan memperoleh hasil prediksi secara instan.
 3. Pengguna memperoleh umpan balik visual yang intuitif terkait tingkat bahaya lesi kulit melalui pewarnaan teks hasil: warna merah dengan ikon peringatan untuk prediksi *Melanoma*, dan warna hijau dengan ikon centang untuk prediksi .
- d. Kebutuhan Data
- Jenis data yang digunakan adalah data non-struktural berupa citra dermatoskopik digital. *Dataset* yang digunakan bersumber dari *HAM10000*, yang secara spesifik difilter atau di-*subset* menjadi dua kelas target utama, yaitu dan *Melanositik Nevi*, untuk keperluan pemodelan klasifikasi biner.

2. Objek Penelitian

Objek penelitian ini adalah performa model klasifikasi biner citra kulit (*Melanoma* dan *Nevus*) berbasis *Convolutional Neural Network (CNN)* menggunakan arsitektur *MobileNetV3-Small*. Model ini dipilih karena dirancang khusus untuk memiliki komputasi yang sangat ringan dan efisien untuk diimplementasikan pada perangkat *mobile* sekaligus keluaran terbaru model *MobileNet*.

Penelitian ini menitikberatkan pada:

1. Evaluasi *hyperparameter tuning* dan teknik penyeimbangan data sehingga sistem mampu mengenali pola visual secara lebih akurat pada kedua kelas yang diteliti.
2. Optimasi model agar memiliki ukuran dan waktu proses yang sesuai untuk implementasi pada perangkat dengan keterbatasan sumber daya, tanpa mengurangi kualitas prediksi secara signifikan.

Pemilihan objek ini relevan dengan tujuan penelitian, yaitu menghasilkan model klasifikasi citra kulit yang akurat dan efisien untuk deteksi dini *Melanoma* dan dapat diimplementasikan pada perangkat mobile.

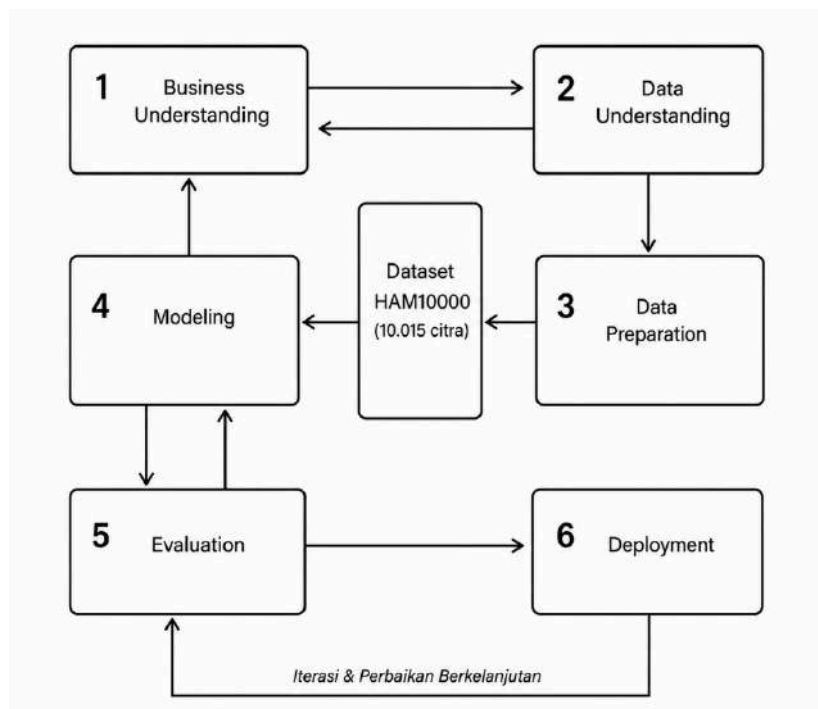
3. Subjek Penelitian

Subjek penelitian ini adalah:

1. Dataset *HAM10000*, yang digunakan sebagai sumber data citra untuk pelatihan dan pengujian model. Dataset ini diambil dari Harvard Dataverse dan berisi berbagai jenis lesi kulit dengan label diagnosis dari ahli dermatologi.
2. Model *MobileNetV3-Small*, sebagai subjek utama yang akan diuji dan dioptimasi.
3. Perangkat Uji (Hardware) berupa laptop dengan GPU NVIDIA GTX 1650 (4 GB) untuk proses pelatihan dan smartphone SAMSUNG A07 untuk pengujian aplikasi.

E. Prosedur Penelitian

Prosedur penelitian ini menggunakan pendekatan *Software Development Life Cycle (SDLC)* model *Cross-Industry Standard Process for Data Mining (CRISP-DM)* yang disesuaikan untuk penelitian berbasis pengembangan dan eksperimen *machine learning*. Metode ini dipilih karena memiliki tahapan yang fleksibel dan sesuai dengan kebutuhan penelitian yang berorientasi pada hasil implementatif berupa model klasifikasi citra yang siap dioptimalkan untuk perangkat mobile.



Gambar 3. 2 Model CRISP-DM

Gambar 3.1 menjelaskan tentang tahapan dari pendekatan *Software Development Life Cycle (SDLC)* dengan model *Cross-Industry Standard Process for Data Mining (CRISP-DM)*. Adapun tahapan penelitian yang dilakukan dapat dijelaskan sebagai berikut:

1. *Business Understanding*

Studi literatur dan analisis kebutuhan sistem dengan meninjau penelitian terdahulu yang relevan mengenai klasifikasi *Melanoma* dan *Melanocytic Nevi* menggunakan deep learning serta mencari tahu dan memahami research gap dari penelitian terdahulu.

2. Data Understanding

Tahap ini adalah tahapan mencari dan menemukan dataset yang pas dan cocok untuk penelitian meliputi Analisis dataset, struktur data, jumlah kelas, ukuran gambar, serta distribusi antar kelas.

3. Data Preparation

preprocessing data, augmentasi data dengan cara *resizing*, *flip*, *rotation* untuk memastikan data siap digunakan dalam proses pelatihan model.

4. Modeling

Pada tahapan ini dilakukan proses pelatihan model secara bertahap melalui empat skenario eksperimen yang dirancang secara berurutan ($S1 \rightarrow S2 \rightarrow S3 \rightarrow S4$). Setiap skenario memiliki tujuan spesifik dan hasilnya menjadi landasan bagi skenario berikutnya. Pembagian data ke dalam train set (70%), validation set (20%), dan test set (10%) dilakukan secara stratifikasi (stratified split) dengan random seed=42 untuk menjamin reproduktibilitas dan mempertahankan proporsi kelas pada setiap subset.

– Skenario 1 Baseline

Skenario pertama bertujuan menetapkan acuan performa awal model tanpa penerapan strategi penanganan ketidakseimbangan data apapun. Backbone *MobileNetV3-Small* dibekukan (*frozen*) seluruhnya sehingga hanya lapisan *classifier* yang dilatih, menggunakan bobot *pretrained ImageNet* sebagai titik awal. Konfigurasi pelatihan yang digunakan: optimizer Adam ($lr=1 \times 10^{-4}$), *batch size* 32, maksimum 100 *epoch*, dan *early stopping* dengan *patience*=7. Konfigurasi ini bersifat tetap (*fixed/fixed*) dan identik di seluruh skenario berikutnya agar perbandingan antar metode adil dan terkontrol. Hasil *baseline* ini dijadikan tolak ukur untuk mengevaluasi efektivitas setiap strategi penanganan *imbalance* pada skenario selanjutnya.

– Skenario 2 Penanganan Ketidakseimbangan Data

Skenario kedua bertujuan membandingkan efektivitas lima strategi penanganan ketidakseimbangan data secara sistematis. Seluruh konfigurasi pelatihan diidentikkan dengan Skenario 1 sehingga perbedaan performa yang terukur sepenuhnya disebabkan oleh strategi imbalance handling yang divariasikan. Lima metode yang dibandingkan adalah: (S2a) *Class Weight* memberikan bobot loss yang lebih besar pada kelas *Melanoma* secara proporsional terbalik terhadap frekuensi kelas, (S2b) *SMOTE* membangkitkan sampel sintetis kelas *Melanoma* melalui interpolasi linier, diterapkan pada data latih setelah *split* untuk menghindari *data leakage*, (S2c) *Weighted Random Sampler (WRS)* memberikan probabilitas sampling lebih tinggi pada kelas minoritas di setiap *batch* tanpa menambah data secara fisik, (S2d) Kombinasi menggabungkan *SMOTE* dan *Class Weight* secara bersamaan, dan (S2e) *Focal Loss* menggantikan *CrossEntropyLoss* standar dengan fungsi *loss* dinamis $FL(p) = -\alpha(1-p)^\beta \cdot \log(p)$ dengan $\alpha=0,858$

dan $\gamma=2$, sehingga model terfokus pada sampel yang sulit diklasifikasikan. Metode terbaik ditentukan berdasarkan nilai *Recall Melanoma* tertinggi yang tidak mengalami *Precision collapse*.

– Skenario 3 Optimasi Hyperparameter

Skenario ketiga bertujuan mencari konfigurasi hiperparameter optimal secara otomatis menggunakan *framework* Optuna berbasis *Tree-structured Parzen Estimator* (TPE). Metode terbaik dari Skenario 2 (Class WeightClass Weight) digunakan sebagai basis. Ruang pencarian mencakup: *learning rate* (10^{-5} hingga 10^{-2}), *weight decay*, *dropout rate*, strategi *fine-tuning backbone* (*freeze_all*, *unfreeze_last*, *unfreeze_all*), sudut rotasi augmentasi, dan penggunaan *horizontal flip*. Fungsi objektif Optuna dirancang dengan kendala *Precision* minimum $\geq 0,30$ pada *validation set* untuk mencegah konfigurasi yang mengalami *Precision collapse*. Dari 20 *trial* yang dijalankan, Trial 9 dipilih sebagai konfigurasi optimal karena menghasilkan keseimbangan terbaik antara *Recall* tinggi dan *Precision* yang dapat diterima secara klinis.

– Skenario 4 Pelatihan Final

Skenario keempat merupakan tahap pelatihan final menggunakan konfigurasi hiperparameter terbaik hasil Skenario 3 (Trial 9). Perbedaan utama dari skenario sebelumnya adalah strategi *fine-tuning unfreeze_all*, di mana seluruh lapisan *backbone MobileNetV3-Small* dibuka untuk dilatih bersama lapisan *classifier*. Untuk mencegah *learning rate* yang terlalu besar merusak fitur yang sudah dipelajari *backbone*, diterapkan *differential learning rate*: *backbone* menggunakan $lr=4,811 \times 10^{-5}$ (sepersepuluh dari *lr classifier*) sementara *classifier* menggunakan $lr=4,811 \times 10^{-4}$. Class WeightClass Weight tetap diterapkan sebagai strategi penanganan *imbalance*. Pelatihan menghasilkan model terbaik berdasarkan *validation loss* terendah yang disimpan sebagai *best_model.pth*.

– Threshold Tuning

Setelah pelatihan final selesai, dilakukan analisis *threshold tuning* untuk menentukan batas keputusan optimal selain nilai *default 0,5*. Model menghasilkan probabilitas $P(\text{Melanoma})$ untuk setiap sampel; secara *default*, sampel diklasifikasikan sebagai *Melanoma* jika $P(\text{Melanoma}) > 0,5$. Proses *tuning* dilakukan dengan pencarian berurutan untuk nilai *threshold* dari 0,05 hingga 0,95 (langkah 0,01) pada *validation set*. Kriteria pemilihan: dari seluruh nilai *threshold* yang menghasilkan *Recall Melanoma* $\geq 0,80$ pada *validation set*, dipilih yang menghasilkan nilai *Precision* tertinggi, sehingga *false positive* dapat diminimalkan tanpa mengorbankan *Recall* di bawah batas yang ditetapkan. Nilai batas *Recall* 0,80 ditetapkan oleh peneliti sebagai batas minimum yang dinilai memadai untuk tujuan skrining awal. Hasil analisis menunjukkan *threshold* optimal pada nilai 0,67, yang menghasilkan *Recall*=0,8288 dan *Precision*=0,5610 pada *test set*. Meskipun demikian, untuk keperluan aplikasi *Android* yang ditujukan sebagai alat skrining awal, *threshold* 0,5 tetap digunakan karena *Recall* yang lebih tinggi (0,8829) lebih diprioritaskan dalam konteks skrining klinis. Hasil *threshold tuning* tetap dilaporkan sebagai analisis *trade-off* untuk memberikan informasi kepada pengembang atau penelitian lanjutan yang mungkin memiliki prioritas klinis berbeda.

5. Evaluation

Tahap ini berfokus pada analisis dan evaluasi hasil pelatihan untuk menentukan apakah model telah memenuhi kebutuhan penelitian, yaitu memperoleh performa dan akurasi yang baik untuk implementasi pada mobile device.

6. Deployment

Pada tahap ini model diimplementasikan ke perangkat mobile (*Android*) melalui pembuatan aplikasi sederhana, sebagai pengujian keberhasilan penelitian sekaligus pencapaian tujuan utama penelitian.

F. Teknik dan Analisis Data

Desain sistem merupakan proses perancangan struktur dan komponen sistem yang digunakan untuk menjawab rumusan masalah penelitian, yaitu bagaimana meningkatkan performa model klasifikasi citra *Melanoma* pada dataset *HAM10000* dengan menerapkan teknik penyeimbangan data dan optimasi model untuk implementasi pada perangkat mobile. Desain sistem ini dibagi menjadi tiga bagian utama, yaitu Arsitektur Sistem, Desain Data, dan Desain Antarmuka Pengguna (*UI/UX*).

1. Desain Sistem

a. Arsitektur Sistem (*System Architecture*)

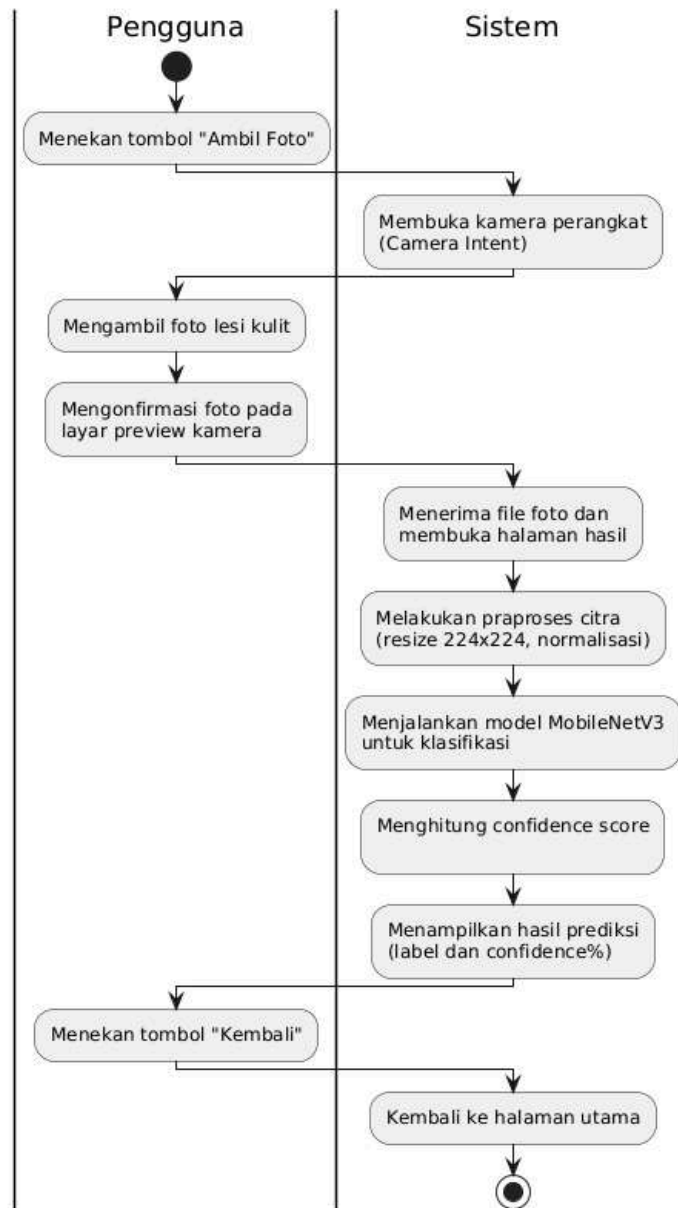
1) Diagram Use Case



Gambar 3. 3 Diagram Use Case

Diagram *use case* pada Gambar 3.3 mengilustrasikan alur interaksi langsung antara Pengguna dan sistem melalui tiga tahapan berurutan, yaitu pengambilan foto lesi kulit, pemrosesan otomatis oleh model *MobileNetV3*, serta penampilan hasil klasifikasi.. Walaupun proses klasifikasi berjalan otomatis di belakang layar tanpa campur tangan aktif pengguna, fase ini tetap dimasukkan sebagai *use case* utama karena posisinya sangat vital sebagai proses inti yang menjembatani aksi pengguna dengan hasil yang didapatkan, di mana rincian mekanismenya akan diuraikan lebih lanjut melalui diagram aktivitas dan sekuensial.

2) Diagram Aktivitas



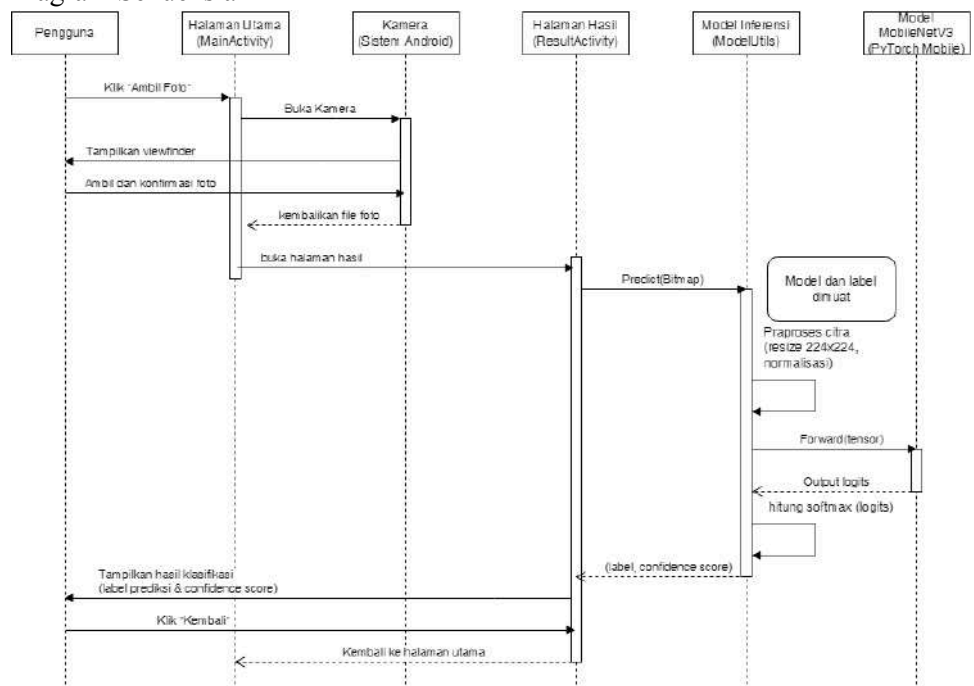
Gambar 3. 4 Diagram Aktivitas

Diagram aktivitas pada Gambar 3.4 membagi aliran proses ke dalam dua *swimlane* (Pengguna dan Sistem) untuk mempertegas batas tanggung jawab, dengan tetap mematuhi standar notasi UML seperti penggunaan *initial node*, *final node*, dan *rounded rectangle* pada setiap langkahnya. Proses bermula ketika Pengguna menekan tombol "Ambil Foto", yang langsung direspons Sistem dengan memanggil kamera perangkat melalui *Camera Intent*. Menariknya, meskipun tampilan layar beralih ke aplikasi kamera bawaan *Android*, tahapan memotret dan mengonfirmasi gambar tetap diletakkan pada jalur Pengguna karena fokus diagram ini adalah siapa pelaku aksinya, bukan antarmuka siapa yang dipakai. Logika pemisahan inilah yang menjadi alasan mengapa

pada dokumentasi Bab IV, tangkapan layar untuk proses konfirmasi menampilkan desain kamera bawaan sistem operasi alih-alih layar kustom dari aplikasi.

Setelah pengguna memberikan konfirmasi, kendali proses dikembalikan sepenuhnya kepada Sistem yang langsung menerima *file* foto dan membuka halaman hasil. Di balik layar, Sistem secara otomatis melakukan pra-pemrosesan citra, yaitu menyesuaikan ukuran gambar menjadi 224x224 piksel dan menormalisasinya menggunakan nilai *mean* serta standar deviasi *ImageNet* sesuai dengan rancangan data di Bab III. Citra yang sudah siap tersebut kemudian diproses oleh model *MobileNetV3* untuk diklasifikasikan dan dihitung *confidence score*-nya. Setelah itu, barulah Sistem menampilkan label prediksi akhir—apakah itu *Melanoma* atau *Nevus*—lengkap dengan persentase probabilitasnya ke layar pengguna. Seluruh rangkaian proses ini dinyatakan selesai (mencapai *final node*) saat Pengguna menekan tombol "Kembali" dan Sistem mengarahkan layar kembali ke halaman utama.

3) Diagram Sekuensial



Gambar 3.5 Diagram sekuensial

Diagram sekuensial pada gambar 3.5 ini menggambarkan interaksi antar enam objek dalam urutan waktu, yaitu Pengguna, Halaman Utama, Kamera, Halaman Hasil, Model Inferensi, dan Model *MobileNetV3*. Setiap kotak pada bagian atas merepresentasikan satu objek, dan garis putus-putus vertikal yang turun dari setiap objek disebut lifeline, yang merepresentasikan keberadaan objek tersebut selama interaksi berlangsung. Kotak persegi panjang sempit pada lifeline disebut

activation bar, yang merepresentasikan periode ketika suatu objek sedang aktif memproses sesuatu.

Penting untuk dijelaskan bahwa penamaan objek pada diagram sekuensial ini, yaitu Halaman Utama, Halaman Hasil, dan Model Inferensi, berbeda dengan nama kelas pada diagram kelas, yaitu *MainActivity*, *ResultActivity*, dan *ModelUtils*. Hal ini bukan inkonsistensi, melainkan konvensi UML yang valid: diagram kelas menampilkan blueprint atau definisi kelas sesuai kode program, sedangkan diagram sekuensial menampilkan *instance* atau objek dalam interaksi, yang dapat diberi nama berdasarkan peran fungsionalnya agar lebih mudah dipahami pembaca. Halaman Utama adalah instance dari kelas *MainActivity*, Halaman Hasil adalah instance dari kelas *ResultActivity*, dan Model Inferensi adalah instance dari objek *ModelUtils*.

Alur dimulai ketika Pengguna mengklik Ambil Foto pada Halaman Utama, yang kemudian mengirim pesan synchronous (digambarkan dengan anak panah bergaris penuh) ke objek Kamera untuk membuka kamera. Pesan Tampilkan viewfinder dan Ambil dan konfirmasi foto digambarkan melintasi lifeline Halaman Utama menuju Pengguna, yang merepresentasikan bahwa selama proses ini, antarmuka kamera berinteraksi langsung dengan Pengguna, sementara Halaman Utama berada dalam status menunggu. Setelah foto diambil dan dikonfirmasi, objek Kamera mengirimkan pesan balik (digambarkan dengan anak panah bergaris putus-putus, disebut return message) berupa file foto kepada Halaman Utama.

Halaman Utama kemudian membuka Halaman Hasil, yang langsung memanggil *Predict(Bitmap)* pada Model Inferensi. Catatan bertuliskan Model dan label dimuat pada bagian ini menjelaskan bahwa file model (.ptl) dan daftar label kelas (mel dan nv) dimuat ke memori hanya sekali, yaitu pada pemanggilan pertama (lazy loading), bukan setiap kali prediksi dilakukan, dengan tujuan efisiensi penggunaan memori dan waktu.

Selanjutnya Model Inferensi melakukan dua self-message, yaitu pesan yang dikirim oleh suatu objek kepada dirinya sendiri, digambarkan sebagai anak panah kecil yang melengkung kembali ke lifeline yang sama. Self-message pertama adalah Praproses citra (resize 224x224, normalisasi), yang merepresentasikan pemanggilan *method* preprocess() secara internal untuk mengubah bitmap menjadi tensor, yaitu struktur data array multidimensi (berukuran 3x224x224 yang merepresentasikan tiga kanal warna RGB dengan resolusi 224x224 piksel) yang dapat diproses oleh *neural network*.

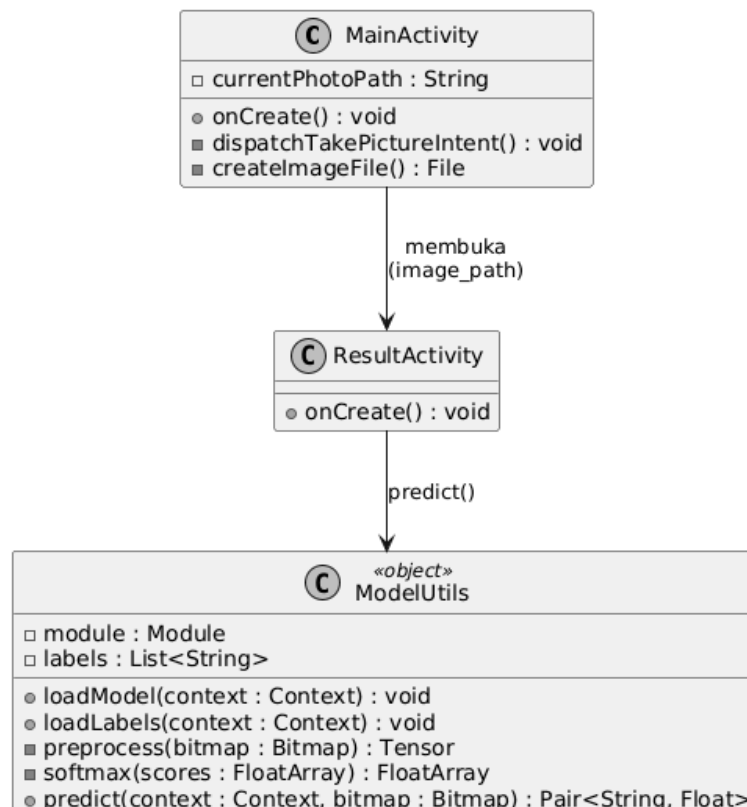
Pesan *Forward(tensor)* yang dikirim dari Model Inferensi ke Model *MobileNetV3* merepresentasikan pemanggilan *method* forward() milik PyTorch Mobile, yang menjalankan forward propagation, yaitu proses komputasi tensor input melalui seluruh lapisan jaringan *MobileNetV3* (sesuai konsep yang telah dijelaskan pada Bab II bagian Mekanisme Pelatihan Model) untuk menghasilkan output berupa logits. Logits adalah

nilai numerik mentah (raw scores) untuk setiap kelas, dalam kasus ini dua nilai yang merepresentasikan skor untuk kelas mel dan kelas nv, yang belum dinormalisasi menjadi probabilitas. Model *MobileNetV3* kemudian mengembalikan logits tersebut kepada Model Inferensi melalui return message Output logits.

Model Inferensi selanjutnya melakukan self-message kedua, yaitu hitung softmax(logits), yang mengubah logits menjadi nilai probabilitas yang berjumlah 1 untuk kedua kelas, sehingga dapat diinterpretasikan sebagai confidence score. Hasil akhir berupa pasangan (label, confidence score) dikembalikan kepada Halaman Hasil, yang kemudian menampilkan hasil klasifikasi kepada Pengguna. Alur diakhiri ketika Pengguna mengklik Kembali, dan Halaman Hasil mengirimkan return message Kembali ke halaman utama kepada Halaman Utama.

Penanganan kesalahan, seperti file gambar tidak ditemukan atau bitmap gagal dimuat, tidak digambarkan pada diagram ini karena diagram sekuensial difokuskan pada alur utama sistem. Skenario kesalahan tersebut dijelaskan secara terpisah pada bagian implementasi dan pengujian di Bab IV.

b. Desain Database



Gambar 3. 6 Diagram kelas

Diagram kelas pada gambar 3.6 menggambarkan tiga kelas yang membentuk struktur aplikasi, yaitu *MainActivity*, *ResultActivity*, dan *ModelUtils*. Setiap kelas

39

29

digambarkan dengan kotak yang terbagi menjadi tiga kompartemen sesuai konvensi UML, yaitu kompartemen nama kelas pada bagian atas, kompartemen atribut pada bagian tengah, dan kompartemen *method* pada bagian bawah. Simbol lingkaran dan kotak kecil yang berada di sebelah kiri setiap atribut dan *method* merepresentasikan tingkat visibilitas (access modifier) pada bahasa Kotlin mengikuti notasi standar UML, di mana simbol lingkaran merepresentasikan visibilitas public (anggota kelas yang dapat diakses dari luar kelas), sedangkan simbol kotak merepresentasikan visibilitas *private* (anggota kelas yang hanya dapat diakses dari dalam kelas itu sendiri). Relasi antar kelas digambarkan dengan anak panah bergaris penuh berkepala panah terbuka, yang dalam UML merepresentasikan relasi *dependency* atau *association*, yaitu satu kelas menggunakan atau berinteraksi dengan kelas lain, bukan relasi inheritance (kepala panah segitiga kosong) maupun composition atau aggregation (simbol diamond).

1) *MainActivity*

Kelas *MainActivity* merupakan halaman utama aplikasi yang bertanggung jawab menangani interaksi awal pengguna dan pengambilan foto melalui kamera. Atribut *currentPhotoPath* bersimbol kotak (*private*) karena nilai *path* foto hanya digunakan secara internal oleh kelas ini untuk diteruskan ke Activity selanjutnya, tidak diakses langsung oleh kelas lain. *Method* *onCreate()* bersimbol lingkaran (public) karena merupakan *lifecycle callback* yang dipanggil secara otomatis oleh sistem operasi *Android* setiap kali Activity dibuat, sehingga harus bersifat public agar dapat diakses oleh framework *Android* dari luar kelas. *Method* *dispatchTakePictureIntent()* dan *createImageFile()* bersimbol kotak (*private*) karena keduanya merupakan *method* bantuan (*helper method*) yang hanya dipanggil dari dalam *method* *onCreate()* pada kelas yang sama, dan tidak perlu diakses dari luar. *MainActivity* memiliki relasi menuju *ResultActivity* dengan label membuka (*image_path*), menunjukkan bahwa *MainActivity* memulai (*start*) *ResultActivity* sambil mengirimkan data *path* gambar hasil pengambilan foto.

2) *ResultActivity*

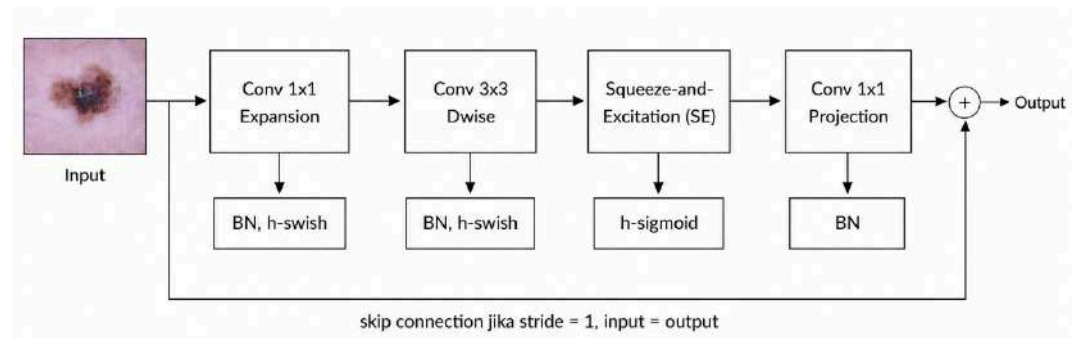
Kelas *ResultActivity* merupakan halaman yang menampilkan hasil klasifikasi kepada pengguna setelah proses inferensi selesai. *Method* *onCreate()* bersimbol lingkaran (publik) dengan alasan yang sama seperti pada *MainActivity*, yaitu sebagai *lifecycle callback* yang dipanggil oleh sistem *Android*, sehingga harus dapat diakses dari luar kelas oleh framework. *ResultActivity* memiliki relasi menuju *ModelUtils* dengan label *predict()*, menunjukkan bahwa *ResultActivity* memanggil *method* *predict()* milik *ModelUtils* untuk memperoleh hasil klasifikasi berupa label dan confidence score.

3) *ModelUtils*

Kelas *ModelUtils* bertanggung jawab memuat model machine learning dan melakukan seluruh proses inferensi, mulai dari praproses citra hingga penghitungan confidence score. Label *stereotype* <<object>> yang berada di atas nama kelas menunjukkan bahwa *ModelUtils* diimplementasikan menggunakan deklarasi *object* pada Kotlin, bukan class biasa, yang berarti *ModelUtils* merupakan singleton, yaitu hanya terdapat satu instance dari *ModelUtils* yang digunakan di seluruh aplikasi. Atribut *module* dan *labels* bersimbol kotak (*private*) karena

keduanya adalah state internal yang hanya dikelola oleh ModelUtils sendiri. Atribut module menyimpan model *MobileNetV3* yang telah dimuat ke memori, dan atribut labels menyimpan daftar nama kelas (mel dan nv). Method *loadModel()*, *loadLabels()*, dan *predict()* bersimbol lingkaran (public) karena merupakan antarmuka (interface) yang dipanggil oleh kelas lain, dalam hal ini *ResultActivity*. Method *preprocess()* dan *softmax()* bersimbol kotak (*private*) karena merupakan langkah internal yang hanya dipanggil dari dalam *method predict()* pada kelas yang sama, sebagaimana terlihat sebagai self-message pada diagram sekuensial. Implementasi sebagai singleton bertujuan agar model machine learning yang telah dimuat ke memori dapat digunakan kembali untuk setiap prediksi tanpa perlu memuat ulang dari file setiap kali, sehingga lebih efisien dari segi waktu dan penggunaan memori.

c. Desain Arsitektur *MobileNetV3*



Gambar 3. 6 Gambar arsitektur *MobileNetV3*

Gambar 3.6 mengilustrasikan struktur satu *bottleneck block* pada *MobileNetV3*, yang menjadi unit pembangun utama arsitektur ini. Blok ini terdiri dari beberapa komponen yang bekerja secara berurutan, masing-masing memiliki peran tersendiri. Input yang masuk ke blok ini adalah *feature map* bertipe tensor tiga dimensi berukuran $H \times W \times C$, di mana H dan W adalah dimensi spasial sedangkan C adalah jumlah kanal. Pada lapisan awal jaringan, input berasal dari konvolusi pembuka, sedangkan pada lapisan berikutnya input merupakan keluaran dari blok sebelumnya.

Komponen pertama adalah *Conv 1x1 Expansion*, yaitu konvolusi *pointwise* yang bertugas memperluas jumlah kanal ke dimensi yang lebih besar. Tujuannya adalah memastikan proses *filtering* pada lapisan berikutnya terjadi di ruang fitur yang lebih kaya, sehingga kemampuan model dalam menangkap pola menjadi lebih ekspresif. Lapisan ini tidak mengubah dimensi spasial, hanya jumlah kanalnya.

Setelah konvolusi ekspansi, keluaran diproses oleh *Batch Normalization* (BN) yang berfungsi menormalisasi distribusi aktivasi antar *mini-batch* selama pelatihan, menstabilkan proses konvergensi, dan mengurangi sensitivitas terhadap inisialisasi bobot. BN kemudian diikuti fungsi aktivasi *h-swish*, yaitu aproksimasi efisien dari fungsi *swish* yang didefinisikan sebagai $h-swish(x) = x \cdot$

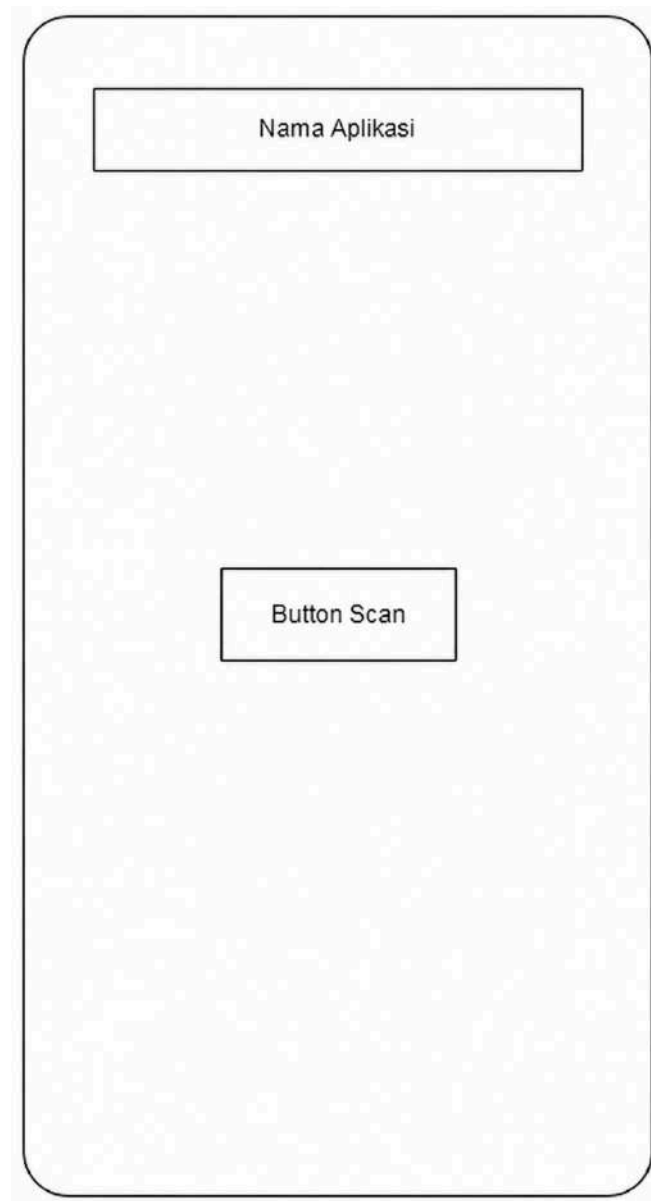
$\text{ReLU6}(x+3) / 6$. Fungsi ini dipilih karena *swish* asli yang melibatkan *sigmoid* terlalu mahal secara komputasi untuk perangkat *mobile*.

Selanjutnya adalah $\text{Conv } 3 \times 3$ *Depthwise*, yaitu konvolusi yang memproses setiap kanal secara terpisah menggunakan satu *filter* per kanal. Pendekatan ini menangkap fitur spasial seperti tepi dan tekstur dengan biaya komputasi yang jauh lebih rendah dibandingkan konvolusi standar. Keluarannya kembali dinormalisasi dengan BN lalu diaktivasi dengan *h-swish*.

Berikutnya adalah modul Squeeze-and-Excitation (SE), sebuah mekanisme *attention* berbasis kanal yang diletakkan setelah lapisan *depthwise*. Modul ini secara eksplisit memodelkan ketergantungan antar-kanal: pertama dengan meringkas seluruh informasi spasial menjadi satu vektor (*squeeze*), kemudian memberikan bobot kepentingan pada setiap kanal (*excitation*). Dengan cara ini, model belajar untuk memperkuat kanal yang relevan dan menekan yang tidak relevan. Pada bagian akhir modul SE digunakan fungsi aktivasi *h-sigmoid*, didefinisikan sebagai $\text{h-sigmoid}(x) = \text{ReLU6}(x+3) / 6$, yang menghasilkan nilai antara 0 dan 1 sebagai bobot kanal. Fungsi ini merupakan aproksimasi *piecewise linear* dari *sigmoid* standar yang lebih efisien karena tidak melibatkan operasi eksponensial.

Komponen terakhir dalam alur utama adalah $\text{Conv } 1 \times 1$ *Projection*, yang bertugas mengompresi kembali jumlah kanal ke ukuran lebih kecil sesuai dimensi keluaran yang diinginkan. Lapisan ini bersifat linear (tanpa fungsi aktivasi) dan diikuti BN untuk menjaga stabilitas sebelum hasil dijumlahkan dengan *skip connection*. *Skip connection* menghubungkan input blok langsung ke output, namun hanya aktif apabila $\text{stride} = 1$ dan dimensi input sama dengan dimensi output. Koneksi ini memberi jalur langsung bagi gradien selama *backpropagation*, mengurangi risiko *vanishing gradient*, dan memudahkan pelatihan jaringan yang lebih dalam.

d. Desain Antarmuka Pengguna (*UI/UX Design*)



Gambar 3. 7 Desain Aplikasi

Desain Antarmuka Pengguna (UI) Rancangan antarmuka aplikasi *mobile* pada Gambar 3.8 didesain secara minimalis untuk memudahkan interaksi pengguna. UI aplikasi menampilkan judul aplikasi dan tiga tombol navigasi utama. Tombol “Scan” berfungsi mengaktifkan kamera perangkat untuk melakukan klasifikasi lesi kulit secara *real-time*, sedangkan tombol “Upload” memungkinkan pengguna untuk mengunggah citra lesi yang tersimpan di galeri perangkat untuk dianalisis. Selain itu, terdapat tombol “Riwayat” yang menyediakan informasi detail riwayat prediksi yang telah tersimpan di metadata.

2. Simulasi Proses Penyelesaian Masalah

a. Data Input

1) Pemilihan Label Kelas (*Class Filtering*)

Dataset *HAM10000* terdiri dari tujuh kelas diagnosis lesi kulit. Penelitian ini difokuskan pada dua kelas yang paling relevan dengan tujuan studi, yaitu *Melanocytic Nevi (nv)* dan (*mel*).

Jumlah data awal per kelas:

- *nv* = 6.705 citra
- *mel* = 1.113 citra
- kelas lain (*akiec*, *bcc*, *bkl*, *df*, *vasc*) = 2.197 citra

Maka jumlah data yang dipakai:

$$N_{pakai} = N_{nv} + N_{mel} = 6705 + 1113 = 7818$$

Data di luar dua kelas tersebut dikeluarkan dari proses selanjutnya.

2) Pemeriksaan dan Pembersihan Data (*Data Cleaning*)

Setelah kelas ditentukan, dilakukan pemeriksaan terhadap kelengkapan informasi dan kesesuaian data.

Contoh perhitungan manual:

Jika dari 7.818 baris metadata ditemukan 10 baris dengan *image_id* kosong dan 5 citra rusak, maka:

$$N_{bersih} = 7818 - 10 - 5 = 7803$$

Hanya 7.803 sampel inilah yang dipakai untuk tahap berikutnya.

3) Penyesuaian Ukuran Citra (*Resizing*)

Citra dalam dataset memiliki ukuran asli yang beragam. Seluruh citra akan diubah ke ukuran seragam sesuai kebutuhan arsitektur *CNN* yang digunakan, yaitu 224×224 piksel. Penyeragaman ini memudahkan pengolahan dan menyesuaikan data dengan standar input model.

Contoh perhitungan manual:

Satu citra RGB berukuran 224×224, secara matematis jumlah piksel dalam sebuah citra digital ditentukan oleh dimensi spasialnya, yaitu tinggi (*height*) dan lebar (*width*) citra. Rumus umum untuk menghitung total piksel (N_{pixel}) adalah:

$$N_{pixel} = H \times W \dots$$

Dimana:

- *H* = Tinggi citra (*height*)
- *W* = Lebar citra (*width*)

Berdasarkan arsitektur yang digunakan, dimensi target adalah *H* = 224 dan *W* = 224. Maka perhitungannya adalah:

$$N_{pixel} = 224 \times 224 = 50176$$

Selanjutnya, karena citra yang digunakan adalah citra berwarna (*RGB*), maka terdapat kanal warna (*channels*). Total elemen data input (N_{elemen}) dihitung dengan mengalikan jumlah piksel dengan jumlah kanal warna (*C*):

$$N_{elemen} = N_{pixel} \times C$$

Diketahui jumlah kanal warna $C = 3$ (Red, Green, Blue), maka:

$$N_{elemen} = 50176 \times 3 = 150528$$

4) Normalisasi Nilai Piksel (*Normalization*)

Nilai piksel yang semula berada pada rentang 0–255 dinormalisasi, kemudian distandarisasi dengan mean dan standar deviasi tertentu. Langkah ini bertujuan menjaga kestabilan pelatihan dan membantu mempercepat konvergensi model. pada penelitian ini digunakan nilai mean (0,485; 0,456; 0,406) dan standar deviasi (0,229; 0,224; 0,225) yang merupakan statistik dataset *ImageNet* dan umum digunakan pada model yang telah melalui proses pretraining menggunakan *ImageNet* (Deng et al., 2009).

Rumus normalisasi per kanal:

$$x_{norm} = \frac{x}{255}$$

Jika setelah itu digunakan mean dan standar deviasi ($\mu = 0,485$, $\sigma = 0,229$ untuk kanal merah), maka:

$$x_{std} = \frac{x_{norm} - \mu}{\sigma}$$

Contoh perhitungan manual:

Misal nilai piksel merah = 128:

$$x_{norm} = \frac{128}{255} = 0,502$$

$$x_{std} = \frac{0,502 - 0,485}{0,229} \approx 0,074$$

5) Augmentasi Data (*Data Augmentation*)

Untuk meningkatkan keragaman sampel pada data latih, digunakan beberapa teknik seperti rotasi dan flipping. Augmentasi membantu memperluas variasi visual dan mendukung peningkatan kemampuan generalisasi model, terutama pada *Imbalance data*.

Contoh perhitungan manual:

Misal dari kelas tersedia 1.113 citra. Jika direncanakan dibuat 1 augmentasi tambahan per citra (misal satu rotasi 15°):

$$N_{baru} = 1113 \text{ (asli)} + (\text{Augmentasi}) = 2226$$

Augmentasi dapat dilakukan lebih dari satu transformasi tergantung kebutuhan.

6) *Encoding Label* dan Pembagian Dataset (*Label Encoding & Data Splitting*)

Label kelas dikodekan dalam bentuk numerik untuk keperluan pelatihan model. Dataset kemudian direncanakan untuk dibagi menjadi data latih,

71

validasi, dan uji dengan rasio 70:20:10 menggunakan pendekatan *stratified split* agar distribusi kelas tetap terjaga.

Contoh perhitungan manual:

$$\text{Dengan } N_{\text{bersih}} = 7803$$

$$N_{\text{train}} = 0,70 \times 7803 = 5462$$

$$N_{\text{val}} = 0,20 \times 7803 = 1560$$

$$N_{\text{test}} = 0,10 \times 7803 = 780$$

Pembulatan dilakukan sehingga totalnya tetap sama dengan jumlah data bersih dan proporsi tiap kelas dijaga menggunakan *stratified split*.

7) Penyeimbangan Data (*Handling Imbalance*)

– Class weight

Untuk menangani ketidakseimbangan (imbalance), digunakan class weighting berdasarkan rasio kelas.

Contoh perhitungan manual:

Jika pada data latih terdapat 4.693 citra nv dan 779 citra mel, maka:

$$\text{Rasio} = \frac{N_{\text{nv}}}{N_{\text{mel}}} = \frac{4693}{779} = 6,02$$

Nilai rasio ini dapat digunakan, sebagai dasar pembobotan kelas (*class weight*).

106

– SMOTE (*Synthetic Minority Over-sampling Technique*)

SMOTE diterapkan pada indeks data latih menggunakan library *imbalanced-learn* (*RandomOverSampler*). Sampel sintesis kelas *Melanoma* dibangkitkan melalui interpolasi linier antara sampel yang ada dan tetangga terdekatnya: $x_{\text{baru}} = x_i + \lambda(x_{nn} - x_i)$, di mana $\lambda \in [0,1]$ acak. SMOTE hanya diterapkan pada data latih sebelum proses pelatihan dimulai.

167

– Weighted Random Sampling (WRS)

WRS diterapkan menggunakan *WeightedRandomSampler* dari *PyTorch*. Setiap sampel data latih diberi bobot sampling sebesar $\frac{1}{n(\text{kelas sampel})}$, di mana n adalah jumlah sampel kelas tersebut. Dengan demikian, kelas *Melanoma* (minoritas) memiliki probabilitas lebih tinggi untuk dipilih setiap batch tanpa menambah data secara fisik.

– Kombinasi (SMOTE + Class Weight)

Metode Kombinasi menerapkan SMOTE dan Class Weight secara bersamaan. SMOTE menyeimbangkan distribusi data pada level dataset, sementara Class Weight memberikan penalti tambahan pada fungsi loss. Tujuan pendekatan ganda ini adalah memaksimalkan perhatian model terhadap kelas *Melanoma* dari dua sisi sekaligus.

– Focal Loss

2

Focal Loss mengganti CrossEntropyLoss standar dengan fungsi loss dinamis: $L(p_t) = -\alpha_t \cdot (1 - p_t)^\gamma \cdot \log(p_t)$. Dalam penelitian ini $\alpha = 0,8576$ (proporsional terhadap distribusi kelas *Nevus* di data latih) dan $\gamma=2$. Parameter γ menurunkan bobot sampel yang mudah diklasifikasikan sehingga model terfokus pada kasus-kasus sulit, termasuk sampel *Melanoma* yang ambigu.

8) Penentuan threshold Keputusan

Keluaran model *MobileNetV3*-Small pada setiap inferensi berupa dua nilai probabilitas, yaitu $P(Nevus)$ dan $P(Melanoma)$, yang keduanya berjumlah 1,0. Secara default, keputusan kelas diambil berdasarkan probabilitas tertinggi, yang secara matematika setara dengan threshold 0,5 sampel diklasifikasikan sebagai *Melanoma* apabila $P(Melanoma) > 0,5$. Nilai threshold ini dapat digeser untuk menyesuaikan keseimbangan antara *Recall* dan *Precision* sesuai kebutuhan konteks penggunaan. Pada penelitian ini, dilakukan threshold tuning sebagai analisis tambahan untuk mengetahui nilai threshold yang menghasilkan *Precision* lebih tinggi dengan tetap mempertahankan *Recall Melanoma* pada tingkat yang memadai. Prosedur penentuan threshold ditetapkan sebagai berikut:

- Model S4 dijalankan pada validation set untuk menghasilkan nilai $P(Melanoma)$ bagi setiap sampel.
- Nilai threshold disapu (sweep) dari 0,05 hingga 0,95 dengan langkah 0,01.
- Dari seluruh nilai threshold yang menghasilkan *Recall Melanoma* $\geq 0,80$ pada validation set, dipilih nilai yang menghasilkan *Precision* tertinggi.
- Nilai batas minimum *Recall* sebesar 0,80 ditetapkan oleh peneliti berdasarkan pertimbangan bahwa tingkat deteksi di bawah angka tersebut dinilai tidak memadai untuk tujuan skrining awal *Melanoma*.
- Threshold terpilih kemudian dievaluasi pada test set yang sepenuhnya terpisah untuk mendapatkan estimasi performa yang tidak bias.

b. Proses pengolahan data

Proses pengolahan data pada penelitian ini menggunakan Convolutional Neural Network (CNN) berbasis arsitektur *MobileNetV3* untuk melakukan klasifikasi citra lesi kulit pada dataset *HAM10000*. CNN merupakan salah satu algoritma deep learning yang sangat efektif untuk analisis citra karena kemampuannya mengekstraksi fitur spasial dan pola visual secara otomatis melalui operasi konvolusi.

Arsitektur *MobileNetV3*, yang dikembangkan oleh Howard et al. (2019), merupakan pengembangan dari *MobileNetV2* dengan tambahan squeeze-and-excitation modules serta hard-swish activation function untuk meningkatkan efisiensi dan akurasi. Model ini dirancang khusus agar ringan dan cepat, sehingga cocok untuk implementasi pada perangkat mobile.

Langkah-langkah Algoritma CNN (*MobileNetV3*)

1) Inisialisasi

82

28

48

65

Penelitian ini dirancang dalam empat skenario eksperimen berurutan (S1→S2→S3→S4) yang membentuk pipeline optimasi bertahap. Tabel 3.6 merangkum keseluruhan alur skenario tersebut.

Tabel 3. 6 Alur skenario eksperimen penelitian

Skenario	Tujuan	Konfigurasi	Output
S1	Baseline	Tanpa imbalance handling, backbone frozen	Acuan performa awal
S2	Memilih metode imbalance terbaik	5 metode: Class Weight, SMOTE, WRS, Kombinasi, Focal Loss (config identik)	Class Weight terpilih (<i>Recall</i> tertinggi yang valid)
S3	Optimasi hiperparameter otomatis	Optuna 20 trial, objective: $Recall \times \min(1.0, Precision/0.30)$	Konfigurasi Trial 9 terpilih
S4	Pelatihan final + threshold tuning	Parameter Trial 9 + unfreeze_all backbone (differential LR)	Model final, threshold optimal=0,67

Tahapan awal dimulai dengan inisialisasi parameter model dan konfigurasi pelatihan:

a. Arsitektur Model

Model dasar yang digunakan adalah *MobileNetV3 Small*, yang dimodifikasi pada bagian *fully connected layer* untuk menghasilkan DUA output neuron (nn.Linear(in_features, 2)), merepresentasikan logit kelas *Nevus* (nv) dan *Melanoma* (mel). Fungsi loss yang digunakan adalah CrossEntropyLoss yang secara internal menerapkan Softmax. Modifikasi ini berbeda dari pendekatan satu neuron + Sigmoid yang digunakan pada BCELoss. Dataset yang digunakan merupakan klasifikasi biner antara kelas (mel) dan (nv).

b. Bobot Awal

Model diinisialisasi menggunakan *pretrained weights* dari dataset *ImageNet*, sehingga model sudah memiliki pengetahuan dasar terhadap pola umum pada citra RGB.

c. Parameter Pelatihan

Konfigurasi pelatihan ditetapkan tetap (fixed) untuk Skenario 1 hingga Skenario 3 guna menjamin perbandingan yang terkontrol antar metode penanganan ketidakseimbangan data, yaitu optimizer Adam dengan *learning rate* 1×10^{-4} , batch size 32, maksimum 100 epoch, dan early stopping dengan patience 7 (pelatihan dihentikan otomatis apabila validation loss tidak menurun selama 7 epoch berturut-turut). Pada Skenario 4, konfigurasi ini disesuaikan berdasarkan hasil pencarian hiperparameter otomatis menggunakan Optuna (lihat Skenario 3), menghasilkan *learning rate* $4,811 \times 10^{-4}$ untuk lapisan classifier dan $4,811 \times 10^{-5}$ untuk backbone (differential *learning rate*), dengan batch size dan kriteria early stopping yang tetap sama.

2) Iterasi (Tahap Training)

Proses pelatihan model *CNN* dilakukan melalui forward propagation untuk menghasilkan prediksi dan menghitung nilai loss, dilanjutkan dengan backward propagation untuk menghitung gradien dan

memperbarui bobot jaringan. Setiap epoch terdiri dari dua fase utama: *forward propagation* dan *backward propagation*.

Simulasi perhitungan:

- Input (x): 1.0 (Fitur dari gambar)
- Target (y): 1.0 (Label: *Melanoma*)
- Bobot Awal (w): 0.5
- *Learning rate* (η): 0.1

1. Tahap *Forward Propagation*

Berdasarkan konsep *Deep Feedforward Networks* yang diadaptasi dari penelitian (Vakalopoulou et al., 2023), fungsi prediksi jaringan ($\hat{y}$) adalah komposisi dari fungsi transformasi setiap lapisan. Untuk satu lapisan sederhana, ini dihitung sebagai:

$$z = w \cdot x$$

Perhitungan:

$$z = 0.5 \times 1.0 = 0.5$$

Karena kasus ini adalah klasifikasi biner, output linear (z) diproses oleh model menjadi dua logit output. Untuk keperluan ilustrasi matematis simulasi ini, digunakan pendekatan satu neuron + Sigmoid. Pada implementasi aktual, model menggunakan DUA output neuron + CrossEntropyLoss (Softmax), konsisten dengan modifikasi arsitektur yang dijelaskan di atas. (Fungsi Aktivasi Sigmoid):

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Perhitungan:

$$P = \frac{1}{1+e^{-0.5}}$$

$$P = \frac{1}{1+0.6065}$$

$$P \approx 0.6225$$

Model memprediksi 62,25% kemungkinan input adalah *Melanoma*.

Untuk mengukur kesalahan prediksi, kita membandingkan hasil prediksi ($P=0.6225$) dengan label target ($y=1.0$). Rumus yang digunakan adalah *Cross-Entropy Loss* yang tercantum pada Persamaan *Forward Propagation*:

$$J(W) = - \sum_{i=1}^n \log(P(y = y^{(i)} | x = x^{(i)}; W))$$

Untuk satu data tunggal, rumus disederhanakan menjadi:

$$J(W) = - [y \cdot \log(P) + (1 - y) \cdot \log(1 - P)]$$

Perhitungan:

$$J(W) = - [1.0 \cdot (-0.474) + 0]$$

$$J(W) \approx 0.474$$

Output Prediksi (P): 0.6225

Total Error (J): 0.474

2. Tahap *Backward Propagation*

Setelah mendapatkan nilai loss (0.474) dari tahap sebelumnya, kita perlu mengetahui seberapa besar kontribusi bobot w terhadap kesalahan tersebut. Kita menggunakan metode Aturan Rantai (*Backward Propagation*).

$$\frac{\partial J(W)}{\partial w} = \frac{\partial J(W)}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial z} \times \frac{\partial z}{\partial w}$$

$$\frac{\partial J(W)}{\partial w} = \frac{\hat{y} - y}{\hat{y}(1 - \hat{y})} \times \hat{y}(1 - \hat{y}) \times x$$

$$\frac{\partial J(W)}{\partial w} = (P - y) \cdot x$$

Gunakan hasil penyederhanaan persamaan

$$\text{Gradien} = (P - y) \cdot x$$

$$\text{Gradien} = (0.6225 - 1.0) \times 1.0$$

$$\text{Gradien} = -0.3775$$

Perhitungan:

- $P = 0.6225$
- $y = 1.0$ (Target)
- $X = 1.0$ (Input)

Nilai negatif menunjukkan bahwa bobot harus dinaikkan (digeser ke arah positif) untuk mengurangi error.

3. Tahap Optimasi

Setelah memperoleh nilai gradien (-0,3775) dari tahap *backward propagation*, bobot model diperbarui menggunakan algoritma Adam (Kingma & Ba, 2015). Berbeda dengan *Gradient Descent* murni yang memperbarui bobot langsung dari gradien mentah, Adam terlebih dahulu menghitung dua estimasi momen gradien secara adaptif sebelum melakukan pembaruan. Hal ini membuat Adam lebih stabil dan cepat konvergen, terutama pada kasus data tidak seimbang seperti pada penelitian ini.

Parameter:

- w_t (bobot lama) = 0,5
- g_t (gradien) = -0,3775
- α (*learning rate*) = 0,1
- $\beta_1 = 0,9$; $\beta_2 = 0,999$; $\varepsilon = 10^{-8}$
- $t = 1$ (iterasi pertama, $m_0 = 0$, $v_0 = 0$)

Langkah 1 Update momen pertama:

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

$$m_1 = 0.9 \times 0 + (1 - 0.9) \times (-0.3775)$$

$$m_1 = 0 + 0.1 \times (-0.3775)$$

$$m_1 = -0.03775$$

Langkah 2 Update momen kedua:

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

$$v_1 = 0.999 \times 0 + (1 - 0.999) \times (-0.3775)^2$$

$$v_1 = 0 + 0.001 \times 0.1425$$

$$v_1 = 0.0001425$$

Langkah 3 Koreksi bias momen pertama:

$$m_1^* = \frac{-0.03775}{1-0.9} m_1^* = \frac{-0.03775}{0.1} m_1^* = -0.3775$$

Langkah 4 Koreksi bias momen kedua:

$$v_1^* = \frac{0.0001425}{1-0.999} v_1^* = \frac{0.0001425}{0.001} v_1^* = 0.1425$$

Langkah 5 Pembaruan parameter:

$$W_1 = w_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}$$

$$W_1 = 0.5 - 0.1 \left(\frac{-0.3775}{\sqrt{0.1425 + 10^{-8}}} \right) W_1 = 0.5 - 0.1 \left(\frac{-0.3775}{0.3775} \right) W_1 = 0.5 - 0.1(-1.0000) W_1 = 0.5 + 0.1$$

Kesimpulan Akhir Simulasi Satu Iterasi:

- Sebelum Training: Bobot = 0.5 | Prediksi = 62.25% (Kurang yakin).
- Setelah 1 Iterasi: Bobot naik menjadi 0.6000.
- Efek: Kenaikan bobot ini akan membuat nilai z naik pada iterasi berikutnya, sehingga prediksi P akan mendekati target 1,0 dan nilai loss akan semakin kecil. Momen pertama $\hat{m}_1 = -0.3775$ menunjukkan arah update yang tepat (menaikkan bobot), sementara momen kedua $\hat{v}_1 = 0.1425$ bertindak sebagai pengendali besarnya langkah sehingga update tidak terlalu agresif.

3) Kriteria berhenti

Batas maksimal iterasi pelatihan ditetapkan sebanyak 20 *epoch*. Sebagai langkah mitigasi terhadap risiko *overfitting* dan untuk memastikan model berhenti pada titik performa optimal, penelitian ini mengimplementasikan fungsi *callback* berupa *Early Stopping*.

103

Mekanisme ini bekerja dengan cara menghentikan proses pelatihan secara otomatis apabila nilai *validation loss* (val_{loss}) tidak menunjukkan penurunan minimal sebesar 0,001 (min_{delta}) selama lima *epoch* berturut-turut (*patience* = 5). Pada saat pelatihan dihentikan, model akan secara otomatis memulihkan dan menggunakan konfigurasi bobot terbaik yang tercatat selama proses evaluasi (*restore_best_weights*).

c. Data Output dan Evaluasi

28

1) Data Output

Luaran (*output*) utama dari proses pengolahan data menggunakan model *Convolutional Neural Network* (CNN) berbasis *MobileNetV3* ini adalah hasil klasifikasi citra lesi kulit menjadi dua kategori, yaitu:

1. *Melanocytic Nevi* (nv) lesi kulit jinak (benign).
2. *Melanoma* (mel) lesi kulit ganas (malignant).

Bentuk keluaran yang dihasilkan oleh sistem dalam konteks hasil prediksi dalam bentuk antarmuka pengguna pada implementasi di aplikasi mobile berupa Citra masukan yang diunggah atau diambil melalui kamera dan Label hasil prediksi dan nilai kepercayaan (*confidence score*).

2) Evaluasi

Pada akhir setiap epoch, model diuji terhadap *validation set* untuk menghitung metrik:

66

1. *Accuracy*
2. *Recall*
3. *Precision*
4. *F1-score*
5. *AUC* (*Area Under the ROC Curve*)

Metrik AUC dan *Recall* menjadi fokus utama karena relevan dengan kasus medis (Hicks et al., 2022), di mana *false negative* (*Melanoma* yang tidak terdeteksi) harus diminimalkan. Pada metrik AUC, perhitungannya dilakukan menggunakan probabilitas keluaran model yang diperoleh setelah proses validasi pada setiap epoch. Dengan demikian, AUC tetap dihitung pada akhir epoch saat probabilitas prediksi tersedia.

Simulasi perhitungan manual:

Hasil data uji

76

- True Positive (TP) = 88
- False Positive (FP) = 12
- True Negative (TN) = 800
- False Negative (FN) = 20

Hitung metrik:

- *Accuracy*

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{88+800}{88+800+12+20} = \frac{888}{920} = 0,965 \text{ (96\%)}$$

- *Recall*

$$Recall = \frac{TP}{TP+FN} = \frac{88}{88+20} = \frac{88}{108} = 0,815 \text{ (81,5\%)}$$

- *Precision*

$$Precision = \frac{TP}{TP+FP} = \frac{88}{88+12} = \frac{88}{100} = 0,88 \text{ (88\%)}$$

$$- \text{ F1-score}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{0,88 \times 0,815}{0,88 + 0,815} = 0,846 \text{ (84\%)}$$

BAB IV HASIL DAN PEMBAHASAN

A. Hasil Penelitian

1. Implementasi Desain Sistem

a. Implementasi Lembar Kerja

1) Pelatihan Model

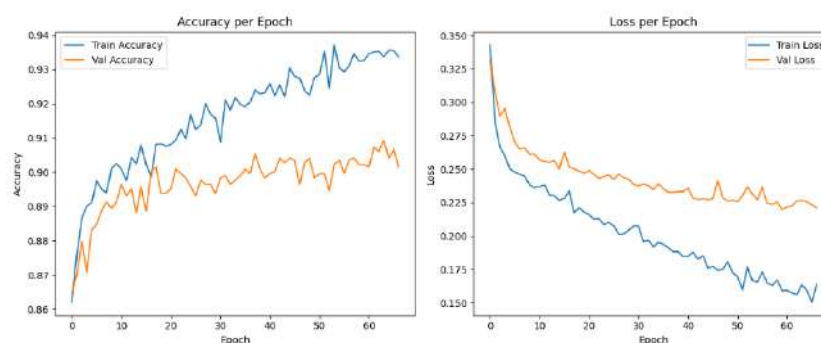
Pelatihan model diimplementasikan dalam sebelas notebook Jupyter yang masing-masing bersifat mandiri dan bertanggung jawab atas satu tahap eksperimen yang berbeda. Pendekatan satu notebook per skenario diterapkan untuk memastikan setiap eksperimen dapat direproduksi secara independen tanpa ketergantungan antar file. Tabel 4.1 menyajikan ringkasan seluruh notebook beserta fungsinya masing-masing.

Tabel 4.1 Daftar Notebook Pelatihan Model

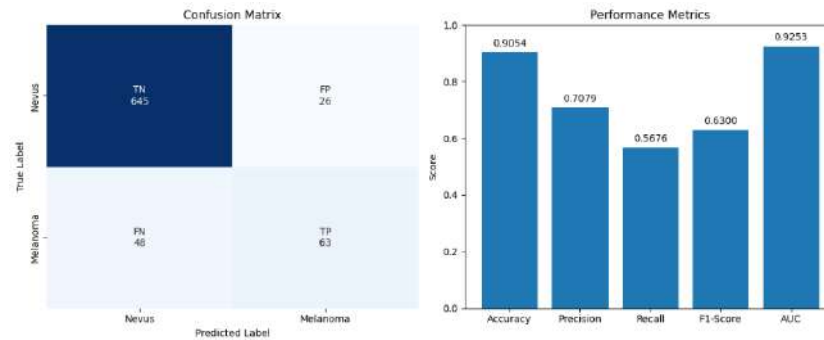
N o	Nama File	Fungsi
1	Skenario1_baseline.ipynb	Pelatihan model tanpa penanganan imbalance sebagai acuan performa awal
2	Skenario2_ClassWeight.ipynb	Pelatihan dengan Class Weight: bobot kelas minoritas diperbesar pada fungsi loss
3	Skenario2_SMOTE.ipynb	Pelatihan dengan SMOTE: oversampling sintetis pada data latih
4	Skenario2_WRS.ipynb	Pelatihan dengan Weighted Random Sampler: probabilitas sampling disesuaikan per kelas
5	Skenario2_Kombinasi.ipynb	Pelatihan dengan kombinasi SMOTE dan Class Weight secara bersamaan

6	Skenario2e_FocalLoss.ipynb	Pelatihan dengan Focal Loss: bobot sampel disesuaikan secara dinamis per iterasi
7	Skenario3_Optuna_Rerun.ipynb	Optimasi hiperparameter menggunakan Optuna (20 trial) berbasis class weight
8	Skenario4_FinalTraining.ipynb	Pelatihan ulang dengan parameter terbaik dari Skenario 3 Optuna dalam 100 trial
9	ThresholdTuning_Skenario4.ipynb	Pencarian threshold keputusan optimal untuk menyeimbangkan <i>Recall</i> dan <i>Precision</i>
10	benchmark_nonfungsional.ipynb	Pengukuran ukuran model, jumlah parameter, dan waktu inferensi pada CPU dan GPU

Skenario 1 merupakan tahap awal yang bertujuan menetapkan acuan performa awal (baseline) model *MobileNetV3-Small* pada dataset *HAM10000* tanpa penerapan strategi penanganan ketidakseimbangan data apapun. Backbone *MobileNetV3-Small* dibekukan (frozen) seluruhnya sehingga hanya lapisan classifier yang dilatih, menggunakan bobot pretrained *ImageNet* sebagai titik awal. Konfigurasi pelatihan yang digunakan adalah optimizer Adam dengan *learning rate* 1×10^{-4} , batch size 32, maksimum 100 epoch, dan early stopping dengan patience 7. Hasil pengujian pada test set menunjukkan bahwa tanpa penanganan ketidakseimbangan data, model sangat bias terhadap kelas mayoritas (*Nevus*). *Recall Melanoma* hanya mencapai 0,5676 dengan *Precision* 0,7079 dan AUC 0,9253. Artinya, dari 111 sampel *Melanoma* pada test set, sebanyak 48 kasus (FN) tidak berhasil dideteksi. Meskipun akurasi keseluruhan tergolong tinggi (90,54%), nilai ini menyesatkan dalam konteks klinis karena model cenderung memprediksi semua sampel sebagai *Nevus*. Hasil Skenario 1 ini menjadi acuan yang digunakan untuk mengevaluasi efektivitas setiap strategi penanganan imbalance pada skenario berikutnya.



Gambar 4. 1 kurva training validasi skenario 1



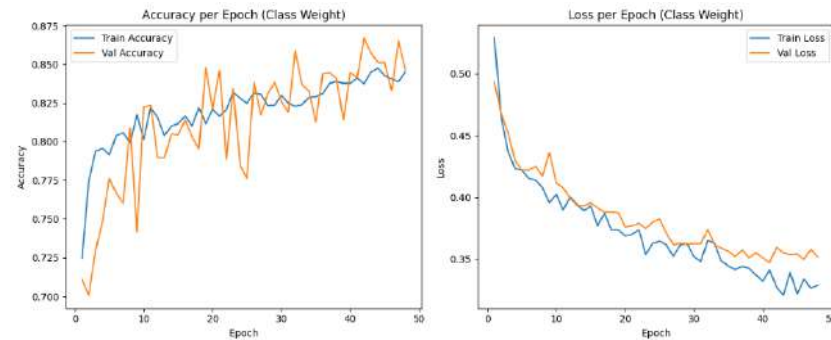
Gambar 4. 2 Confusion matrix skenario 1

Skenario 2 bertujuan membandingkan efektivitas lima strategi penanganan ketidakseimbangan data secara sistematis. Seluruh konfigurasi pelatihan diidentikkan dengan Skenario 1 optimizer Adam ($lr=1 \times 10^{-4}$), batch size 32, maksimum 100 epoch, early stopping patience 7, dan backbone frozen sehingga perbedaan performa yang terukur dapat sepenuhnya dikaitkan dengan strategi imbalance handling yang divariasikan, bukan karena perbedaan konfigurasi lainnya.

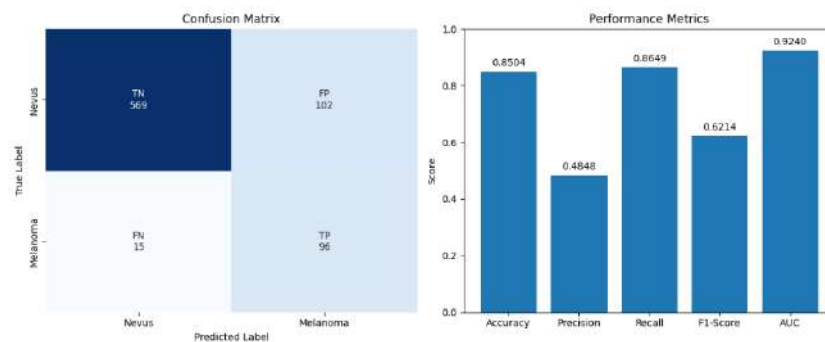
Tabel 4. 2 Perbandingan Metode Imbalance data

Metode	Recall	Precision	F1	AUC	FN	FP
Class Weight	0,8649	0,4848	0,6214	0,9240	15	102
SMOTE	0,7387	0,5816	0,6508	0,9318	29	59
Weighted Random Sampler	0,7477	0,5764	0,6510	0,9275	28	61
SMOTE + Class Weight	0,9099	0,4073	0,5627	0,9282	10	147
Focal Loss	0,8288	0,4894	0,6154	0,9239	19	96

Tabel 4.2 menunjukkan bahwa Kombinasi menghasilkan *Recall* tertinggi (0,9099) namun dengan *Precision* terendah (0,4073) dan FP terbanyak (147), sehingga tidak dipilih sebagai metode terbaik. Class Weight dipilih sebagai metode terbaik pada Skenario 2 karena menghasilkan *Recall* tertinggi yang valid (0,8649) tanpa collapse *Precision*. S4 Final dengan threshold default (0,5) menghasilkan *Recall* tertinggi keseluruhan (0,8829), sedangkan threshold 0,67 meningkatkan *Precision* menjadi 0,5610 dengan konsekuensi penurunan *Recall* menjadi 0,8288.

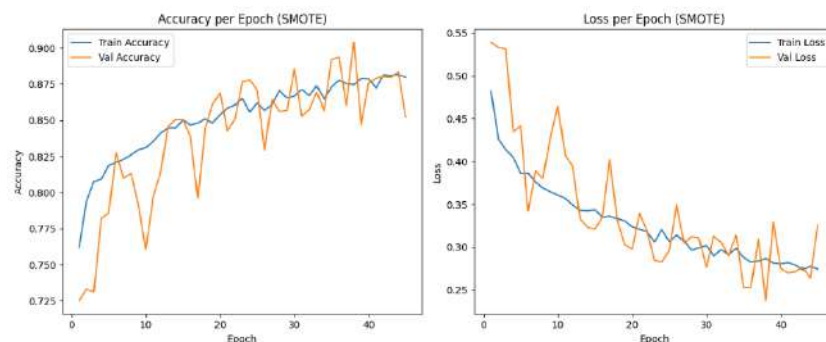


Gambar 4. 3 kurva training validasi skenario 2 class weight

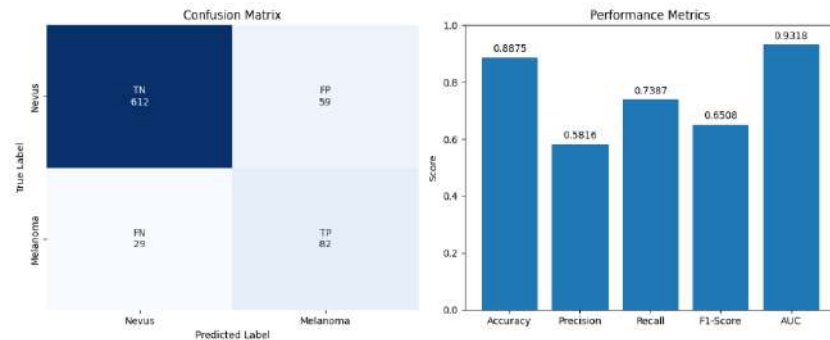


Gambar 4. 4 Confusion skenario 2 class weight

Metode Class Weight. Bobot kelas dihitung dari distribusi data latih secara terbalik proporsional ($w_{mel} \approx 6,02$, $w_{nv} = 1,0$) dan diterapkan pada CrossEntropyLoss. Metode ini memberikan penalti lebih besar pada kesalahan prediksi kelas minoritas (*Melanoma*) tanpa mengubah atau menambah data secara fisik. Hasil: *Recall* = 0,8649, *Precision* = 0,4848, *AUC* = 0,9240, *FN* = 15, *FP* = 102. Model berhasil mendeteksi 96 dari 111 *Melanoma* pada test set. Metode ini dipilih sebagai metode terbaik S2 karena menghasilkan *Recall* valid tertinggi tanpa mengalami *Precision* collapse.

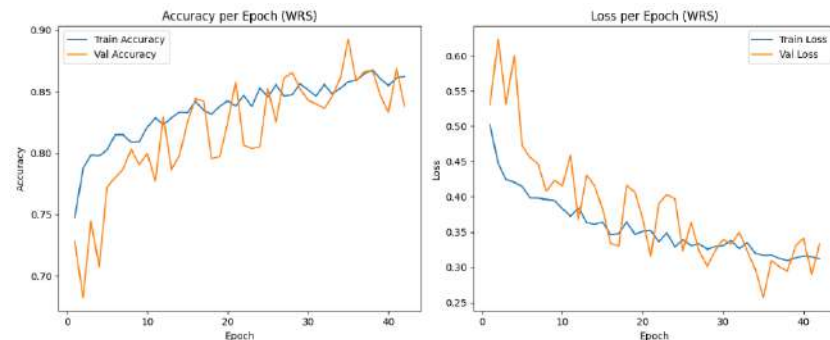


Gambar 4. 5 kurva training validasi skenario 2 SMOTE

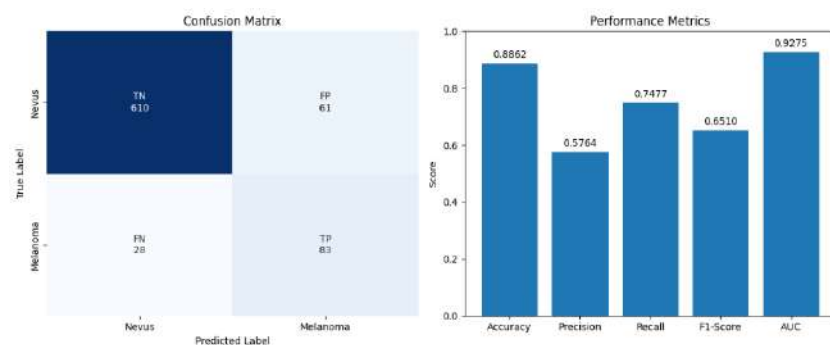


Gambar 4. 6 Confusion matrix skenario 2 SMOTE

Metode Synthetic Minority Over-sampling Technique (SMOTE) diterapkan pada data latih setelah proses split untuk menghindari data *leakage*. Sampel sintesis *Melanoma* dibangkitkan melalui interpolasi linier antar sampel yang ada sehingga distribusi kelas menjadi seimbang sebelum pelatihan dimulai. Hasil: $Recall = 0,7387$, $Precision = 0,5816$, $AUC = 0,9318$, $FN = 29$, $FP = 59$. SMOTE menghasilkan nilai $Precision$ tertinggi di antara semua metode Skenario 2 karena data sintesis memperluas ruang keputusan kelas *Melanoma*, namun recallnya lebih rendah dibandingkan Class Weight.

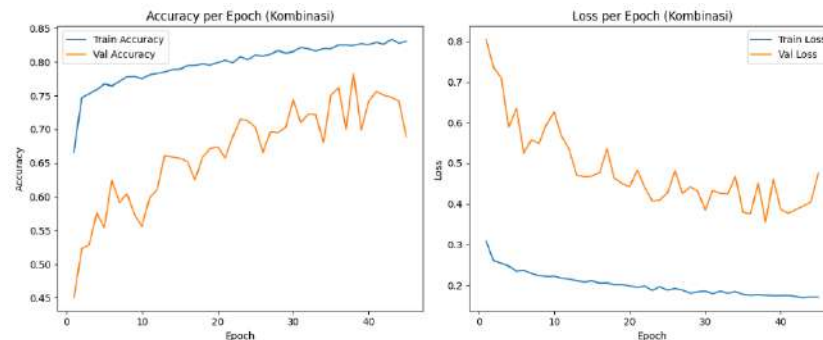


Gambar 4. 7 kurva training validasi skenario 2 Weighted Random Sampling

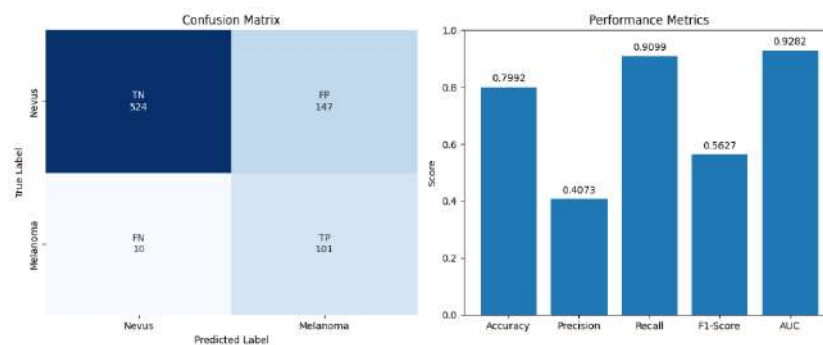


Gambar 4. 8 Confusion matrix skenario 2 Weighted Random Sampling

Metode Weighted Random Sampler (WRS). Setiap sampel data latih diberi bobot sampling sebesar $1/n(\text{kelas})$, sehingga kelas *Melanoma* memiliki probabilitas lebih tinggi untuk dipilih pada setiap batch tanpa menambah data secara fisik. Hasil: *Recall* = 0,7477, *Precision* = 0,5764, *AUC* = 0,9275, *FN* = 28, *FP* = 61. Performa WRS serupa dengan SMOTE, namun lebih hemat memori karena tidak memerlukan pembuatan data sintetis.

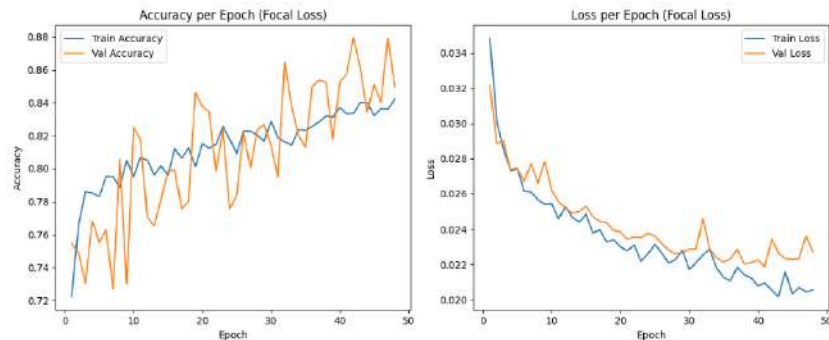


Gambar 4. 9 kurva training validasi skenario 2 kombinasi

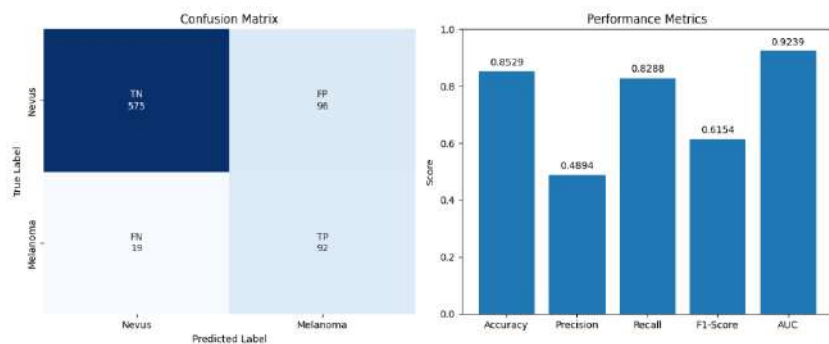


Gambar 4. 10 Confusion matrix skenario 2 kombinasi

Metode Kombinasi (SMOTE + Class Weight). Pendekatan ganda yang menerapkan SMOTE pada level dataset sekaligus Class Weight pada fungsi loss. Tujuannya memaksimalkan perhatian model terhadap *Melanoma* dari dua sisi sekaligus. Hasil: *Recall* = 0,9099, *Precision* = 0,4073, *AUC* = 0,9282, *FN* = 10, *FP* = 147. Meskipun menghasilkan *Recall* tertinggi di antara semua skenario 2, *Precision* mengalami collapse (hanya 0,4073) dengan *FP* mencapai 147 artinya hampir seperempat dari seluruh prediksi positif adalah false alarm. Metode ini tidak dipilih sebagai yang terbaik karena *Precision* yang terlalu rendah dinilai tidak dapat diterima secara klinis.

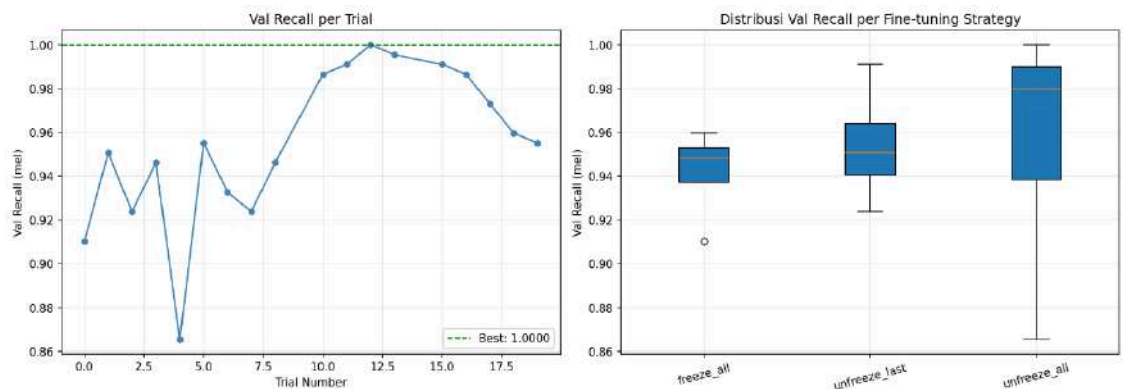


Gambar 4. 11 kurva training validasi skenario 2 focal loss



Gambar 4. 12 Confusion matrix skenario 2 focal loss

Metode Focal Loss menggantikan CrossEntropyLoss standar dengan fungsi loss dinamis $FL(pt) = -\alpha(1 - pt)^{\gamma} \log(pt)$, dengan $\alpha=0,8576$ (proporsional distribusi *Nevus* di data latih) dan $\gamma=2$. Parameter γ menurunkan bobot sampel yang mudah diklasifikasikan sehingga model terfokus pada kasus-kasus sulit, termasuk sampel *Melanoma* yang ambigu. Hasil: *Recall* = 0,8288, *Precision* = 0,4894, *AUC* = 0,9239, *FN* = 19, *FP* = 96. Focal Loss adalah satu-satunya metode berbasis loss dinamis (cost-sensitive dinamis) dibandingkan Class Weight yang bersifat statis, menjadikannya perbandingan utama dalam paper jurnal.



Gambar 4. 13 Visualisasi 20 Trial Hyper Parameter tuning

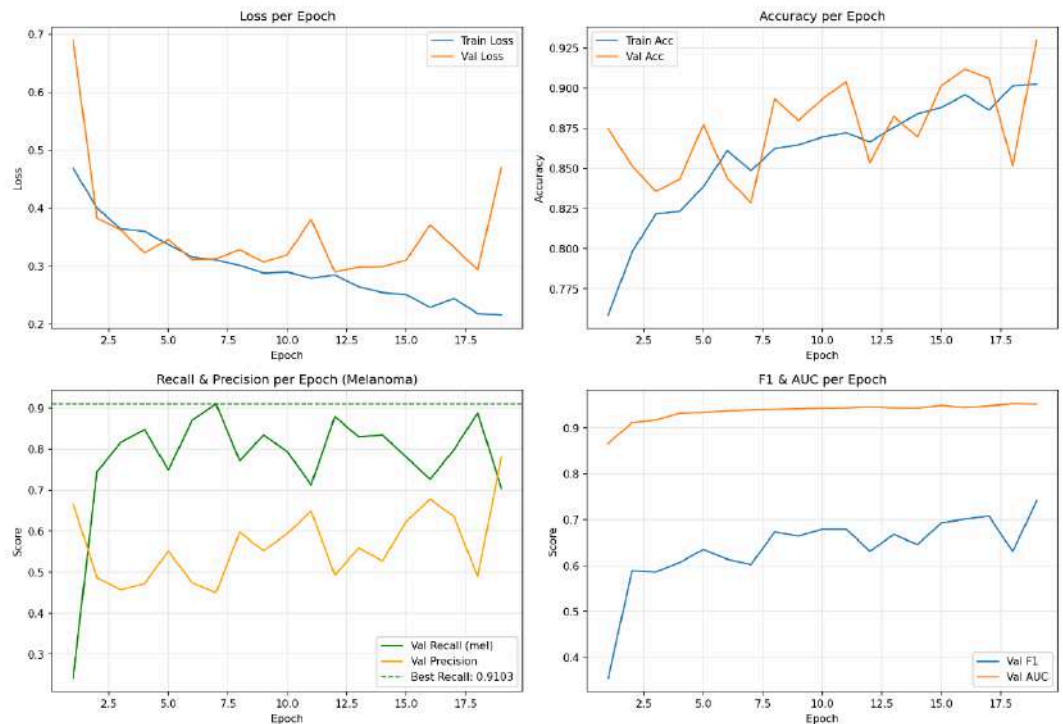
Skenario 3 bertujuan mencari konfigurasi hiperparameter optimal secara otomatis menggunakan Optuna, sebuah framework optimasi hiperparameter berbasis Tree-structured Parzen Estimator (TPE). Metode terbaik dari Skenario 2 (Class Weight) digunakan sebagai basis, dengan ruang pencarian mencakup: *learning rate* (10^{-5} hingga 10^{-2}), *weight decay*, *dropout rate*, strategi fine-tuning backbone (*freeze_all*, *unfreeze_last*, *unfreeze_all*), sudut rotasi augmentasi, dan penggunaan horizontal flip. Fungsi objektif Optuna dirancang dengan kendala *Precision* minimum ($Precision \geq 0,30$ pada validasi) untuk menghindari konfigurasi yang *Recall*-nya tinggi namun mengalami *Precision collapse*.

Dari 20 trial yang dijalankan, Trial 12 menghasilkan skor tertinggi (1,0) namun dikecualikan karena menunjukkan indikasi collapse *Recall* validasi sempurna mengindikasikan model memprediksi semua sampel sebagai *Melanoma*. Trial 9 dipilih sebagai konfigurasi optimal karena menghasilkan keseimbangan terbaik antara *Recall* tinggi dan *Precision* yang dapat diterima secara klinis. Konfigurasi Trial 9:

- $lr = 4,811 \times 10^{-4}$
- $weight_decay = 2,107 \times 10^{-5}$
- $dropout = 0,1254$
- fine-tuning = *unfreeze_all*
- rotasi = 19°
- horizontal flip = aktif

Actual Label	nv (actual)	553 TN	118 FP
	mel (actual)	13 FN	98 TP
		nv (pred)	mel (pred)
		Predicted Label	

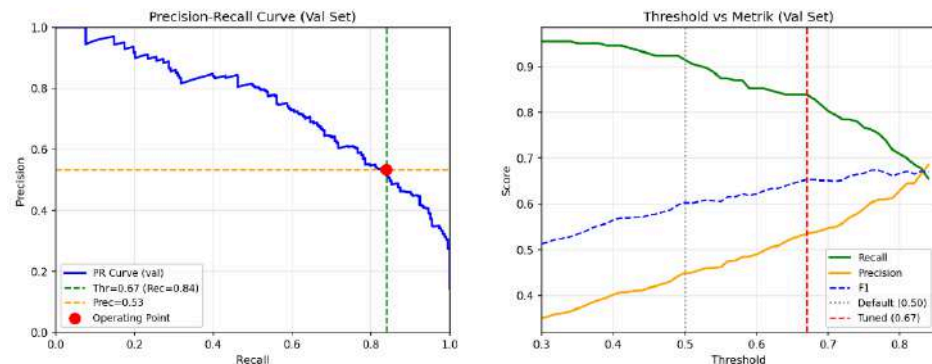
Gambar 4. 14 confusion matrix skenario 4



Gambar 4. 15 kurva training validasi skenario 4

Skenario 4 merupakan tahap pelatihan final menggunakan konfigurasi hiperparameter terbaik hasil Skenario 3 (Trial 9). Perbedaan utama dari skenario sebelumnya adalah strategi fine-tuning unfreeze_all, di mana seluruh lapisan backbone *MobileNetV3-Small* dibuka untuk dilatih bersama lapisan classifier. Untuk mencegah *learning rate* yang terlalu besar merusak fitur yang sudah dipelajari backbone, diterapkan differential *learning rate*: backbone menggunakan $lr=4,811 \times 10^{-5}$ (sepersepuluh dari *lr* classifier) sementara classifier menggunakan $lr=4,811 \times 10^{-4}$. Class Weight tetap diterapkan sebagai strategi penanganan imbalance.

Hasil pengujian pada test set dengan threshold default (0,5): *Recall*=0,8829, *Precision*=0,4537, *AUC*=0,9437, FN=13, FP=118. Model berhasil mendeteksi 98 dari 111 kasus *Melanoma* pada test set meningkat signifikan dari baseline (63 dari 111 kasus). *AUC* sebesar 0,9437 merupakan nilai tertinggi dari seluruh skenario, mencerminkan kemampuan diskriminasi model yang sangat baik secara keseluruhan.



Gambar 4. 16 Analisis Threshold Tuning: Kurva *Precision-Recall* vs Threshold

Pada klasifikasi biner berbasis softmax, model menghasilkan probabilitas $P(\text{Melanoma})$ untuk setiap sampel. Secara default, keputusan diambil berdasarkan kelas dengan probabilitas tertinggi, yang setara dengan threshold 0,5 sampel diklasifikasikan sebagai *Melanoma* jika $P(\text{Melanoma}) > 0,5$. Threshold tuning adalah proses menggeser batas keputusan ini untuk menyeimbangkan trade-off antara *Recall* dan *Precision* sesuai kebutuhan klinis.

Threshold yang lebih rendah dari 0,5 akan meningkatkan *Recall* (lebih banyak *Melanoma* terdeteksi) namun meningkatkan false positive. Sebaliknya, threshold yang lebih tinggi akan meningkatkan *Precision* (lebih sedikit false alarm) namun berisiko melewati kasus *Melanoma* yang sesungguhnya. Pencarian threshold optimal dilakukan pada validation set untuk menghindari data leakage threshold kemudian dievaluasi pada test set yang sepenuhnya terpisah.

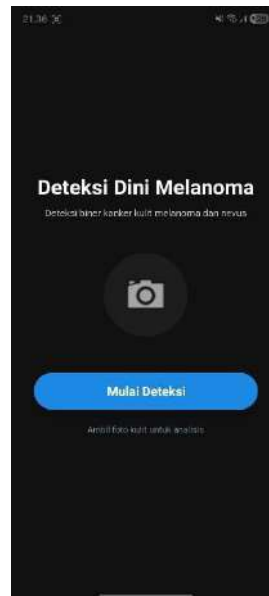
Kriteria pemilihan threshold: $\text{Recall} \geq 0,80$ pada validation set, kemudian memaksimalkan F1-score. Hasilnya, threshold optimal adalah 0,67. Evaluasi pada test set dengan threshold 0,67: $\text{Recall}=0,8288$, $\text{Precision}=0,5610$, $\text{F1}=0,6691$, $\text{FN}=19$, $\text{FP}=72$. Dibandingkan dengan threshold default 0,5 ($\text{FP}=118$), threshold 0,67 berhasil mengurangi false positive sebesar 39% (dari 118 menjadi 72) dengan konsekuensi *Recall* turun dari 0,8829 menjadi 0,8288. Untuk keperluan aplikasi *Android* yang ditujukan sebagai alat skrining awal, threshold 0,5 tetap digunakan karena *Recall* yang lebih tinggi lebih diprioritaskan dalam konteks skrining klinis.

Tabel 4. 3 Hasil lengkap seluruh skenario eksperimen (Test set)

Skenario	Metode	<i>Recall</i>	<i>Precision</i>	F1	AUC	FN	FP
S1 Baseline	Tanpa imbalance	0,5676	0,7079	0,630 0	0,9253	48	26
S2a Class Weight	Class Weight	0,8649	0,4848	0,621 4	0,9240	15	102
S2b SMOTE	SMOTE	0,7387	0,5816	0,650 8	0,9318	29	59
S2c WRS	Weighted Random Sampler	0,7477	0,5764	0,651 0	0,9275	28	61
S2d Kombinasi	SMOTE + Class Weight	0,9099	0,4073	0,562 7	0,9282	10	147
S2e Focal Loss	Focal Loss ($\alpha=0,858$, $\gamma=2$)	0,8288	0,4894	0,615 4	0,9239	19	96
S4 Final (thr=0,5)	CW + Optuna + Unfreeze	0,8829	0,4537	0,599 4	0,9437	13	118
S4 Final (thr=0,67)	CW + Optuna + Unfreeze	0,8288	0,5610	0,669 1	0,9437	19	72

2) Aplikasi *Android*

Aplikasi *Android* dikembangkan menggunakan bahasa Kotlin dengan library PyTorch Mobile sebagai engine inferensi pada perangkat. Aplikasi ini terdiri dari beberapa layar utama yang membentuk alur penggunaan yang sederhana dan intuitif, mulai dari layar pembuka hingga layar hasil analisis. layar utama (*MainActivity*) merupakan halaman awal yang ditampilkan saat aplikasi pertama kali dibuka. Layar ini menampilkan judul "Deteksi Dini *Melanoma*" beserta subjudul "Deteksi biner kanker kulit *Melanoma* dan *Nevus*", ikon kamera sebagai penanda fungsi utama, serta tombol "Mulai Deteksi" berwarna biru yang menjadi titik masuk proses deteksi. Di bawah tombol tersebut terdapat teks petunjuk "Ambil foto kulit untuk analisis". Tampilan layar utama ditunjukkan pada Gambar 4.17.



Gambar 4. 17 Layar Utama Aplikasi Deteksi Dini *Melanoma*

Layar Input Kamera ditampilkan setelah pengguna menekan tombol "Mulai Deteksi". Layar ini menggunakan antarmuka kamera sistem *Android* yang menampilkan preview secara real-time. Pengguna mengarahkan kamera ke area lesi kulit yang ingin diperiksa, kemudian menekan tombol rana untuk mengambil foto. Tampilan layar input kamera ditunjukkan pada Gambar 4.18.



Gambar 4. 18 Layar Input Gambar via Kamera

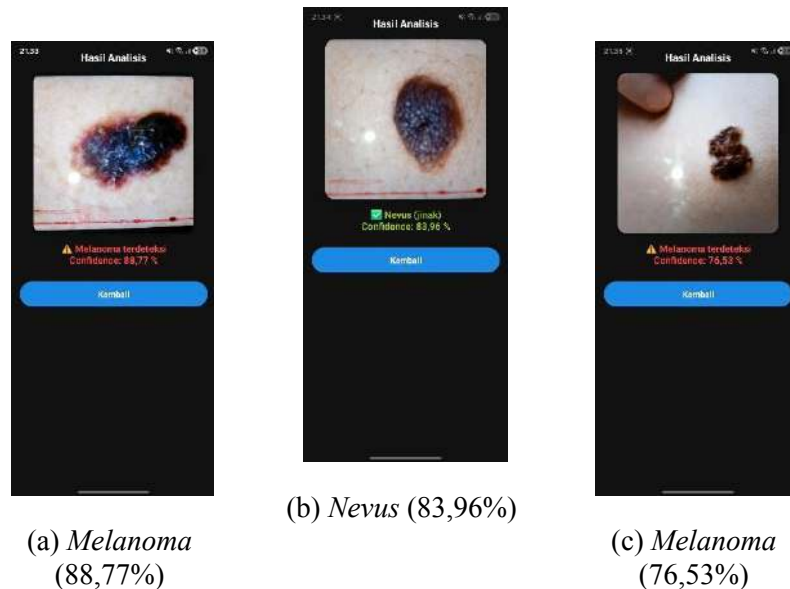
Layar Konfirmasi ditampilkan setelah foto berhasil diambil. Pada layar ini, pengguna dapat melihat pratinjau gambar lesi yang telah diambil secara penuh layar, disertai dua pilihan di bagian bawah: tombol "Coba lagi" untuk mengambil ulang foto apabila hasil kurang memuaskan, dan tombol "Ok" untuk melanjutkan proses klasifikasi. Layar konfirmasi ditunjukkan pada Gambar 4.19.



22

Gambar 4. 19 Layar Konfirmasi Gambar Sebelum Diklasifikasi

Layar Hasil Analisis (*ResultActivity*) menampilkan keluaran akhir dari proses inferensi model. Layar ini memuat gambar lesi yang dianalisis, label prediksi (*Melanoma* atau *Nevus*), dan nilai *confidence* dalam persen. Teks hasil ditampilkan dengan warna merah disertai ikon peringatan ($\triangle$) untuk prediksi *Melanoma*, dan warna hijau disertai ikon centang ($\checkmark$) untuk prediksi *Nevus*, sehingga pengguna dapat membaca hasil dengan cepat secara visual. Tombol "Kembali" tersedia di bagian bawah layar untuk kembali ke Layar Utama. Gambar 4.20 menyajikan tiga contoh hasil analisis yang menampilkan variasi prediksi model.



Gambar 4. 20 Layar Hasil Analisis: (a) Prediksi *Melanoma* confidence tinggi, (b) Prediksi *Nevus*, (c) Prediksi *Melanoma* confidence sedang

3) Snippet Kode Utama Aplikasi *Android*

Berikut adalah tiga snippet kode yang merepresentasikan fungsi inti aplikasi *Android*. Ketiga fungsi ini membentuk alur utama sistem, mulai dari memuat model hingga menampilkan hasil kepada pengguna.

Segmen program 4. 1 : Memuat model

```
fun loadModel(context: Context) {
    if (module != null) return
    val file = File(context.filesDir, "mobilenet_classweight.ptl")
    if (!file.exists()) {
        context.assets.open("mobilenet_classweight.ptl").use { input ->
            FileOutputStream(file).use { output ->
                input.copyTo(output)
            }
        }
    }
    module = Module.load(file.absolutePath)
}
```

Fungsi loadModel() memuat model PyTorch Mobile berformat .ptl dari direktori assets aplikasi ke penyimpanan internal perangkat. Pengecekan if (!file.exists()) dilakukan untuk menghindari penyalinan ulang model setiap kali aplikasi dibuka, sehingga proses muat model hanya berlangsung sekali pada penggunaan pertama *ImageNet*.

Segmen program 4. 2 : Preprocessing Gambar Sebelum Inferensi

```
private fun preprocess(bitmap: Bitmap): Tensor {
    val targetResize = 256
    val cropSize = 224
    val w = bitmap.width
    val h = bitmap.height
    val scale = targetResize.toFloat() / minOf(w, h)
    val newW = (w * scale).toInt()
    val newH = (h * scale).toInt()
    val resized = Bitmap.createScaledBitmap(bitmap, newW, newH, true)

    val cropX = (resized.width - cropSize) / 2
    val cropY = (resized.height - cropSize) / 2
    val cropped = Bitmap.createBitmap(resized, cropX, cropY, cropSize, cropSize)

    transforms.Normalize([0.485f, 0.456f, 0.406f], [0.229f, 0.224f, 0.225f])
    return TensorImageUtils.bitmapToFloat32Tensor(
        cropped,
        floatArrayOf(0.485f, 0.456f, 0.406f),
        floatArrayOf(0.229f, 0.224f, 0.225f)
    )
}
```

Fungsi preprocess() mengubah Bitmap masukan menjadi tensor yang siap diproses model. Prosesnya berjalan dalam tiga tahap berurutan: pertama, mengubah ukuran gambar dengan mempertahankan rasio aspek hingga sisi terpendeknya mencapai 256 piksel; kedua, memotong tepat di bagian tengah

gambar (*center crop*) menjadi ukuran 224x224 piksel; dan terakhir, mengonversi gambar tersebut menjadi Tensor 32-bit sekaligus menormalisasi warna pikselnya menggunakan standar rata-rata (*mean*) dan simpangan baku (*std*) bawaan *ImageNet*.

Segmen program 4.3 : Mengambil dan Menampilkan Output Prediksi

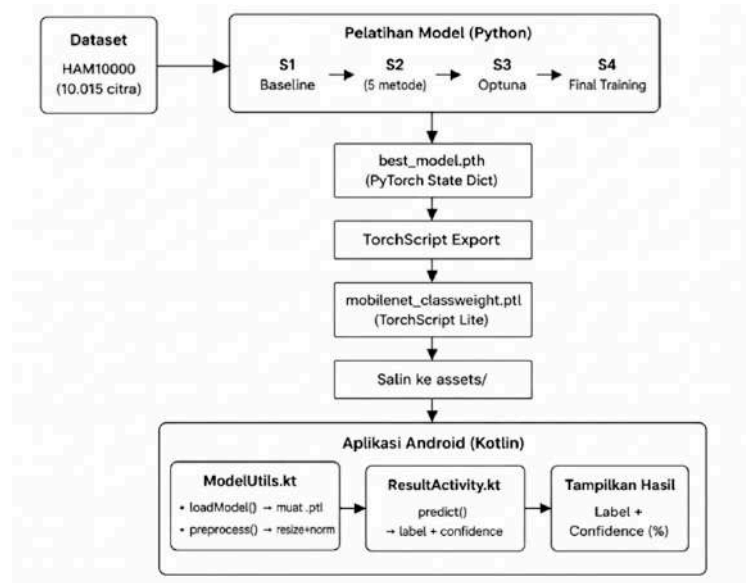
```
try {
    val (label, score) = ModelUtils.predict(this@ResultActivity,
finalBitmap)
    val percent = String.format("%.2f", score * 100)

    if (label == "mel") {
        resultText.text = "Melanoma terdeteksi\nConfidence: $percent %"
        resultText.setTextColor(getColor(Android.R.color.holo_red_light))
    } else {
        resultText.text = "Nevus (jinak)\nConfidence: $percent %"
        resultText.setTextColor(getColor(Android.R.color.holo_green_light))
    }
} catch (e: Exception) {
    resultText.text = "Error: ${e.message}"
}
```

Blok kode pada *ResultActivity* memanggil *ModelUtils.predict()* yang mengembalikan pasangan label dan skor kepercayaan. Label "mel" (*Melanoma*) ditampilkan dalam warna merah sebagai peringatan, sementara label "nv" (*Nevus*) ditampilkan dalam warna hijau. Blok try/catch memastikan aplikasi tidak berhenti mendadak apabila terjadi kesalahan pada proses inferensi.

b. Keterkaitan lembar kerja

Seluruh lembar kerja yang dikembangkan pada penelitian ini saling berkaitan membentuk satu proses end-to-end yang utuh, mulai dari data mentah hingga aplikasi yang dapat dioperasikan di perangkat mobile. Alur integrasi antar komponen ditunjukkan pada Gambar 4.5.



Gambar 4. 21 Diagram Alur Integrasi Sistem dari Proses Pelatihan hingga Aplikasi *Android*

Alur integrasi dimulai dari dataset *HAM10000* yang diproses oleh sebelas notebook pelatihan secara berurutan. Notebook S1 hingga S4 menghasilkan model terbaik yang disimpan dalam format .pth (PyTorch state dict). Model dari Skenario 4 yang menggunakan parameter Trial 9 Optuna kemudian diekspor ke format TorchScript Lite (.ptl) agar kompatibel dengan library PyTorch Mobile pada platform *Android*.

File .ptl ditempatkan di direktori assets/ proyek *Android Studio*. Saat aplikasi pertama kali dijalankan, modul *ModelUtils.kt* menyalin file tersebut ke penyimpanan internal perangkat, kemudian memuat model menggunakan *Module.load()*. Ketika pengguna mengambil foto, *ModelUtils.kt* melakukan preprocessing gambar (resize ke 224×224 dan normalisasi *ImageNet*) sebelum mengirimkannya ke model. Hasil prediksi diterima oleh *ResultActivity.kt* yang menampilkan label dan confidence kepada pengguna.

2. Pengujian Fungsional

Pengujian fungsional dilakukan menggunakan metode black-box testing untuk memverifikasi bahwa seluruh fungsi aplikasi berjalan sesuai dengan rancangan yang ditetapkan pada Bab III. Pengujian difokuskan pada alur utama penggunaan aplikasi, mulai dari membuka aplikasi hingga menampilkan hasil klasifikasi. Tabel 4.2 menyajikan hasil pengujian fungsional secara lengkap.

Tabel 4. 4 Hasil Pengujian Fungsional Aplikasi *Android*

No	Fungsi	Input / Aksi	Output yang Diharapkan	Output Aktual	Status
1	Membuka aplikasi	Pengguna membuka aplikasi	Layar Utama dengan judul, ikon kamera, dan tombol Mulai Deteksi tampil	Layar Utama berhasil ditampilkan dengan judul "Deteksi Dini <i>Melanoma</i> " dan	Berhasil

				tombol "Mulai Deteksi"	
2	Mengambil foto via kamera	Pengguna menekan tombol "Mulai Deteksi"	Kamera sistem <i>Android</i> terbuka dan menampilkan preview	Kamera terbuka menampilkan preview lesi kulit secara real-time	Berhasil
3	Konfirmasi gambar sebelum klasifikasi	Pengguna menekan tombol potret kamera	Layar konfirmasi tampil dengan preview gambar, tombol "Coba lagi" dan "Ok"	Layar konfirmasi berhasil tampil dengan gambar lesi dan dua opsi pilihan	Berhasil
4	Mengulang pengambilan gambar	Pengguna menekan "Coba lagi" pada layar konfirmasi	Kembali ke tampilan kamera untuk mengambil ulang foto	Kamera kembali terbuka dan pengguna dapat mengambil foto baru	Berhasil
5	Melakukan klasifikasi	Pengguna menekan "Ok" pada layar konfirmasi	Proses inferensi berjalan dan layar Hasil Analisis tampil	Layar Hasil Analisis berhasil tampil beserta gambar dan prediksi model	Berhasil
6	Menampilkan label prediksi dan confidence	Model selesai melakukan inferensi	Label (<i>Melanoma/Nevus</i>) dan nilai confidence dalam persen ditampilkan	Label dan confidence tampil sesuai output model (contoh: " <i>Melanoma</i> terdeteksi, Confidence: 88,77%")	Berhasil
7	Kembali ke layar utama	Pengguna menekan tombol "Kembali"	Aplikasi kembali ke Layar Utama	Layar Utama berhasil ditampilkan kembali	Berhasil

3. Pengujian Non Fungsional

Pengujian non-fungsional mencakup dua aspek: (1) pengukuran ukuran model dan waktu inferensi pada lingkungan pelatihan menggunakan benchmark notebook, dan (2) pengujian performa aplikasi pada perangkat *Android* nyata.

a. Ukuran dan model Inferensi

Pengukuran dilakukan menggunakan notebook benchmark_nonfungsional.ipynb pada model Skenario 4 (*MobileNetV3-Small*). Tabel 4.3 merangkum hasil pengukuran.

Tabel 4. 5 Hasil Benchmark Model *MobileNetV3-Small*

Parameter	Nilai
Arsitektur	<i>MobileNetV3-Small</i>
Format model (deployment)	.ptl (PyTorch Mobile / TorchScript Lite)
Ukuran file .pth	5,93 MB (6.214.213 byte)
Total parameter	1.519.906 (~1,5 juta)
Estimasi kebutuhan RAM (fp32)	5,8 MB
Waktu inferensi CPU (mean $\pm$ std)	131,26 ms $\pm$ 8,94 ms
Waktu inferensi CPU (P95)	144,35 ms
Waktu inferensi GPU (mean $\pm$ std)	40,04 ms $\pm$ 2,47 ms
Waktu inferensi GPU (P95)	45,78 ms
Perangkat benchmark CPU	Intel Core i5-10500H (Intel64, GenuineIntel)
Perangkat benchmark GPU	NVIDIA GeForce GTX 1650 Max-Q Design
Jumlah run (setelah warmup)	100 run (20 warmup)

Ukuran model sebesar 5,93 MB dan total parameter ~1,5 juta menunjukkan bahwa *MobileNetV3-Small* merupakan arsitektur yang sangat ringan dan sesuai untuk deployment mobile. Batasan ukuran APK *Android* umumnya berada di kisaran 50–100 MB, sehingga model ini memiliki ruang yang sangat luas untuk komponen aplikasi lainnya. Waktu inferensi CPU rata-rata 131,26 ms menandakan bahwa model mampu memproses satu gambar dalam waktu kurang dari seperlima detik pada lingkungan notebook PC, sementara akselerasi GPU menghasilkan inferensi yang jauh lebih cepat di angka 40,04 ms.

b. Konfigurasi Hiperparameter Optimal (Trial 9 Optuna)

Tabel 4. 6 Konfigurasi Hiperparameter Terpilih (Trial 9 Optuna)

Hiperparameter	Nilai	Keterangan
----------------	-------	------------

<i>Learning rate</i> (lr)	$4,811 \times 10^{-4}$	Ditemukan oleh Optuna (range: $10^{-5} - 10^{-2}$)
Weight Decay	$2,107 \times 10^{-5}$	Regularisasi L2 untuk mencegah overfitting
Dropout Rate	0,1254	Dropout pada lapisan classifier
Fine-tune Strategy	unfreeze_all	Seluruh backbone dan classifier dilatih
Rotasi Augmentasi	19°	Sudut rotasi acak maksimum pada data latih
Horizontal Flip	Ya (True)	Augmentasi flip horizontal diaktifkan
Batch Size	32	Tetap (tidak dioptimasi, sesuai Standing Instructions)
Optimizer	Adam	Digunakan di semua skenario (lr backbone 1/10 lr classifier pada S4)
Epoch	100	jumlah total keseluruhan training pada model
Best Epoch	41	Nilai epoch dengan metrik evaluasi terbaik
Early stop (patience)	7	Di trial ke berapa training akan dihentikan ketika performa model mulai memburuk dari trail sebelumnya

Tabel di atas menampilkan konfigurasi hiperparameter Trial 9 yang dipilih dari 20 trial Optuna. Meskipun Trial 12 menghasilkan skor tertinggi (val *Recall* 1,0), trial tersebut dikecualikan karena menunjukkan indikasi collapse (*Recall* validasi sempurna namun *Precision* sangat rendah). Trial 9 dipilih karena menghasilkan keseimbangan terbaik antara *Recall* yang tinggi dan *Precision* yang dapat diterima secara klinis (*Precision* test > 0,45 pada threshold default). Nilai *learning rate* $4,811 \times 10^{-4}$ merupakan hasil pencarian adaptif Optuna yang lebih besar dari lr default S1–S2 (1×10^{-4}), konsisten dengan kebutuhan fine-tuning seluruh backbone.

c. Pengujian di Perangkat *Android*

Pengujian dilakukan pada perangkat Samsung Galaxy A07 (RAM 4 GB, *Android* 16, minSdk 27) menggunakan sepuluh gambar lesi kulit (lima *Melanoma* dan lima *Nevus*). Waktu yang diukur mencakup seluruh Proses end-to-end: memuat gambar, preprocessing, inferensi model, hingga tampilan hasil. Tabel 4.4 menyajikan hasil pengujian secara rinci. (*) Tanda bintang pada Test 10 menandakan prediksi yang salah (false positive).

Tabel 4. 7 Hasil Pengujian Inferensi di Perangkat *Android*

Test	Gambar Asli	Prediksi	Confidence	Waktu (detik)	Status	Gambar
1	<i>Melanoma</i>	<i>Melanoma</i>	54,83%	2,47	Benar	
2	<i>Melanoma</i>	<i>Melanoma</i>	97,11%	1,46	Benar	
3	<i>Melanoma</i>	<i>Melanoma</i>	89,82%	0,90	Benar	
4	<i>Melanoma</i>	<i>Melanoma</i>	94,41%	1,81	Benar	
5	<i>Melanoma</i>	<i>Melanoma</i>	91,93%	0,85	Benar	
6	<i>Nevus</i>	<i>Nevus</i>	57,20%	1,27	Benar	
7	<i>Nevus</i>	<i>Nevus</i>	59,21%	0,93	Benar	
8	<i>Nevus</i>	<i>Nevus</i>	54,66%	0,92	Benar	
9	<i>Nevus</i>	<i>Nevus</i>	85,18%	0,83	Benar	

10	<i>Nevus</i>	<i>Melanoma*</i>	51,57%	0,91	Salah (FP)	
Rata-rata	1,235 detik	9/10 Benar (90%)				

Dari sepuluh pengujian, sembilan diantaranya menghasilkan prediksi yang benar (akurasi 90%) dengan waktu inferensi rata-rata 1,235 detik. Perbedaan waktu antara pengujian *Android* (rata-rata 1,235 detik) dan *benchmark* CPU notebook (131 ms) disebabkan oleh perbedaan cakupan pengukuran: benchmark notebook hanya mengukur waktu *forward pass* model, sementara pengukuran *Android* mencakup seluruh Proses mulai dari memuat gambar, *preprocessing*, eksekusi model, hingga rendering antarmuka pengguna. Selain itu, keterbatasan komputasi perangkat mobile (CPU Octa-core tanpa akselerasi GPU neural) turut berkontribusi pada perbedaan waktu tersebut.

Test ke-10 menghasilkan *false positive* (*Nevus* diprediksi sebagai *Melanoma*) dengan nilai *confidence* 51,57%. Nilai kepercayaan yang rendah ini konsisten dengan karakteristik model yang bersifat *Recall-oriented*: model lebih memilih memprediksi *Melanoma* pada kasus ambigu daripada melewatkan kasus *Melanoma* sejati. Dalam konteks klinis, *false positive* dengan kepercayaan rendah justru memberikan sinyal bahwa kasus tersebut perlu pemeriksaan lebih lanjut, yang merupakan perilaku yang diharapkan dari sistem skrining dini.

B. Pembahasan

1. Perbandingan dengan Penelitian Terdahulu

Tabel 4.8 membandingkan hasil penelitian ini dengan penelitian terdahulu yang menggunakan dataset *HAM10000* untuk klasifikasi *Melanoma*. Perbandingan difokuskan pada metrik *Recall Melanoma* sebagai prioritas klinis dan AUC-ROC sebagai ukuran kemampuan diskriminasi model.

Tabel 4. 8 Perbandingan Hasil Penelitian dengan Penelitian Terdahulu

No	Peneliti	Dataset	Arsitektur	Penanganan imbalance data	Akurasi	<i>Recall</i> (mel)	AUC	Input	Mobile Ready
1.	Al-Asadi & Altun (2022)	HAM10000 (7 kelas)	Custom DLCNN	SMOTE	85,99 %	88%	-	28x28	Tidak
2.	Javaid et al (2025)	HAM10000 ISIC 2018, PH2	DSSC-NET (Custom CNN)	SMOTE-Tomek	97,82 %	97%	0,9943	28x28	Tidak
3.	Kumar Lihore et al (2024)	HAM10000 (7 kelas)	U-Net + MobileNetV3 + Bayesian Opt	Augmentasi	98,86 %	96,45 %	0,9845	192x256	Tidak

4.	Mateen et al (2024)	ISIC 2020, HAM10000	Inception ResNet-v2 + ViT + U-Net	Augmentasi	98,65 %	99,20 %	-	224x224	Tidak
5.	Nguyen et al (2022)	HAM10000 (7 kelas)	Inception ResNetv2 + Self attention	Custom weight loss	90%	0,58	0,99	224x224	Tidak
6.	Areej Aloitabi & Al Saeed (2025)	HAM1000 (biner: kanker vs normal)	Xception + Self attention	Augmentasi	94,11 %	95,47 %	-	224x224	Tidak
7.	Widhia KZ et al (2025)	HAM10000 (7 kelas)	DenseNet201 + Random oversampling	Random Oversampling	96,52 %	98%	-	32x32	Tidak
8.	Juni ismail et al (2025)	HAM10000 (3 kelas)	MobileNetv2 + CBAM	Augmentasi	86%	86,46 %	-	224x224	Tidak
9.	Mamun et al (2025)	HAM10000 (7 kelas)	ResNet50 + Transfer Learning	tidak ada	89,08 %	-	-	224x224	Tidak
10.	Penelitian ini (S4, threshold=0,5)	HAM10000 (biner: mel vs nv)	MobileNetV3-Small	Class Weight	88,56 %	88,29 %	0,9437	224x224	Ya
11.	Penelitian ini (S4, threshold=0,67)	HAM10000 (biner: mel vs nv)	MobileNetV3-Small	Class Weight	87,44 %	82,88 %	0,9437	224x224	Ya

Tabel 4.5 menyajikan perbandingan hasil penelitian ini dengan sembilan penelitian terdahulu yang menggunakan dataset *HAM10000* atau dataset dermoskopi serupa. Terdapat beberapa hal penting yang perlu dicermati dalam membaca tabel ini. Pertama, sebagian besar penelitian terdahulu mengevaluasi model pada tujuh kelas sekaligus (multi-class), sehingga nilai *Recall* yang dilaporkan merupakan rata-rata dari semua kelas, bukan spesifik untuk kelas *Melanoma*. Penelitian ini secara khusus memfokuskan klasifikasi biner *Melanoma* versus melanocytic nevi dan melaporkan *Recall* kelas *Melanoma* secara eksplisit, sehingga perbandingan langsung dengan sebagian besar penelitian terdahulu terbatas pada metrik akurasi dan AUC saja.

Kedua, satu-satunya penelitian yang juga menggunakan *MobileNetV3* pada *HAM10000* adalah Nguyen et al. (2022) dan Kumar Lilhore et al. (2024). Nguyen et al. melaporkan bahwa *MobileNetV3Large* hanya menghasilkan *Recall Melanoma* sebesar 0,58 nilai yang jauh di bawah hasil penelitian ini sebesar 0,8829. Hal ini menunjukkan bahwa penggunaan arsitektur ringan saja tidak

cukup tanpa strategi penanganan ketidakseimbangan data yang tepat. Kumar Lilhore et al. mencapai sensitivity 96,35% namun dengan arsitektur hybrid U-Net + *MobileNetV3* yang jauh lebih berat secara komputasi dan tidak dirancang untuk deployment pada perangkat mobile.

Ketiga, penelitian dengan akurasi tertinggi seperti Javaid et al. (2025) dengan 97,82% dan Kumar Lilhore et al. dengan 98,86% menggunakan resolusi input gambar yang sangat rendah (28×28 piksel) atau arsitektur yang sangat kompleks. Resolusi 28×28 piksel berpotensi kehilangan fitur tekstur dan warna yang kritis pada citra dermoskopi fitur-fitur yang justru menjadi dasar diagnosis klinis *Melanoma*. Penelitian ini menggunakan resolusi standar 224×224 piksel, konsisten dengan rekomendasi untuk citra dermoskopi.

Keempat, tidak ada satu pun dari sembilan penelitian terdahulu yang mengimplementasikan model ke dalam aplikasi mobile dan mengevaluasinya pada perangkat nyata. Penelitian ini membuktikan bahwa *MobileNetV3-Small* dengan ukuran model 5,93 MB mampu melakukan inferensi rata-rata 1,235 detik pada Samsung Galaxy A07 memenuhi kebutuhan praktis skrining berbasis smartphone di lapangan.

Dari keseluruhan perbandingan, penelitian ini memiliki posisi yang unik sebagai satu-satunya penelitian yang secara eksplisit melaporkan *Recall Melanoma* pada setting biner *Melanoma* versus melanocytic nevi dengan lima metode imbalance handling yang dibandingkan secara sistematis, sekaligus membuktikan kelayakan deployment model pada perangkat *Android* nyata.

2. Analisis Kelebihan dan Kekurangan

Penelitian ini memiliki beberapa keunggulan yang relevan secara klinis dan teknis. Pertama, model mencapai *Recall Melanoma* sebesar 0,8829 pada test set, yang berarti 88,29% kasus *Melanoma* berhasil terdeteksi. Dalam konteks skrining klinis, nilai ini bermakna bahwa hanya 13 dari 111 kasus *Melanoma* yang tidak terdeteksi (false negative), mengurangi risiko kondisi yang paling berbahaya dari perspektif diagnostik.

Kedua, model mencapai AUC-ROC sebesar 0,9437 yang merupakan nilai tertinggi dari seluruh skenario dan kompetitif dibandingkan penelitian terdahulu dengan arsitektur yang lebih kompleks. AUC yang tinggi menunjukkan bahwa kemampuan diskriminasi model sangat baik, terlepas dari pilihan threshold keputusan.

Ketiga, ukuran model yang hanya 5,93 MB dengan ~1,5 juta parameter menjadikan *MobileNetV3-Small* pilihan yang tepat untuk deployment mobile. Model ini dapat diintegrasikan ke dalam aplikasi *Android* tanpa membebani kapasitas penyimpanan perangkat, dan dengan waktu inferensi rata-rata 1,235 detik pada Samsung Galaxy A07 (RAM 4 GB), pengalaman pengguna tetap dapat diterima untuk aplikasi skrining nonrealtime.

Di sisi lain, penelitian ini juga memiliki beberapa keterbatasan yang perlu diakui. Pertama, nilai *Precision* pada threshold default (0,4537) dinilai rendah, yang berarti 54,63% dari seluruh prediksi *Melanoma* adalah false positive. Melalui threshold tuning pada nilai 0,67, *Precision* dapat ditingkatkan menjadi 0,5610 dengan *Recall* tetap berada di atas 0,82. Namun demikian, validasi lebih lanjut oleh tenaga medis profesional tetap diperlukan sebelum sistem ini dapat

digunakan dalam praktik klinis sesungguhnya. Kedua, seluruh eksperimen dijalankan dengan satu random seed (seed=42) tanpa validasi silang, sehingga estimasi variabilitas performa model tidak tersedia.

Pengujian langsung pada perangkat *Android* Samsung Galaxy A07 menunjukkan bahwa aplikasi dapat beroperasi dengan baik pada perangkat kelas menengah bawah (RAM 4 GB, tanpa akselerasi NPU/GPU khusus). Antarmuka yang disederhanakan menjadi satu alur linear (buka aplikasi → foto → konfirmasi → hasil) memudahkan pengguna awam untuk mengoperasikan aplikasi tanpa membutuhkan pengetahuan teknis. Pewarnaan hasil prediksi (merah untuk *Melanoma*, hijau untuk *Nevus*) memberikan umpan balik visual yang intuitif.

Keunggulan utama sistem dibandingkan penelitian terdahulu adalah efisiensi komputasi: *MobileNetV3-Small* menghasilkan AUC yang kompetitif (0,9437) dengan ukuran model yang sangat kecil (5,93 MB), menjadikannya kandidat yang layak untuk diintegrasikan ke dalam aplikasi skrining mobile tanpa infrastruktur server eksternal.

Berdasarkan seluruh hasil penelitian dan analisis yang telah dipaparkan, tujuan penelitian sebagaimana dirumuskan pada Bab I dapat dinyatakan tercapai. Pertama, arsitektur *MobileNetV3-Small* berhasil diimplementasikan untuk klasifikasi biner *Melanoma* dan *Nevus* pada dataset *HAM10000* dengan nilai AUC tertinggi 0,9437 dan *Recall Melanoma* 0,8829. Kedua, strategi penanganan ketidakseimbangan data khususnya Class Weight yang menghasilkan *Recall* 0,8649, meningkat menjadi 0,8829 setelah optimasi hiperparameter dan fine-tuning seluruh backbone terbukti secara signifikan meningkatkan sensitivitas model terhadap kelas *Melanoma* dibandingkan baseline (*Recall*=0,5676). Ketiga, optimasi hiperparameter menggunakan Optuna menghasilkan konfigurasi yang meningkatkan AUC dari 0,9240 (S2a) menjadi 0,9437 (S4), mengonfirmasi manfaat tuning sistematis. Keempat, model berhasil dideploy ke dalam aplikasi *Android* yang dapat beroperasi pada perangkat nyata dengan waktu inferensi rata-rata 1,235 detik, membuktikan kelayakan teknis arsitektur lightweight untuk implementasi mobile. Meskipun terdapat keterbatasan pada aspek *Precision* dan fitur galeri yang belum diimplementasi, sistem yang dikembangkan telah memenuhi target utama penelitian sebagai prototipe sistem deteksi dini *Melanoma* berbasis mobile yang efisien secara komputasi. Penelitian ini memberikan kontribusi pada literatur ilmiah berupa perbandingan sistematis lima metode penanganan ketidakseimbangan data pada arsitektur yang identik (*MobileNetV3-Small*, dataset biner *Melanoma* vs. *nevi HAM10000*), yang sebelumnya belum pernah dilakukan dalam satu kerangka eksperimen terkontrol. Selain itu, implementasi end-to-end dari model PyTorch ke aplikasi *Android* yang diuji pada perangkat nyata membuktikan kelayakan arsitektur lightweight untuk skrining *Melanoma* di lingkungan dengan sumber daya komputasi terbatas.

BAB 5

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian yang telah dipaparkan pada Bab IV, dapat ditarik kesimpulan sebagai berikut.

1. Arsitektur *MobileNetV3-Small* berhasil diimplementasikan untuk klasifikasi biner citra dermoskopi *Melanoma* dan *Nevus* pada dataset *HAM10000* yang memiliki distribusi kelas tidak seimbang. Model menghasilkan nilai AUC tertinggi sebesar 0,9437 dan *Recall Melanoma* sebesar 0,8829 pada test set, membuktikan bahwa arsitektur ringan (lightweight) ini mampu mempelajari pola visual yang membedakan kedua kelas tersebut secara efektif meskipun jumlah parameter yang dimiliki jauh lebih kecil dibandingkan arsitektur konvensional.
2. Penerapan strategi penanganan ketidakseimbangan data terbukti berpengaruh signifikan terhadap peningkatan nilai *Recall Melanoma*. Dari kelima strategi yang dibandingkan secara sistematis pada kerangka eksperimen yang identik, yaitu class weighting, SMOTE, weighted random sampler, kombinasi class weighting dan SMOTE, serta Focal Loss, metode class weighting terpilih sebagai strategi terbaik dengan *Recall* sebesar 0,8649, meningkat signifikan dari baseline tanpa penanganan imbalance yang hanya mencapai *Recall* 0,5676. Setelah dikombinasikan dengan optimasi hiperparameter dan fine-tuning seluruh backbone pada tahap akhir, *Recall* meningkat lebih lanjut menjadi 0,8829.
3. Optimasi hiperparameter menggunakan metode pencarian otomatis Optuna berhasil menghasilkan konfigurasi optimal yang meningkatkan kemampuan diskriminasi model, ditunjukkan oleh peningkatan AUC dari 0,9240 pada skenario class weighting menjadi 0,9437 setelah optimasi. Konfigurasi terpilih mencakup *learning rate* $4,811 \times 10^{-4}$, dropout rate 0,1254, dan strategi fine-tuning unfreeze_all dengan differential *learning rate* antara backbone dan classifier, mengonfirmasi bahwa tuning sistematis memberikan kontribusi nyata terhadap performa model dibandingkan penggunaan hiperparameter default.
4. Model yang dihasilkan berhasil diimplementasikan ke dalam aplikasi *Android* dan diuji pada perangkat nyata (Samsung Galaxy A07, RAM 4 GB), dengan ukuran model hanya 5,93 MB dan waktu inferensi rata-rata 1,235 detik per citra. Hasil ini membuktikan kelayakan teknis arsitektur *MobileNetV3-Small* untuk implementasi pada perangkat mobile dengan keterbatasan sumber daya komputasi, sekaligus menjadi pembeda utama penelitian ini dibandingkan sembilan penelitian terdahulu yang dikaji, karena tidak satu pun di antaranya menguji implementasi model pada perangkat nyata.

Secara keseluruhan, ketiga rumusan masalah penelitian telah terjawab dan seluruh tujuan penelitian telah tercapai. Penelitian ini berkontribusi pada literatur ilmiah melalui perbandingan sistematis lima metode penanganan ketidakseimbangan data pada arsitektur yang identik, yang sebelumnya belum pernah dilakukan dalam satu kerangka eksperimen terkontrol, serta melalui pembuktian empiris bahwa arsitektur lightweight dapat diandalkan untuk skrining dini *Melanoma* berbasis mobile tanpa mengorbankan sensitivitas deteksi secara signifikan.

B. Saran

Berdasarkan keterbatasan yang telah diidentifikasi pada pembahasan Bab IV, beberapa saran diajukan untuk pengembangan sistem dan penelitian selanjutnya dengan tema serupa.

1. Penambahan fitur unggah citra dari galeri perangkat perlu dipertimbangkan untuk melengkapi fitur pengambilan foto langsung melalui kamera yang saat ini menjadi satu-satunya metode input, sehingga pengguna memiliki fleksibilitas lebih dalam menggunakan citra yang telah tersimpan sebelumnya.
2. Peningkatan nilai *Precision* model perlu menjadi perhatian pada pengembangan selanjutnya, mengingat nilai *Precision* pada threshold default (0,4537) masih tergolong rendah. Eksplorasi teknik kalibrasi probabilitas atau ensemble model dapat dipertimbangkan untuk mengurangi jumlah false positive tanpa mengorbankan *Recall* yang telah dicapai.
3. Penambahan mekanisme penjelasan visual (explainability) pada aplikasi, seperti visualisasi Grad-CAM yang telah digunakan pada tahap analisis di Bab IV, dapat diintegrasikan langsung ke dalam antarmuka aplikasi *Android* untuk meningkatkan kepercayaan pengguna terhadap hasil prediksi yang ditampilkan.
4. Penelitian selanjutnya disarankan untuk melakukan validasi model menggunakan teknik cross-validation dengan beberapa variasi random seed, mengingat penelitian ini hanya menggunakan satu seed (seed=42) sehingga estimasi variabilitas performa model belum dapat diketahui secara pasti.
5. Pengujian model pada dataset eksternal di luar *HAM10000*, seperti ISIC 2019 atau ISIC 2020, perlu dilakukan untuk mengevaluasi kemampuan generalisasi model terhadap variasi sumber data, kondisi pencahayaan, dan populasi pasien yang berbeda.
6. Perluasan cakupan klasifikasi dari dua kelas (*Melanoma* dan *Nevus*) menjadi multi-kelas yang mencakup seluruh tujuh kategori lesi kulit pada dataset *HAM10000* dapat menjadi arah penelitian lanjutan, mengingat penelitian ini secara sengaja membatasi cakupan pada klasifikasi biner untuk fokus pada perbandingan metode penanganan ketidakseimbangan data.
7. Validasi performa aplikasi pada citra yang diambil langsung oleh pengguna awam menggunakan kamera smartphone dalam kondisi lapangan yang bervariasi, seperti pencahayaan tidak terkontrol dan sudut pengambilan yang tidak standar, perlu dilakukan secara sistematis, mengingat pengujian pada penelitian ini masih menggunakan citra dari dataset yang dikumpulkan dalam kondisi dermoskopi terkontrol.
8. Penelitian lanjutan dapat mengeksplorasi penerapan teknik penanganan ketidakseimbangan data lain yang belum dibandingkan pada penelitian ini, seperti varian SMOTE yang lebih baru (Borderline-SMOTE atau ADASYN), maupun pendekatan ensemble learning untuk melihat apakah kombinasi beberapa metode

dapat menghasilkan *Recall* yang lebih tinggi tanpa mengorbankan *Precision* secara signifikan.