

Deteksi Jenis Jerawat Berbasis Android menggunakan Ensemble Deep Learning dengan Optimasi Layer Parsial

Hanggoro Aji Al Kautsar¹, Waeisul Bismi^{2*}, Deny Novianti³,
Muhammad Qommarudin⁴

^{1,2} Program Studi Informatika, Universitas Bina Sarana Informatika

³ Program Studi Teknologi Informasi, Universitas Bina Sarana Informatika

⁴ Program Studi Informatika, Universitas Nusa Mandiri

*Email: Waeisul.wbn@bsi.ac.id

Abstrak

Klasifikasi jenis jerawat seperti *whitehead*, *blackhead*, *papula*, *pustula*, dan *nodula* secara tepat sangat penting untuk menentukan perawatan yang sesuai. Penelitian ini mengembangkan sistem deteksi otomatis berbasis *Android* menggunakan pendekatan *ensemble Deep Learning* dan optimasi *layer parsial* untuk mengatasi tantangan visualisasi yang serupa antar jenis jerawat. Model *MobileNet* dan *EfficientNetB0* digunakan sebagai *base model*, kemudian digabungkan menggunakan teknik *majority voting* dan *weighted averaging*. Dataset yang digunakan terdiri dari 6.875 citra jerawat yang telah melalui proses *preprocessing* dan *augmentasi*. Untuk meningkatkan efisiensi dan mencegah *overfitting*, dilakukan *freeze layer* pada bagian awal model dan *fine-tuning* hanya pada layer akhir. Model dengan performa terbaik kemudian dikonversi ke format *TensorFlow Lite (TFLite)* dan diintegrasikan ke dalam aplikasi *Android*. Aplikasi memungkinkan pengguna mengklasifikasikan jerawat secara *real-time* melalui kamera atau galeri. Hasil evaluasi menunjukkan bahwa model *ensemble* memiliki akurasi dan stabilitas lebih baik dibandingkan model tunggal, serta waktu inferensi yang cepat di perangkat *Android*. Sistem ini menawarkan solusi praktis dan akurat bagi pengguna umum dan tenaga medis dalam mendeteksi jenis jerawat.

Kata kunci: Deteksi jerawat, Deep Learning, Ensemble learning, Optimasi layer parsial, TensorFlow Lite, Android.

Abstract

Accurate classification of acne types is essential for determining the appropriate treatment. This study developed an automatic detection system based on Android using an ensemble Deep Learning approach and partial layer optimization to address the visual similarity challenges among different acne types. MobileNet and EfficientNetB0 were used as base models, then combined using majority voting and weighted averaging techniques. The dataset used consisted of 6,875 acne images that underwent preprocessing and augmentation. To improve efficiency and prevent overfitting, early layers of the models were frozen, and fine-tuning was applied only to the top layers. The best-performing model was then converted into TensorFlow Lite (TFLite) format and integrated into an Android application. The application allows users to classify acne in real time using either the camera or gallery. Evaluation results showed that the ensemble model offered better accuracy and stability compared to individual models, with fast inference times on Android devices. This system provides a practical and accurate solution for both general users and medical professionals in detecting acne types.

Keywords : Acne detection, Deep Learning, Ensemble learning, Partial layer optimization, TensorFlow Lite, Android

PENDAHULUAN

Masalah jerawat (*acne vulgaris*) merupakan gangguan kulit yang paling umum dialami oleh remaja dan dewasa muda, yang dapat berdampak pada aspek fisik maupun psikologis penderita. Klasifikasi jenis jerawat seperti komedo, *papula*, *pustula*, dan nodul penting dilakukan untuk menentukan perawatan yang tepat. Namun, diagnosis jenis jerawat secara manual memerlukan keahlian dermatologis dan belum tersedia secara luas di masyarakat. Oleh karena itu, pengembangan sistem otomatis berbasis kecerdasan buatan menjadi sangat relevan dan dibutuhkan (Muhaba et al., 2022).

Kemajuan teknologi pengolahan citra dan *Deep Learning* telah membuka peluang besar dalam bidang dermatologi digital. Beberapa penelitian menunjukkan bahwa model *convolutional neural networks* mampu mendeteksi dan mengklasifikasi kondisi kulit dengan akurasi yang tinggi, bahkan mendekati kemampuan dokter ahli (Durgabhavani et al., 2023). Namun, tantangan tetap ada dalam mengklasifikasi berbagai jenis jerawat yang memiliki kemiripan visual tinggi, terutama pada pencahayaan dan warna kulit yang bervariasi (Javaid & Ubaid, 2024).

Ensemble learning merupakan pendekatan yang menjanjikan untuk meningkatkan akurasi model klasifikasi (Bhargavi et al., 2023). Dengan menggabungkan prediksi dari beberapa model CNN, sistem dapat mengurangi variansi dan bias, sehingga hasil klasifikasi menjadi lebih stabil dan andal. Dalam konteks klasifikasi jerawat, pendekatan *ensemble* dapat memperbaiki kelemahan model tunggal yang mungkin gagal mengenali fitur halus pada citra (Liu et al., 2022).

Selain itu, penggunaan teknik optimasi *layer* parsial menjadi strategi penting dalam *transfer learning*. Dengan membekukan beberapa *layer* awal dari model *pretrained* dan hanya melatih *layer* tertentu yang relevan dengan domain target, model dapat mencapai konvergensi lebih cepat dan menghindari *overfitting*, terutama pada dataset terbatas (Riaz et al., 2023). Teknik ini sangat efektif ketika digunakan pada model *ensemble* dengan arsitektur yang berbeda (De et al., 2024).

Penggunaan perangkat *mobile* seperti *Android* untuk implementasi model deteksi jerawat memberikan nilai praktis yang tinggi. *Android* merupakan sistem operasi *mobile* paling

populer di dunia, sehingga memungkinkan akses luas ke teknologi deteksi jerawat otomatis. Namun, keterbatasan sumber daya komputasi pada perangkat *mobile* menuntut adanya optimasi model, termasuk konversi ke format *TensorFlow Lite (TFLite)* dan teknik *pruning* atau *quantization* (Tanantong et al., 2024).

Dengan latar belakang tersebut, penelitian ini bertujuan untuk merancang dan mengembangkan sistem deteksi jenis jerawat berbasis *Android* menggunakan pendekatan *ensemble Deep Learning* dengan optimasi *layer* parsial. Sistem ini diharapkan mampu memberikan klasifikasi jenis jerawat secara cepat, akurat, dan praktis bagi pengguna umum maupun klinis. Kontribusi utama dari penelitian ini meliputi desain arsitektur *ensemble*, strategi *fine-tuning layer* parsial, dan *deployment* model ke dalam aplikasi *Android* berbasis *TFLite* yang efisien.

TINJAUAN PUSTAKA

Beberapa studi sebelumnya telah mengembangkan aplikasi deteksi jerawat berbasis *mobile*, namun masih terbatas pada klasifikasi biner (berjerawat atau tidak) atau jenis yang terbatas, serta belum mengintegrasikan pendekatan *ensemble* dan optimasi *layer* parsial secara komprehensif seperti pada penelitian (Isa & Mangshor, 2021) menggunakan metode teknologi *Deep Learning*, *YOLOv4* dalam mendeteksi dan mengenali jenis jerawat, kemudian pada penelitian (Sudana Putra et al., 2021) hanya menggunakan metode *Convolutional neural network* dalam mendeteksi otomatis 4 jenis jerawat, adapun pada penelitian (Tutiani & Wati, 2022) menggunakan metode CNN serta model *Efficientdet-lite* dan aplikasi hanya dapat bekerja pada *smartphone* dengan versi minimal *nougat*. Adapun (Ramadhani et al., 2024) hanya menerapkan model CNN *MobilenetV2* untuk mengklasifikasikan tingkat keparahan jenis jerawat.

Penggunaan *ensemble learning* dalam klasifikasi citra memberikan keunggulan dalam hal generalisasi dan robust terhadap variasi data. Metode seperti *majority voting*, *stacking*, dan *averaging* pada beberapa arsitektur CNN misalnya *Xception*, *InceptionV3*, *Mobilenet*, *Densenet*, dan *VGG16* telah digunakan dalam berbagai penelitian untuk meningkatkan performa klasifikasi pada domain medis (Fajarendra, 2025). *Ensemble* model terbukti

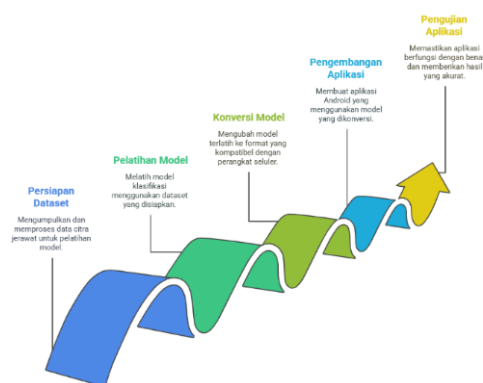
dapat mengurangi *overfitting* dan memberikan hasil yang lebih stabil dibandingkan model tunggal.

Namun, tantangan utama pada implementasi model *Deep Learning* di perangkat *Android* adalah keterbatasan sumber daya komputasi. Oleh karena itu, optimasi model menjadi krusial. Salah satu pendekatan yang relevan adalah optimasi *layer* parsial, yaitu strategi yang hanya melakukan pelatihan atau *fine-tuning* pada sebagian *layer* dari model pre-trained (Kaya & Gürsoy, 2023), sehingga dapat mengurangi kompleksitas model tanpa mengorbankan akurasi secara signifikan (Agustiani et al., 2024). Strategi ini juga sangat mendukung pengembangan aplikasi *mobile real-time* karena model yang dihasilkan lebih ringan dan cepat dieksekusi.

Di sisi lain, pengembangan aplikasi deteksi jerawat berbasis *Android* juga memerlukan integrasi antara model AI dengan antarmuka pengguna dan modul kamera secara efisien. *TensorFlow Lite* menjadi salah satu framework populer (David et al., 2021) yang mendukung *deployment* model *Deep Learning* di perangkat *mobile* (*TensorFlow Lite*, 2022) karena kemampuannya melakukan optimasi kuantisasi dan konversi model menjadi format yang ringan (Bismi, Riana, et al., 2024). Dengan menggabungkan pendekatan *ensemble Deep Learning* dan optimasi *layer* parsial dalam konteks pengembangan aplikasi *Android*, penelitian ini bertujuan untuk memberikan kontribusi dalam menciptakan sistem deteksi jerawat yang akurat, efisien, dan mudah diakses oleh masyarakat luas.

METODE

Pada Gambar 1 merupakan alur metode yang digunakan dalam penelitian yang dibagi menjadi beberapa tahap utama.



Gambar 1. Alur Metode Penelitian

Persiapan Dataset

Tahapan pertama dalam penelitian ini adalah persiapan dataset dan preprocessing. Dataset dapat diperoleh dari sumber terbuka seperti (DermNet, 2025). Kemudian dilakukan pelabelan jenis jerawat secara manual agar data bisa diteliti secara valid yang divertifikasi oleh pakar dermatologi.

Setelah pelabelan data, dilakukan preprocessing yang meliputi *resize* citra agar sesuai dengan ukuran input model 224x224 piksel (Wahyudi et al., 2023), konversi *RGB* to *BGR* (Bismi, Novianti, et al., 2024), normalisasi nilai piksel ke rentang 0–1 serta teknik *one-hot encoding* dan augmentasi data (Bismi et al., 2021). Setelah itu, dataset dibagi menjadi tiga bagian yaitu data pelatihan (80%), validasi (10%), dan pengujian (10%).

Pelatihan Model

Tahap kedua adalah perancangan dan pelatihan model *Deep Learning*. Dalam tahap ini digunakan arsitektur model dasar (*base models*) seperti *MobileNet* dan *EfficientNetB0* yang telah dilatih sebelumnya (*pretrained*). Strategi *ensemble* diterapkan untuk menggabungkan keunggulan masing-masing model melalui metode seperti majority voting, weighted averaging, dan stacking. Untuk meningkatkan efisiensi dan menghindari *overfitting*, dilakukan optimasi *layer* parsial dengan teknik *freeze layer*, yaitu membekukan sebagian *layer* awal dan hanya melakukan *fine-tuning* pada *layer* atas serta mengimplementasikan teknik *EarlyStopping* dan teknik *ReduceLROnPlateau*.

Eksperimen dilakukan untuk menentukan kombinasi *freeze -unfreeze layer* yang optimal bagi setiap model. Evaluasi model dilakukan berdasarkan metrik (Bismi, Riana, et al., 2024) diantaranya akurasi, *precision*, *recall*, dan *F1-score*, serta didukung oleh visualisasi *confusion matrix* guna menilai performa klasifikasi masing-masing model maupun hasil *ensemble*.

Konversi Model

Setelah mendapatkan model dengan performa terbaik, tahap ketiga adalah konversi model ke format yang dapat dijalankan pada perangkat *mobile*. Model dilatih di format Keras kemudian dikonversi ke *TensorFlow Lite* (TFLite) menggunakan *TFLite Converter* (*TensorFlow Lite Guide*, 2022). Konversi dilakukan untuk masing-masing model *base* maupun model *ensemble*. Untuk meningkatkan

efisiensi dan memperkecil ukuran model di perangkat *Android*, diterapkan teknik optimasi seperti *quantization* ke format int8 atau float16. Setelah konversi, model diuji kembali untuk memastikan bahwa performanya tetap konsisten dengan versi aslinya.

Pengembangan Aplikasi

Tahapan keempat mencakup pengembangan aplikasi *Android*. Aplikasi dibangun menggunakan *Android Studio* dengan bahasa pemrograman *Java* dan *XML*. Model *TFLite* diintegrasikan ke dalam aplikasi menggunakan *TFLite Interpreter* (Fadlilah et al., 2021). Aplikasi dilengkapi fitur input citra melalui kamera maupun galeri, sehingga pengguna dapat mengambil atau memilih foto jerawat secara langsung. Proses inferensi dilakukan di dalam aplikasi untuk mengklasifikasikan jenis jerawat berdasarkan model *TFLite*. Hasil klasifikasi ditampilkan dalam bentuk nama jenis jerawat beserta *confidence score*. Aplikasi juga memberikan informasi edukatif mengenai berbagai jenis jerawat untuk membantu pemahaman pengguna.

Pengujian Aplikasi

Terakhir, dilakukan pengujian aplikasi dan evaluasi menyeluruh. Pengujian dilakukan pada berbagai perangkat *Android* untuk memastikan kompatibilitas dan performa aplikasi di berbagai spesifikasi hardware. Evaluasi fokus pada waktu inferensi model (kecepatan prediksi), stabilitas aplikasi, serta kemudahan penggunaan oleh pengguna awam. Evaluasi akhir memastikan bahwa seluruh sistem — mulai dari model *Deep Learning* hingga antarmuka pengguna — telah berjalan optimal dan sesuai dengan standar kinerja dan usability yang diharapkan.

HASIL DAN PEMBAHASAN

Hasil Persiapan Dataset

Hasil persiapan mendapati dataset seluruh citra dengan total 6.875 citra jerawat yang digunakan dalam penelitian, kemudian dilabeli secara manual berdasarkan kategori jenis jerawat seperti *whitehead*, *blackhead*, *nodula*, *papula*, dan *pustula*. Pada Tabel 1, dapat dilihat jumlah data citra dari masing-masing label.

Tabel 1. Data Citra Jerawat

Nama Citra	Jumlah
<i>Whitehead comedo</i>	1.375 Citra
<i>Blackhead comedo</i>	1.375 Citra
<i>Nodula</i>	1.375 Citra
<i>Papula</i>	1.375 Citra
<i>Pustula</i>	1.375 Citra

Berikut citra jerawat pada Gambar 2. yang telah divaliditas label serta dikonfirmasi oleh pakar dermatologi untuk menjamin keabsahan dataset.



Gambar 2. Citra Jerawat

Tahapan preprocessing dilakukan secara sistematis agar data siap digunakan oleh model *Deep Learning*. Citra diubah ukurannya menjadi 224x224 piksel, dikonversi dari *RGB* ke *BGR*, dan dinormalisasi ke rentang nilai 0–1 serta teknik *one-hot encoding* yang terlihat pada Gambar 3.

```
# PRE-PROCESSING
ResizeImage = tuple((224, 224))
X_train = []
y_train = []
for label in train_labels:
    dir = os.path.join(train_path, label)
    for file in glob.glob(dir + "/*.jpg"):
        im = cv2.imread(file)
        im = cv2.resize(im, ResizeImage, interpolation=cv2.INTER_AREA)
        im = cv2.cvtColor(im, cv2.COLOR_BGR2RGB)
        X_train.append(im)
        y_train.append(label)
    print("[status] telah selesai baca image untuk folder: {}".format(label))

X_train = np.array(X_train)
y_train = np.array(le.transform(y_train))
print("[status] telah selesai baca image untuk seluruh isi folder train")

# Normalisasi
X_train = X_train.astype('float32') / 255
X_test = X_test.astype('float32') / 255

# One-Hot Encoding
num_classes = len(np.unique(y_train))
y_train = to_categorical(y_train, num_classes)
y_test = to_categorical(y_test, num_classes)
```

Gambar 3. Tahap Pre-Processing

Augmentasi dilakukan melalui metode *horizontal flipping*, *zooming*, serta penyesuaian *brightness*, yang bertujuan menambah keragaman data dan meminimalisasi risiko *overfitting*. Dataset kemudian dibagi menjadi tiga subset yang diantaranya 80% untuk pelatihan, 10% untuk validasi, dan 10% untuk pengujian. Adapun rinciannya dapat terlihat pada Tabel 2.

Tabel 2. Data Citra Jerawat

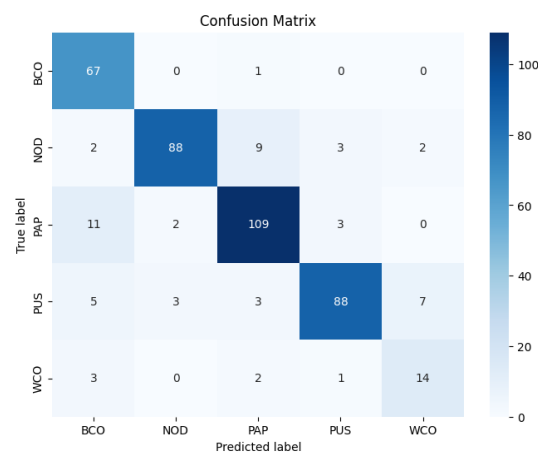
Nama Citra	Persentase	Jumlah
Data Train	80%	5.429 Citra
Data Test	10%	678 Citra
Data Validation	10%	678 Citra

Pembagian ini memastikan bahwa model tidak hanya belajar dari data pelatihan, tetapi juga mampu diuji pada data yang belum pernah dilihat sebelumnya.

Hasil Pelatihan Model

Pada hasil pelatihan model yang menggunakan dua arsitektur *pretrained* utama yaitu *MobileNet* dan *EfficientNetB0*, yang keduanya terkenal karena efisiensi dan akurasi tinggi dalam klasifikasi citra. Strategi *ensemble learning* diterapkan dengan pendekatan *majority voting* dan *weighted averaging*, untuk menggabungkan kekuatan masing-masing model. Dalam proses pelatihan, dilakukan strategi *freeze layer* $[-20]$ pada *Brach MobileNet* dan *freeze layer* $[-15]$ pada *Brach EfficientNetB0* yaitu membekukan sebagian besar layer awal dan hanya melakukan *fine-tuning* pada layer akhir agar pelatihan lebih efisien dan menghindari *overfitting*.

Diterapkan pula mekanisme *EarlyStopping* untuk menghentikan pelatihan saat tidak ada peningkatan performa, serta *ReduceLROnPlateau* untuk menurunkan *learning rate* ketika stagnasi terjadi selama beberapa *epoch*. Pelatihan dilakukan dengan *batch size* 32 dan maksimum 20 *epoch*. Hasil Evaluasi dilakukan menggunakan *confusion matrix* diantaranya akurasi, *precision*, *recall*, dan *F1-score* serta *kappa score* terlihat pada Gambar 4 dan disertai laporan klasifikasi terlihat pada Gambar 5.



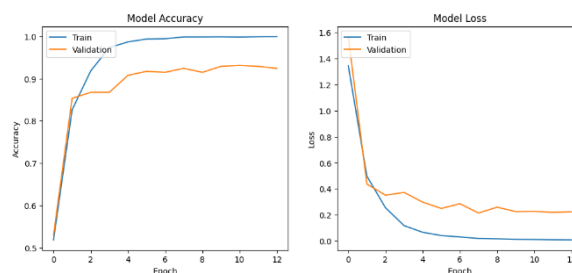
Gambar 4. Confusion matrix

Classification Report:				
	precision	recall	f1-score	support
BCO	0.76	0.99	0.86	68
NOD	0.95	0.85	0.89	104
PAP	0.88	0.87	0.88	125
PUS	0.93	0.83	0.88	106
WCO	0.61	0.70	0.65	20
accuracy			0.87	423
macro avg	0.82	0.85	0.83	423
weighted avg	0.88	0.87	0.87	423

Cohen's Kappa Score: 0.8243

Gambar 5. Laporan klasifikasi

Adapun visualisasi pelatihan model atau graph terlihat pada Gambar 5.



Gambar 5. Visual Pelatihan Model

Hasil menunjukkan bahwa model *ensemble* memiliki performa yang lebih stabil dan akurat dibanding model tunggal, terutama dalam mengklasifikasikan jenis jerawat yang mirip secara visual seperti *papula* dan *pustula* terlihat pada Gambar 6.



Gambar 6. Visual Prediksi Citra

Hasil Konversi Model

Model dengan extension (.h5) merupakan performa terbaik yang kemudian di konversi ke TensorFlow Lite (TFLite) agar dapat digunakan dalam perangkat *Android*. Diterapkan teknik post-training quantization menggunakan (*tf.lite.Optimize.DEFAULT*) yang menghasilkan pengurangan ukuran file model secara signifikan tanpa menurunkan performa secara berarti. Hasil pengukuran menunjukkan bahwa model quantized mengalami pengurangan ukuran hingga lebih dari 74.72% dibandingkan model aslinya, menjadikannya lebih ringan dan cepat untuk dijalankan di perangkat mobile. Adapun perbandingan ukurannya modelnya dapat dilihat pada Tabel 3.

Tabel 3. Perbandingan Ukuran Model

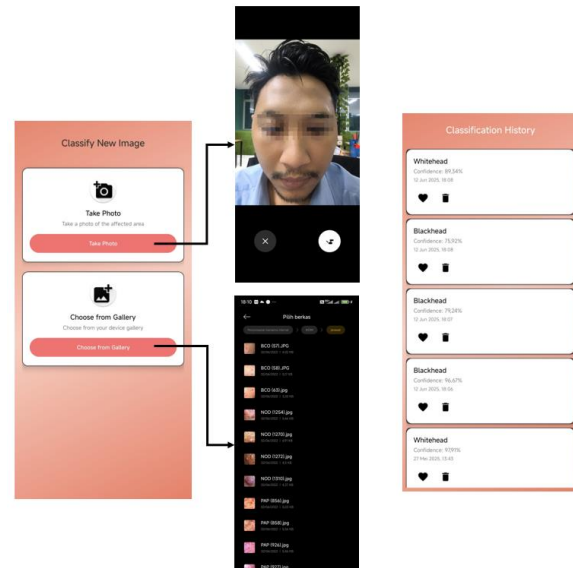
Komponen	Ukuran (MB)	Tipe Data
Original TFLite	249 MB	Float32
Quantized TFLite	62,93 MB	Int8 / Float16

Hasil Pengembangan Aplikasi

Aplikasi *Android* untuk klasifikasi jerawat dikembangkan menggunakan *Android Studio* versi *Meerkat Feature Drop* dengan bahasa pemrograman *Java* dan *XML*. Model *TFLite* diintegrasikan ke dalam aplikasi menggunakan *TFLite Interpreter*, yang memungkinkan inferensi dilakukan secara lokal di perangkat.

Aplikasi menyediakan dua fitur utama untuk input citra: kamera dan galeri, sehingga

pengguna bisa memotret langsung atau memilih dari penyimpanan. Setelah input, proses klasifikasi dilakukan secara real-time, dan hasil ditampilkan pada halaman *classification history* dalam bentuk nama jenis jerawat beserta nilai *confidence*, terlihat pada Gambar 7.



Gambar 7. Tampilan Aplikasi Deteksi Jerawat

Sebagai tambahan, aplikasi menyertakan informasi edukatif mengenai masing-masing jenis jerawat yang dikenali, untuk meningkatkan pemahaman pengguna tentang kondisi kulit mereka, terlihat pada Gambar 8.



Gambar 8. Tampilan Informasi Jenis Jerawat

Hasil Pengujian Aplikasi

Aplikasi diuji pada beberapa perangkat *Android* dengan berbagai spesifikasi untuk

memastikan kompatibilitas dan performa. Hasil pengujian menunjukkan bahwa aplikasi mampu menjalankan inferensi dalam waktu kurang dari 1 detik per gambar, menunjukkan efisiensi yang tinggi.

Stabilitas aplikasi diuji dengan berbagai skenario, termasuk rotasi layar, pemrosesan batch gambar, dan kondisi cahaya rendah. Tidak ditemukan crash atau lag yang signifikan. Dari sisi usability, antarmuka pengguna dirancang sederhana dan intuitif, memudahkan penggunaan bahkan bagi pengguna awam.

Evaluasi akhir menunjukkan bahwa seluruh pipeline — dari dataset hingga aplikasi — telah berjalan dengan optimal dan memenuhi kriteria kinerja serta kenyamanan pengguna.

SIMPULAN

Penelitian ini berhasil mengembangkan sistem deteksi jenis jerawat berbasis *Android* yang akurat dan efisien dengan memanfaatkan pendekatan *ensemble Deep Learning* dan optimasi *layer parsial*. Model gabungan *MobileNet* dan *EfficientNetB0* menunjukkan peningkatan performa klasifikasi dibandingkan model tunggal sebesar 90%, terutama dalam membedakan jenis jerawat yang memiliki kemiripan visual. Optimasi melalui teknik *freeze layer* dan *quantization* berhasil mengurangi ukuran model hingga 74,72% tanpa penurunan performa signifikan, memungkinkan *deployment* yang optimal di perangkat mobile. Aplikasi *Android* yang dikembangkan dapat melakukan klasifikasi secara *real-time* dan memiliki antarmuka yang mudah digunakan. Hasil evaluasi membuktikan bahwa sistem ini layak digunakan baik oleh pengguna umum maupun dalam konteks klinis untuk mendukung diagnosis awal jenis jerawat secara otomatis. Selanjutnya untuk saran penelitian lanjutan dapat menerapkan teknik *NAS (Neural Architectur Search)* dan mengimplementasikan *Real-time video processing* agar dapat membantu pengguna memindai area kulit secara dimanis serta mengintegrasikan dengan fitur *Telemedicine* untuk konsultasi langsung dengan dokter kulit.

DAFTAR PUSTAKA

Agustiani, S., Aryanti, R., Khotimatul Wildah, S., Arifin, Y. T., Marlina, S., & Misriati, T. (2024). Optimisasi Model Deep Learning untuk Deteksi Penyakit Daun Tebu dengan Fine-Tuning MobileNetV2. *Journal of*

Informatics Management and Information Technology, 4(4), 150–157. <https://doi.org/10.47065/jimat.v4i4.411>

Bhargavi, M., Renugadevi, R., Sivabalan, S., Phani, P., Ganesh, J., & Bhanu, K. (2023). Ensemble Learning for Skin Lesion Classification: A Robust Approach for Improved Diagnostic Accuracy (ELSLC). *3rd International Conference on Innovative Mechanisms for Industry Applications, ICIMIA 2023 - Proceedings*, 390–395.

<https://doi.org/10.1109/ICIMIA60377.2023.10425888>

Bismi, W., Dwiza, R., & Sri, H. (2021). *Classification of Myopia Eye Diseases Using* (Vol. 15, Issue 2).

Bismi, W., Novianti, D., & Qomaruddin, M. (2024). Analisis Perbandingan Klasifikasi Citra Genus *Panthera* dengan Pendekatan Deep learning Model MobileNet. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, 6(1). <https://publikasiilmiah.unwahas.ac.id/JINRPL/article/view/9037/pdf>

Bismi, W., Riana, D., & Hewiz, A. S. (2024). Disease Identification on Fig Leaf Images Using Deep Learning Method. *International Journal of Advanced Science Computing and Engineering*, 6(2), 57–63. <https://doi.org/10.62527/ijasce.6.2.203>

David, R., Duke, J., Jain, A., Reddi, V. J., Jeffries, N., & Li, J. (2021). Tensorflow Lite Micro: Embedded Machine Learning On Tynyml Systems. *MLSys Proceedings*, 1–9.

De, A., Mishra, N., & Chang, H. T. (2024). An approach to the dermatological classification of histopathological skin images using a hybridized CNN-DenseNet model. *PeerJ Computer Science*, 10. <https://doi.org/10.7717/peerj-cs.1884>

DermNet. (2025). “Acne Affecting the Face Image,.” DermNet New Zealand Trust. <https://dermnetnz.org/topics/acne-face-images>

Durgabhavani, B., Chandana, B., Sandhyarani, Lavanya, G., Srikanth, G., & Bhaskar, N. (2023). Classification of Skin Disease using CNN. *2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques, EASCT 2023*.

- <https://doi.org/10.1109/EASCT59475.2023.10393477>
- Fadlilah, U., Mahamad, A. K., & Handaga, B. (2021). The Development of Android for Indonesian Sign Language Using Tensorflow Lite and CNN: An Initial Study. *Journal of Physics: Conference Series*, 1858(1). <https://doi.org/10.1088/1742-6596/1858/1/012085>
- Fajarendra, Y. iza. (2025). *PENGUNAAN ENSEMBLE LEARNING UNTUK KLASIFIKASI TINGKAT RISIKO ABRASI BERBASIS CITRA SATELIT SENTINEL-2*. UIN Sunan Kalijaga Yogyakarta.
- Isa, N. A. M., & Mangshor, N. N. A. (2021). Acne Type Recognition for Mobile-Based Application Using YOLO. *Journal of Physics: Conference Series*, 1962(1). <https://doi.org/10.1088/1742-6596/1962/1/012041>
- Javaid, S., & Ubaid, M. T. (2024). Skin Disease Detection Using Transfer Learning. *2024 International Visualization, Informatics and Technology Conference (IVIT)*, 122–127. <https://doi.org/10.1109/IVIT62102.2024.10692509>
- Kaya, Y., & Gürsoy, E. (2023). A MobileNet-based CNN model with a novel fine-tuning mechanism for COVID-19 infection detection. *Soft Computing*, 27(9), 5521–5535. <https://doi.org/10.1007/s00500-022-07798-y>
- Liu, S., Fan, Y., Duan, M., Wang, Y., Su, G., Ren, Y., Huang, L., & Zhou, F. (2022). AcneGrader: An ensemble pruning of the deep learning base models to grade acne. *Skin Research and Technology*, 28(5), 677–688. <https://doi.org/10.1111/srt.13166>
- Muhaba, K. A., Dese, K., Aga, T. M., Zewdu, F. T., & Simegn, G. L. (2022). Automatic skin disease diagnosis using deep learning from clinical image and patient information. *Skin Health and Disease*, 2(1). <https://doi.org/10.1002/ski2.81>
- Ramadhani, F., Rahardiantoro, S., & Masjkur, M. (2024). Acne Severity Classification Study Using Convolutional Neural Network Algorithm with MobileNetV2 Architecture. *Indonesian Journal of Statistics and Its Applications*, 8(2), 112–128. <https://doi.org/10.29244/ijisa.v8i2p112-128>
- Riaz, S., Naeem, A., Malik, H., Naqvi, R. A., & Loh, W. K. (2023). Federated and Transfer Learning Methods for the Classification of Melanoma and Nonmelanoma Skin Cancers: A Prospective Study. In *Sensors (Basel, Switzerland)* (Vol. 23, Issue 20). <https://doi.org/10.3390/s23208457>
- Sudana Putra, F., Otomatis Jerawat Wajah, D., & Kurniawan, M. P. (2021). Deteksi Otomatis Jerawat Wajah Menggunakan Metode Convolutional Neural Network (CNN). *JIFOTECH (JOURNAL OF INFORMATION TECHNOLOGY)*, 1(2).
- Tanantong, T., Chalarak, N., Pandecha, P., Tanantong, K., & Srijiranon, K. (2024). MOBILE-BASED DEEP LEARNING FRAMEWORK FOR CLASSIFYING COMMON SKIN DISEASES IN THAILAND. *ICIC Express Letters, Part B: Applications*, 15(5), 495–503. <https://doi.org/10.24507/icicelb.15.05.495>
- TensorFlow Lite Guide. (2022). TensorFlow. <https://www.tensorflow.org/lite/guide>
- Tutiani, & Wati, S. (2022). Implementation of Convolutional Neural Network (CNN) in Android-Based Acne Detection Applications. *2022 International Conference on Information Technology Research and Innovation (ICITRI)*, 151–155. <https://doi.org/10.1109/ICITRI56423.2022.9970231>
- Wahyudi, M., Bismi, W., Firmansyah, Raharjo, M., Pujiastuti, L., & Untung, R. (2023). Gender Recognition Based on Face Image Using Deep Learning Method. *11th International Conference on Cyber and IT Service Management (CITSM)*, 1–6. <https://doi.org/10.1109/CITSM60085.2023.10455315>