



Comparative Analysis of MobileNet Architectures for Acne Type Classification on Facial Images

Analisis Komparatif Arsitektur *MobileNet* untuk Klasifikasi Tipe Jerawat pada Citra Wajah

Sukma Syafika^{1*}, Susi Marianingsih²

^{1,2}Jurusan Teknik Informatika, Fakultas Ilmu Komputer dan Manajemen,
Universitas Sains dan Teknologi Jayapura, Indonesia

E-Mail: ¹sukmasyafikao@gmail.com, ²smarianingsih@gmail.com

Received Feb 15th 2026; Revised Mar 06th 2026; Accepted Apr 24th 2026; Available Online Apr 28th 2026

Corresponding Author: Sukma Syafika

Copyright ©2026 by Authors, Published by Institut Riset dan Publikasi Indonesia (IRPI)

Abstract

Acne vulgaris is one of the common facial skin disorders experienced by many individuals, which can cause both physical and psychological impacts on sufferers. Manual identification of acne types requires careful visual inspection, which can lead to diagnostic errors. Therefore, an automated system is needed to accurately identify and classify acne types. This research aims to build an image-based acne classification model using a Deep Learning approach with the MobileNet architecture. The dataset used consists of 3,500 acne images across five classes: blackhead, whitehead, papules, pustules, and cysts. The dataset is split into training, validation, and test sets at 80:10:10. Prior to model training, the images are preprocessed using the OpenCV library, including resizing, normalization, and data augmentation. The model is developed using the TensorFlow and Keras frameworks. This research compares the performance of three MobileNet architectures: MobileNetV1, MobileNetV2, and MobileNetV3. The test results show that MobileNetV3 achieves the best performance with an accuracy of 94.57%, precision of 0.95, recall of 0.95, and F1-score of 0.95. This research contributes to the comparative evaluation of MobileNet architectures to determine the most effective lightweight model for acne image classification.

Keywords: *Acne Vulgaris, Deep Learning, Image Classification, Image Processing, MobileNet*

Abstrak

Jerawat (*acne vulgaris*) adalah salah satu gangguan kulit wajah yang kerap dialami oleh banyak individu, serta dapat menimbulkan dampak fisik maupun psikologis bagi penderitanya. Proses identifikasi jenis jerawat secara manual memerlukan pengamatan visual yang teliti sehingga berpotensi menimbulkan kesalahan dalam proses diagnosis. Oleh sebab itu, diperlukan pengembangan sistem otomatis untuk mengidentifikasi dan mengklasifikasikan jenis jerawat secara lebih akurat. Penelitian ini dimaksudkan untuk membangun model klasifikasi jerawat berbasis citra menggunakan pendekatan *Deep Learning* dengan arsitektur *MobileNet*. Dataset yang digunakan berjumlah 3.500 citra jerawat yang terdiri dari lima kelas, yakni *blackhead*, *whitehead*, *papules*, *pustules*, dan *cyst*. Dataset dibagi menjadi data pelatihan, validasi, dan pengujian dengan perbandingan 80:10:10. Sebelum proses pelatihan model dilakukan, citra melalui tahap *pre-processing* yang meliputi *image resizing*, normalisasi citra, dan augmentasi data menggunakan pustaka *OpenCV*. Model dikembangkan menggunakan framework *TensorFlow* dan *Keras*. Penelitian ini membandingkan performa tiga arsitektur *MobileNet*, yaitu *MobileNetV1*, *MobileNetV2*, dan *MobileNetV3*. Hasil pengujian menunjukkan bahwa *MobileNetV3* menghasilkan performa terbaik dengan akurasi sebesar 94,57%, *precision* 0,95, *recall* 0,95, dan *F1-score* 0,95. Penelitian ini berkontribusi dalam evaluasi komparatif arsitektur *MobileNet* untuk menentukan model ringan yang paling efektif dalam klasifikasi citra jerawat.

Kata Kunci: *Acne Vulgaris, Deep Learning, Klasifikasi Citra, MobileNet, Pengolahan Citra.*

1. PENDAHULUAN

Kulit wajah merupakan salah satu bagian tubuh yang kerap menjadi sorotan utama dalam bidang perawatan kesehatan dan kecantikan [1]. Berbagai permasalahan kulit dapat terjadi pada area wajah, salah satunya adalah jerawat (*acne vulgaris*) yang sering dialami oleh banyak orang, terutama pada masa remaja

[2]. Jerawat merupakan kondisi peradangan pada folikel rambut dan kelenjar sebacea yang dipengaruhi oleh berbagai faktor, seperti peningkatan produksi sebum, aktivitas bakteri pada folikel kulit, serta perubahan hormon yang memicu terjadinya peradangan pada kulit [3]. Selain menimbulkan gangguan secara fisik, jerawat juga dapat memberikan pengaruh terhadap kondisi psikologis penderitanya. Kondisi ini dapat menyebabkan menurunnya rasa percaya diri, tekanan emosional, hingga gangguan psikologis seperti kecemasan dan depresi [4]. Menurut data yang dirilis oleh Kementerian Kesehatan Republik Indonesia, jerawat termasuk masalah kulit dengan tingkat prevalensi yang cukup tinggi, dengan angka kejadian sekitar 87,5%. Kasus jerawat paling banyak ditemukan pada kelompok usia remaja, dengan persentase sekitar 83–85% pada perempuan berusia 14–17 tahun dan sekitar 95–100% pada laki-laki berusia 16–19 tahun [2].

Seiring dengan perkembangan teknologi kecerdasan buatan, berbagai metode komputasi telah dikembangkan untuk membantu proses identifikasi dan analisis citra secara otomatis. Salah satu metode yang sering digunakan adalah *Machine Learning*, yaitu metode yang memungkinkan komputer mempelajari pola dari data yang tersedia sehingga dapat melakukan prediksi secara independen. Metode ini bekerja dengan menirukan cara manusia dalam mempelajari pola melalui proses *training* sehingga model mampu melakukan generalisasi terhadap data baru [5]. Seiring dengan perkembangan teknologi, muncul pendekatan *Deep Learning* sebagai salah satu cabang dari *Machine Learning*. Metode ini mampu merepresentasikan pola data yang lebih kompleks melalui berbagai lapisan pemrosesan, sehingga dapat meningkatkan akurasi dalam menghasilkan prediksi. Pendekatan ini banyak diterapkan dalam pengolahan citra karena mampu mengklasifikasikan gambar berdasarkan fitur visual yang dipelajari secara otomatis oleh model [6]. Dalam bidang pengolahan citra digital, *Convolutional Neural Network* (CNN) merupakan salah satu algoritma *Deep Learning* yang kerap dipakai. Metode ini dirancang untuk menangani data berbentuk citra dengan memanfaatkan lapisan konvolusi guna mengekstraksi berbagai fitur penting dari gambar secara efisien [7]. Salah satu arsitektur CNN yang banyak dimanfaatkan dalam tugas klasifikasi citra adalah *MobileNet*. Model ini dibuat dengan arsitektur ringan dan efisien melalui penerapan teknik *depthwise separable convolution* guna mengurangi jumlah parameter serta beban komputasi, sambil tetap menjaga performa klasifikasi yang optimal [8]. Dengan desain yang lebih efisien tersebut, *MobileNet* mampu menghasilkan performa klasifikasi yang baik dengan kebutuhan sumber daya komputasi yang lebih rendah.

Beberapa penelitian sebelumnya telah menerapkan metode *Deep Learning* untuk melakukan klasifikasi jenis jerawat. Salah satu penelitian menggunakan algoritma CNN untuk mengelompokkan jenis jerawat ke dalam beberapa kategori, seperti *blackhead*, *papules*, *pustules*, dan *whitehead*. Hasil studi tersebut mengindikasikan bahwa model CNN berhasil mencapai akurasi hingga 96,57%. Meskipun demikian, nilai *precision*, *recall*, dan *F1-score* yang dihasilkan masih tergolong rendah karena keterbatasan jumlah dataset serta ketidakseimbangan distribusi data pada setiap kelas. Selain itu, penerapan CNN konvensional umumnya memiliki kompleksitas komputasi yang cukup tinggi, sehingga memerlukan sumber daya komputasi yang besar selama proses pelatihan model [9].

Penelitian terdahulu umumnya masih menitikberatkan pada penggunaan satu arsitektur tertentu atau hanya melaporkan hasil performa model tanpa melakukan perbandingan yang terkontrol antara beberapa varian *MobileNet* pada dataset jerawat yang sama. Hal ini menunjukkan adanya kesenjangan penelitian, karena belum diketahui secara pasti arsitektur *MobileNet* mana yang paling optimal dalam menyeimbangkan performa klasifikasi dan efisiensi komputasi pada tugas klasifikasi jenis jerawat [10]. Oleh karena itu, penelitian ini merumuskan permasalahan utama berupa perbandingan kinerja *MobileNetV1*, *MobileNetV2*, dan *MobileNetV3* dengan menerapkan skema *preprocessing*, *transfer learning*, dan evaluasi yang seragam pada klasifikasi lima jenis citra jerawat.

Berbagai studi internasional menunjukkan bahwa keluarga *MobileNet* merupakan salah satu arsitektur CNN ringan yang prospektif untuk klasifikasi citra medis dan dermatologis karena mampu memberikan performa yang kompetitif dengan kebutuhan komputasi yang lebih efisien [10], [11]. *MobileNet* pertama kali diperkenalkan oleh Howard dkk. melalui konsep *depthwise separable convolution* untuk menekan kompleksitas komputasi [10]. Pengembangannya kemudian dilanjutkan oleh Sandler dkk. melalui *MobileNetV2* dengan mekanisme *inverted residual* dan *linear bottleneck* untuk meningkatkan kualitas representasi fitur [12]. Selanjutnya, Howard dkk. mengembangkan *MobileNetV3* dengan memadukan *neural architecture search* dan *squeeze-and-excitation* guna meningkatkan efisiensi sekaligus performa model [11]. Dalam bidang dermatologi, pendekatan *deep learning* ringan juga telah menunjukkan potensi yang baik untuk analisis penyakit kulit dan jerawat, khususnya pada sistem dengan keterbatasan sumber daya komputasi [18].

Berdasarkan penelitian terdahulu yang telah diuraikan, metode CNN terbukti efektif dalam melakukan klasifikasi citra, termasuk pada kasus klasifikasi jenis jerawat. Dengan mempertimbangkan hal tersebut, penelitian ini mengimplementasikan pendekatan *Deep Learning* berbasis CNN untuk mengenali dan mengklasifikasikan jenis jerawat berdasarkan citra wajah [13]. Sebelum proses pelatihan model dilakukan, citra terlebih dahulu melalui tahap *pre-processing* menggunakan pustaka OpenCV untuk meningkatkan kualitas citra dan menyiapkan data agar dapat digunakan secara optimal oleh model. Pada tahap ini dilakukan beberapa proses *preprocessing* citra seperti normalisasi, *resizing*, serta penyesuaian format citra sebelum

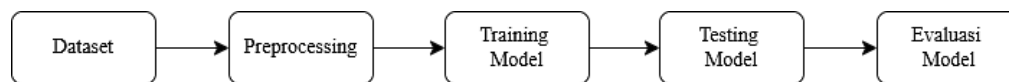
masuk ke tahap pelatihan. Model klasifikasi dibangun menggunakan framework TensorFlow dan Keras sebagai lingkungan implementasi *Deep Learning*. Dataset yang digunakan terdiri dari lima kategori jerawat, yaitu *blackhead*, *whitehead*, *papules*, *pustules*, dan *cyst*, yang dijadikan sebagai kelas dalam proses klasifikasi citra. Penelitian ini menggunakan arsitektur *MobileNet*, yaitu *MobileNetV1*, *MobileNetV2*, dan *MobileNetV3*.

Tujuan penelitian ini adalah untuk mengklasifikasikan jenis jerawat berdasarkan citra wajah serta membandingkan performa ketiga arsitektur *MobileNet* tersebut. Evaluasi performa model dilakukan dengan memanfaatkan metrik seperti akurasi, *presisi*, *recall*, dan *F1-score* untuk menentukan model yang memiliki kinerja paling unggul. Penelitian ini diharapkan berkontribusi pada pengembangan metode klasifikasi jenis jerawat berdasarkan citra wajah dengan memanfaatkan arsitektur *MobileNet* yang efisien serta memberikan perbandingan performa antara *MobileNetV1*, *MobileNetV2*, dan *MobileNetV3*.

2. METODE

Metodologi penelitian ini dirancang berdasarkan hasil kajian terhadap penelitian-penelitian sebelumnya yang membahas klasifikasi citra berbasis CNN, pendekatan *transfer learning*, dan pemanfaatan arsitektur *MobileNet* dalam analisis citra medis yang efisien [10]–[14]. Kajian tersebut digunakan sebagai dasar dalam menetapkan alur *preprocessing*, menentukan *MobileNetV1*, *MobileNetV2*, dan *MobileNetV3* sebagai model yang dibandingkan, serta memilih strategi *transfer learning*, *fine-tuning*, dan metrik evaluasi yang sesuai [12], [11]. Oleh karena itu, penyusunan rancangan eksperimen dalam penelitian ini memiliki dasar metodologis yang kuat karena mengacu pada temuan-temuan yang telah dibuktikan pada penelitian terdahulu.

Tahapan penelitian ditunjukkan pada Gambar 1, terdapat beberapa tahapan utama yang dilakukan. Tahapan tersebut dimulai dari pengumpulan dataset yang berisi citra jerawat, kemudian dilanjutkan dengan proses pelatihan model menggunakan metode CNN berbasis arsitektur *MobileNetV1*, *MobileNetV2*, dan *MobileNetV3*. Setelah proses pelatihan selesai, model kemudian diuji pada tahap pengujian (*testing*), dan tahap terakhir adalah proses evaluasi untuk menilai performa model.



Gambar 1. Tahapan Penelitian

2.1. Dataset

Dalam penelitian ini, dataset diambil dari platform Kaggle. Dataset yang dipakai mencakup 3.500 citra jerawat digital yang dikelompokkan menjadi lima jenis jerawat, yaitu *blackhead*, *whitehead*, *papules*, *pustules*, dan *cyst*. Dataset tersebut dibagi menjadi tiga bagian, yaitu pelatihan (*training*), validasi (*validation*), serta pengujian (*testing*) dengan proporsi 80:10:10. Tujuan dari pembagian ini adalah agar tahapan pelatihan, validasi, dan pengujian model menggunakan data yang tidak sama. Jumlah data pada setiap kelas dapat dilihat pada Tabel 1.

Tabel 1. Dataset Penelitian

No	Jenis Jerawat	Data		
		Training	Validasi	Testing
1	Blackhead	774	80	82
2	Whitehead	201	30	28
3	Papules	616	80	80
4	Pustules	594	80	80
5	Cyst	615	80	80

Gambar 2 menampilkan contoh citra jerawat dari dataset yang dipakai dalam penelitian ini untuk setiap kelas.

2.2. Preprocessing

Tahap *preprocessing* dilakukan guna mempersiapkan dataset supaya sesuai dengan kebutuhan model serta meningkatkan mutu data untuk pelatihan. Pada penelitian ini, tahapan *preprocessing* yang diterapkan terdiri dari beberapa proses sebagai berikut:

2.2.1. Image Resizing

Pada penelitian ini, seluruh citra dalam dataset disesuaikan ukurannya (*resize*) agar memiliki dimensi yang seragam. Model *MobileNet* menggunakan ukuran input citra sebesar 224×224 piksel, sehingga seluruh citra dalam dataset diubah ke ukuran tersebut sebelum digunakan dalam proses pelatihan model [15].

2.2.2. Normalisasi Citra

Pada proses normalisasi, setiap nilai piksel dibagi dengan 255.0 agar rentang nilai asal dari 0–255 berubah menjadi 0–1 [16]. Langkah tersebut bertujuan untuk menstandarisasi data masukan ke model, mempercepat proses konvergensi saat pelatihan, serta meminimalkan dampak variasi iluminasi pada gambar.

2.2.3. Augmentasi Data

Augmentasi data dalam penelitian ini dilaksanakan guna memperkaya keragaman data latih, sehingga model dapat mempelajari pola citra secara lebih baik dan meningkatkan kemampuan generalisasi model. Proses augmentasi dilakukan dengan menerapkan beberapa teknik transformasi citra yang bertujuan untuk menghasilkan variasi citra baru dari dataset yang ada [17]. Metode augmentasi yang digunakan meliputi *rotation range* = 20, yang berarti citra dapat diputar hingga 20 derajat. Selanjutnya diterapkan *width shift range* = 0.1 dan *height shift range* = 0.1, yang memungkinkan citra digeser secara *horizontal* maupun *vertikal* hingga 10% dari ukuran gambar asli. Selain itu, digunakan juga metode *zoom range* = 0.2 yang memungkinkan citra diperbesar atau diperkecil hingga 20% dari ukuran awal. Teknik *horizontal flip* juga diterapkan untuk membalik citra secara *horizontal* sehingga meningkatkan variasi data pelatihan. Penerapan berbagai teknik augmentasi ini bertujuan untuk memperkaya variasi citra dalam dataset sehingga model dapat mempelajari pola objek dengan lebih efektif serta meningkatkan kinerja klasifikasi terhadap data yang sebelumnya belum pernah digunakan.



Gambar 2. Dataset Citra Jerawat Berdasarkan Kelas

2.3. Pelatihan Model *MobileNet*

Penelitian ini memanfaatkan tiga arsitektur dari *MobileNet*, yaitu *MobileNetV1*, *MobileNetV2*, dan *MobileNetV3*. Ketiga model tersebut merupakan arsitektur CNN ringan yang dikembangkan untuk menghasilkan performa klasifikasi yang optimal dengan jumlah parameter yang relatif kecil, sehingga lebih efisien dalam proses komputasi [10]. Secara umum, arsitektur *MobileNet* tersusun dari lapisan konvolusi yang menggunakan teknik *depthwise separable convolution*, yaitu pemisahan proses konvolusi menjadi *depthwise convolution* dan *pointwise convolution*. Pendekatan ini mampu mengurangi jumlah parameter serta operasi komputasi secara signifikan dibandingkan konvolusi standar, tanpa mengurangi performa model secara berarti [18].

MobileNetV1 merupakan versi awal yang memperkenalkan konsep *depthwise separable convolution* untuk mengurangi kompleksitas komputasi [10]. Tabel 2 menunjukkan arsitektur *MobileNetV1*.

Tabel 2. Arsitektur *MobileNetV1*

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5 \times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5 \times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$

Type / Stride	Filter Shape	Input Size
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Selanjutnya, *MobileNetV2* mengembangkan arsitektur sebelumnya dengan memperkenalkan modul *inverted residual block* dan *linear bottleneck* yang bertujuan untuk mempertahankan informasi fitur secara lebih efektif selama proses ekstraksi [12]. Tabel 3 menunjukkan arsitektur *MobileNetV2*.

Tabel 3. Arsitektur *MobileNetV2*

Input	Operator	t	c	n	s
$224^2 \times 3$	Conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	Conv2d 1×1	-	1280	1	1
$7^2 \times 1280$	Avgpool 7×7	-	-	1	-
$1 \times 1 \times 1280$	Conv2d 1×1	-	k	-	-

Sementara itu, *MobileNetV3* merupakan pengembangan lanjutan yang memanfaatkan kombinasi teknik *Neural Architecture Search* (NAS) serta modul *Squeeze-and-Excitation* (SE) untuk meningkatkan efisiensi dan kemampuan model dalam mengekstraksi fitur [11]. Selain itu, arsitektur ini menggunakan fungsi aktivasi yang lebih optimal untuk meningkatkan akurasi klasifikasi. Tabel 4 menunjukkan arsitektur *MobileNetV3*.

Tabel 4. Arsitektur *MobileNetV3*

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	11
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	2
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Penelitian ini menggunakan arsitektur *MobileNet* dengan pendekatan *transfer learning* sehingga model tidak dilatih dari awal. *Transfer learning* digunakan untuk meningkatkan kemampuan model dengan cara mentransfer pengetahuan dari model yang telah dilatih pada domain serupa. Proses tersebut dilakukan dengan menggunakan model yang telah dilatih terlebih dahulu (*pre-trained model*) menggunakan dataset dalam skala besar, kemudian digunakan kembali untuk mengekstraksi fitur pada dataset baru [19]. Selain itu, dilakukan proses *fine-tuning* untuk menyesuaikan model dengan karakteristik dataset penelitian. Pada tahap ini, sebagian lapisan awal dari model *MobileNet* dipertahankan untuk menjaga fitur dasar yang telah

dipelajari sebelumnya, sedangkan beberapa lapisan terakhir dilatih kembali agar model mampu mempelajari fitur-fitur yang lebih spesifik sesuai dengan dataset citra jerawat. Pendekatan ini membantu mempercepat proses pelatihan sekaligus meningkatkan performa model dalam proses klasifikasi.

Model *MobileNet* pada penelitian ini digunakan sebagai *feature extractor* dengan menghilangkan lapisan klasifikasi bawaan dari model (*include_top = False*). Fitur yang dihasilkan oleh *MobileNet* kemudian diproses menggunakan lapisan *GlobalAveragePooling2D* yang bertujuan untuk mereduksi dimensi fitur yang dihasilkan oleh lapisan sebelumnya [20]. Selanjutnya, ditambahkan lapisan *Batch Normalization* untuk menstabilkan proses pelatihan model. Lapisan *Dense* dengan 256 neuron dan fungsi aktivasi *ReLU* digunakan untuk mempelajari representasi fitur yang lebih kompleks [13]. Untuk mengurangi risiko *overfitting*, digunakan lapisan *Dropout* sebesar 0.5. Pada lapisan output digunakan fungsi aktivasi *Softmax* yang digunakan pada lima neuron yang merepresentasikan lima kelas jerawat, yaitu *blackhead*, *whitehead*, *papules*, *pustules*, dan *cyst*. Konfigurasi hyperparameter yang digunakan dalam proses training model dalam penelitian ini dapat diamati pada Tabel 5.

Pemilihan *MobileNet* sebagai metode utama dalam penelitian ini dilandasi oleh beberapa pertimbangan. Pertama, *MobileNet* adalah arsitektur CNN yang ringan, dirancang untuk mencapai efisiensi komputasi tanpa mengorbankan performa klasifikasi citra. Kedua, pendekatan *transfer learning* dinilai sesuai karena dataset jerawat yang digunakan memiliki ukuran yang terbatas, sehingga pemanfaatan bobot awal dari model pra-latih *ImageNet* dapat mempercepat konvergensi dan meningkatkan kualitas ekstraksi fitur. Ketiga, penggunaan metrik evaluasi seperti *accuracy*, *precision*, *recall*, *F1-score*, *confusion matrix*, dan *classification report* memungkinkan penilaian performa model dilakukan secara menyeluruh, baik secara umum maupun pada tiap kelas, sehingga hasil perbandingan antar arsitektur menjadi lebih komprehensif dan memiliki dasar analitis yang lebih kuat.

Tabel 5. Hyperparameter Pelatihan Model

Hyperparameter	Value
Epoch	50
Batch Size	32
Optimizer	Adam
Learning Rate	0.0001
Input Image Size	224 × 224
Activation Function	ReLU
Output Activation	Softmax
Dropout	0.5

2.4. Evaluasi Model

Evaluasi dilaksanakan guna mengukur performa model yang telah melalui proses pelatihan sebelumnya. Pengujian menggunakan data uji yang telah disiapkan pada tahap awal [21]. Performa model selanjutnya dianalisis dengan berbagai metrik evaluasi, yakni confusion matrix, akurasi, presisi, recall, serta F1-score. Confusion matrix merupakan salah satu teknik evaluasi untuk model klasifikasi yang disajikan dalam bentuk tabel (lihat Tabel 6), guna menggambarkan kaitan antara prediksi model dan nilai sebenarnya dari data.

Tabel 6. Confusion Matrix

Aktual	Prediksi	
	<i>True</i>	<i>False</i>
<i>True</i>	TP	FN
<i>False</i>	FP	TN

True Positive (TP) merujuk pada jumlah data yang sebenarnya positif dan berhasil terdeteksi sebagai positif oleh model. False Positive (FP) merupakan data yang sebenarnya negatif, tetapi diklasifikasikan sebagai positif oleh model. False Negative (FN) menunjukkan data yang sebenarnya positif, tetapi diprediksi sebagai negatif oleh model. Sedangkan True Negative (TN) adalah jumlah data yang memang negatif dan juga terprediksi negatif oleh model.

2.4.1. Akurasi (Accuracy)

Akurasi adalah salah satu metrik untuk mengukur tingkat kebenaran model dalam mengklasifikasikan data. Nilai ini didapat dari rasio jumlah prediksi yang benar terhadap total data uji. Rumus perhitungan akurasi tersedia pada Persamaan (1).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

2.4.2. Presisi (*Precision*)

Presisi mengukur tingkat akurasi model dalam memprediksi kelas positif. Metrik ini menunjukkan rasio prediksi positif yang tepat terhadap keseluruhan prediksi positif yang dihasilkan oleh model. Rumus presisi disajikan pada Persamaan (2).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

2.4.3. Sensitivitas (*Recall*)

Sensitivitas atau *recall* mengukur kemampuan model untuk mendeteksi seluruh data dengan label positif. Metrik ini menyatakan persentase data positif yang berhasil diidentifikasi secara benar oleh model. Rumus sensitivitas disajikan pada Persamaan (3).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

2.4.4. F-Measure (*F1-Score*)

F-Measure atau F1-Score adalah metrik evaluasi yang menggabungkan *precision* dan *recall* untuk menghasilkan nilai performa model secara keseluruhan. Rumus F1-Score ditunjukkan pada Persamaan (4).

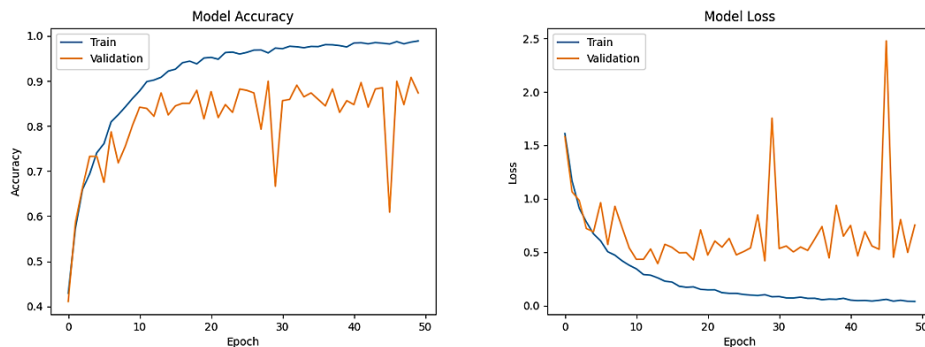
$$\text{F1-Score} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \quad (4)$$

3. HASIL DAN PEMBAHASAN

Pada tahap ini ditunjukkan hasil dari eksperimen yang telah dilakukan dalam studi untuk mengklasifikasikan tipe jerawat dengan menggunakan arsitektur *MobileNet*. Proses eksperimen meliputi tahap pelatihan model, pengujian model, serta evaluasi performa model menggunakan beberapa metrik evaluasi. Hasil dari setiap tahap kemudian dianalisis untuk mengetahui kemampuan model dalam mengidentifikasi dan mengklasifikasikan citra jerawat berdasarkan lima kelas yang digunakan dalam penelitian ini, yaitu *blackhead*, *whitehead*, *papules*, *pustules*, dan *cyst*.

3.1. Pelatihan Model

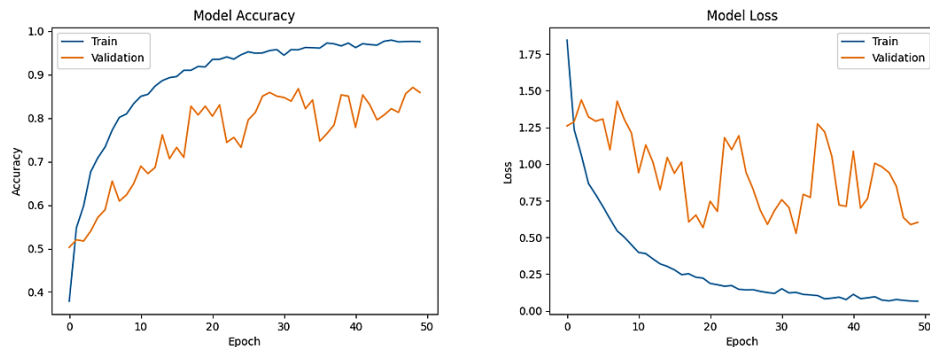
Proses pelatihan model pada penelitian ini dilakukan menggunakan tiga arsitektur *MobileNet*, yaitu *MobileNetV1*, *MobileNetV2*, dan *MobileNetV3*. Model dilatih dengan dataset pelatihan yang telah menjalani tahap *preprocessing*, yang mencakup pencernaan ulang gambar, normalisasi gambar, dan peningkatan data. Semua citra dalam dataset disesuaikan menjadi ukuran 224×224 piksel dalam format tiga kanal warna (RGB) agar cocok dengan dimensi input arsitektur *MobileNet*. Pelatihan model dengan pendekatan *transfer learning* menggunakan bobot awal dari dataset *ImageNet*. Pendekatan ini memungkingkan model memanfaatkan fitur dasar yang sudah dipelajari sebelumnya, sehingga proses pelatihan menjadi lebih cepat dan efisien. Parameter pelatihan yang dipakai meliputi *batch size* sebesar 32, jumlah *epoch* sebanyak 50, serta *optimizer* Adam dengan *learning rate* sebesar 0.0001. Selain itu, pada bagian akhir arsitektur model digunakan fungsi aktivasi *ReLU* pada lapisan *Dense* serta fungsi aktivasi *Softmax* pada lapisan output untuk menghasilkan prediksi klasifikasi pada lima kelas jerawat. Selama proses pelatihan, model mempelajari pola fitur dari citra jerawat yang terdapat pada dataset pelatihan. Untuk mengawasi kinerja model selama proses pelatihan, dimanfaatkan metrik *accuracy* dan *loss* pada data pelatihan dan data validasi.



Gambar 3. Grafik Accuracy dan Loss *MobileNetV1*

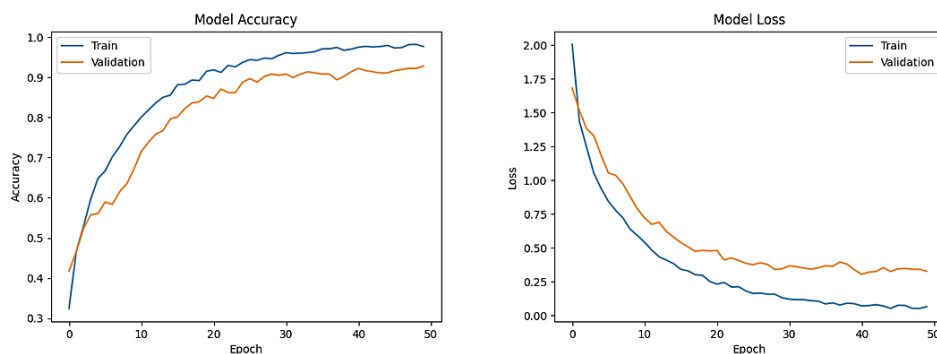
Gambar 3 memperlihatkan bahwa nilai *accuracy* pada data pelatihan mengalami peningkatan secara bertahap dari sekitar 0.42 pada epoch awal hingga mencapai sekitar 0.98–0.99 pada epoch terakhir. Sementara itu, nilai *accuracy* pada data validasi juga menunjukkan peningkatan hingga mencapai kisaran

0.85–0.90, meskipun masih mengalami fluktuasi pada beberapa epoch. Hal ini mengindikasikan bahwa model *MobileNetV1* berhasil menangkap pola fitur dari dataset citra jerawat dengan cukup baik sepanjang proses pelatihan.



Gambar 4. Grafik Accuracy dan Loss *MobileNetV2*

Garfik pada Gambar 4 menggambarkan bahwa nilai *accuracy* pada data pelatihan meningkat secara bertahap dari sekitar 0.38 pada epoch awal hingga mencapai sekitar 0.97 pada epoch terakhir. Sementara itu, nilai *accuracy* pada data validasi juga menunjukkan peningkatan dari sekitar 0.50 hingga berada pada kisaran 0.80–0.86 pada beberapa epoch terakhir, meskipun masih mengalami fluktuasi. Hal ini mengindikasikan bahwa model *MobileNetV2* berhasil menangkap pola fitur dari dataset gambar jerawat sepanjang proses pelatihan, namun performa validasi yang masih berfluktuasi menandakan bahwa model belum sepenuhnya stabil dalam melakukan generalisasi terhadap data validasi.



Gambar 5. Grafik Accuracy dan Loss *MobileNetV3*

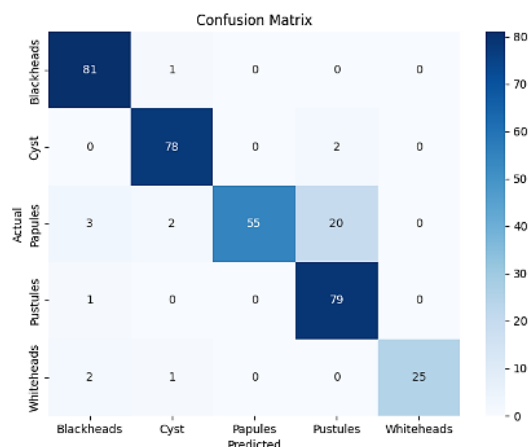
Gambar 5 menunjukkan perkembangan nilai *accuracy* dan *loss* selama proses training model *MobileNetV3*. Berdasarkan grafik yang ditampilkan, terlihat bahwa *accuracy* pada data pelatihan meningkat secara signifikan dari sekitar 0.32 pada epoch awal hingga mencapai sekitar 0.98 pada epoch terakhir. Sementara itu, Performa akurasi pada data validasi juga mengalami peningkatan yang konsisten mulai dari sekitar 0.42 hingga mencapai kisaran 0.91–0.93 pada epoch akhir. Selain itu, grafik *loss* pada data pelatihan menampilkan penurunan yang stabil dari sekitar 2.0 hampir mencapai 0.05, sementara *loss* pada data validasi juga turun hingga berada pada kisaran 0.30–0.35. Hasil ini mengindikasikan bahwa model *MobileNetV3* dapat mempelajari pola citra jerawat secara lebih stabil dan menghasilkan performa pelatihan yang lebih unggul dibandingkan model sebelumnya.

Berdasarkan hasil pelatihan dari ketiga model yang ditampilkan pada Gambar 3, Gambar 4, serta Gambar 5, tampak bahwa seluruh arsitektur *MobileNet* mampu mempelajari pola dari dataset citra jerawat dengan cukup baik. Hal ini ditunjukkan oleh peningkatan nilai *accuracy* serta penurunan nilai *loss* selama proses pelatihan berlangsung. Namun demikian, terdapat perbedaan performa pada masing-masing model. *MobileNetV3* menampilkan peningkatan akurasi yang lebih stabil serta *validation accuracy* lebih tinggi dibandingkan *MobileNetV1* dan *MobileNetV2*. Hal ini mengindikasikan bahwa arsitektur *MobileNetV3* menunjukkan kemampuan superior dalam mempelajari fitur citra jerawat pada dataset yang dipakai dalam penelitian ini.

3.2. Pengujian Model

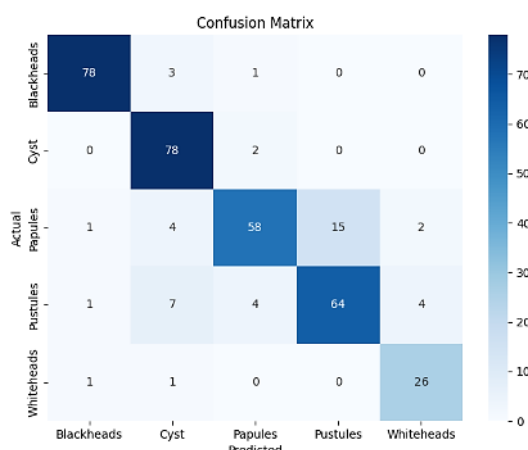
Setelah proses training model telah dilakukan, langkah berikutnya adalah pengujian model dengan memanfaatkan dataset pengujian (*testing dataset*). Dataset pengujian adalah data yang tidak dipakai pada tahap pelatihan atau validasi, sehingga dapat digunakan untuk menilai performa model dalam melakukan

klasifikasi pada data baru. Pada penelitian ini, dataset pengujian terdiri dari 350 citra jerawat yang dikategorikan menjadi lima kelas, yakni *blackhead*, *whitehead*, *papules*, *pustules*, dan *cyst*. Proses pengujian dilakukan dengan memasukkan citra dari dataset pengujian ke dalam model yang telah dilatih sebelumnya. Model kemudian menghasilkan prediksi kelas untuk setiap citra yang diuji. Hasil prediksi yang diperoleh kemudian dibandingkan dengan label asli untuk mengetahui tingkat ketepatan klasifikasi yang dihasilkan oleh model. Selain itu, untuk memperoleh gambaran performa model secara lebih rinci, hasil pengujian juga dianalisis dengan memanfaatkan *confusion matrix*. Analisis ini menunjukkan jumlah prediksi benar dan salah pada setiap kelas jerawat sehingga dapat menggambarkan kemampuan model dalam mengklasifikasikan tiap kelas serta mengidentifikasi kesalahan klasifikasi yang terjadi selama proses pengujian.



Gambar 6. Confusion Matrix *MobileNetV1*

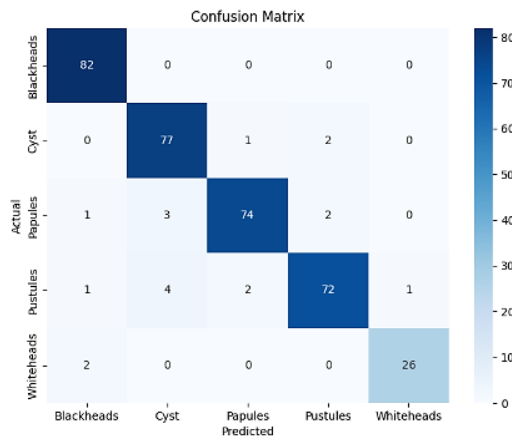
Gambar 6 menunjukkan hasil *confusion matrix* dari proses pengujian model *MobileNetV1*. Berdasarkan matriks tersebut, dapat dilihat bahwa sebagian besar citra berhasil diklasifikasikan secara tepat oleh model. Pada kelas *blackheads*, sebanyak 81 citra berhasil diprediksi dengan tepat dan hanya satu citra yang salah diklasifikasikan sebagai *cyst*. Dalam kelas *cyst*, model sukses mengklasifikasikan 78 citra secara tepat, sementara 2 citra salah diprediksi sebagai *pustules*. Untuk kelas *papules*, model berhasil mengklasifikasikan 55 citra dengan tepat, tetapi ada beberapa kesalahan klasifikasi khususnya pada kelas *pustules* sebanyak 20 citra. Sementara itu, pada kelas *pustules* model mampu mengklasifikasikan 79 citra dengan tepat dengan hanya 1 kesalahan klasifikasi. Pada kelas *whiteheads*, sebanyak 25 citra berhasil diidentifikasi dengan benar, namun masih terdapat beberapa kesalahan klasifikasi pada kelas lain. Secara umum, hasil *confusion matrix* ini mengindikasikan bahwa Model *MobileNetV1* berhasil mengklasifikasikan sebagian besar citra jerawat dengan akurat, walaupun terdapat beberapa kesalahan pada kelas yang memiliki ciri visual yang mirip.



Gambar 7. Confusion Matrix *MobileNetV2*

Gambar 7 mengindikasikan bahwa sebagian besar citra berhasil diklasifikasikan secara tepat oleh model. Pada kelas *blackheads*, sebanyak 78 citra berhasil diprediksi dengan benar, sementara 3 citra salah diklasifikasikan sebagai *cyst* dan 1 citra sebagai *papules*. Pada kelas *cyst*, model mampu mengklasifikasikan 78 citra dengan benar, sedangkan 2 citra lainnya salah diprediksi sebagai *papules*. Untuk kelas *papules*, sebanyak 58 citra berhasil diklasifikasikan dengan tepat, tetapi masih ada beberapa kesalahan klasifikasi

khususnya pada kelas *pustules* sebanyak 15 citra, serta sebagian kecil pada kelas lainnya. Pada kelas *pustules*, model berhasil mengklasifikasikan 64 citra dengan benar, meskipun terdapat beberapa kesalahan prediksi ke kelas *cyst*, *papules*, dan *whiteheads*. Sementara itu, pada kelas *whiteheads*, sebanyak 26 citra berhasil diklasifikasikan dengan tepat, hanya dengan sedikit kesalahan prediksi. Secara keseluruhan, hasil *confusion matrix* ini mengindikasikan bahwa model *MobileNetV2* mampu melakukan klasifikasi dengan performa yang baik, walaupun masih ada beberapa kesalahan prediksi pada kelas-kelas yang memiliki kemiripan ciri visual.



Gambar 8. *Confusion Matrix MobileNetV3*

Gambar 8 menampilkan hasil *confusion matrix MobileNetV3*. Berdasarkan matriks tersebut, tampak bahwa model berhasil mengklasifikasikan sebagian besar citra dengan tepat untuk setiap kelas. Di dalam kelas *blackheads*, seluruh 82 citra berhasil diklasifikasikan dengan tepat tanpa adanya kesalahan prediksi. Pada kelas *cyst*, model berhasil mengklasifikasikan 77 citra dengan benar, sementara itu, 1 citra salah diklasifikasikan sebagai *papules* dan 2 citra sebagai *pustules*. Untuk kelas *papules*, sebanyak 74 citra berhasil diklasifikasikan dengan tepat, meskipun terdapat beberapa kesalahan pada kelas *blackheads*, *cyst*, dan *pustules*. Pada kelas *pustules*, model mampu mengklasifikasikan 72 citra dengan tepat walaupun masih ada beberapa kesalahan prediksi pada kelas lainnya. Sementara itu, di kelas *whiteheads*, sebanyak 26 citra berhasil diklasifikasikan secara tepat dengan hanya 2 kesalahan prediksi pada kelas *blackheads*. Secara keseluruhan, hasil *confusion matrix* ini menunjukkan bahwa model *MobileNetV3* menghasilkan performa klasifikasi yang lebih unggul dan konsisten dibandingkan model sebelumnya.

Berdasarkan hasil evaluasi dengan menggunakan *confusion matrix* pada ketiga arsitektur yang ditunjukkan dalam Gambar 6, Gambar 7, serta Gambar 8, tampak bahwa model *MobileNetV3* menghasilkan prediksi yang lebih akurat dibandingkan dengan *MobileNetV1* dan *MobileNetV2*. Hal ini ditunjukkan oleh jumlah prediksi benar yang lebih tinggi serta jumlah kesalahan klasifikasi yang lebih sedikit pada setiap kelas jerawat. Dengan demikian, arsitektur *MobileNetV3* memiliki kinerja yang lebih unggul dalam melakukan klasifikasi citra jerawat pada kumpulan data yang dipakai dalam studi ini.

3.3. Validasi dan Evaluasi

Setelah proses pengujian model selesai dilakukan, tahap selanjutnya adalah melakukan evaluasi untuk mengukur performa model dalam hal akurasi dan kemampuan klasifikasi. Evaluasi dilakukan dengan menggunakan metrik standar pada klasifikasi citra, yaitu *accuracy*, *precision*, *recall*, dan *F1-score*. Metrik *accuracy* digunakan untuk mengukur tingkat ketepatan model dalam mengklasifikasikan seluruh data uji, sedangkan metrik lainnya digunakan untuk mengevaluasi kinerja model pada masing-masing kelas jerawat.

Hasil evaluasi diperoleh dari pengujian menggunakan dataset uji yang terdiri dari lima kelas jerawat, yaitu *blackhead*, *whitehead*, *papules*, *pustules*, dan *cyst*. Nilai metrik evaluasi dihitung berdasarkan hasil prediksi model terhadap data uji yang disajikan dalam bentuk *classification report*. Selain itu, dilakukan perbandingan performa antara ketiga arsitektur *MobileNet* yang digunakan dalam penelitian ini untuk menentukan model dengan kinerja terbaik.

Tabel 7. Perbandingan Performa Model

Model	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
<i>MobileNetV1</i>	90.86	92	91	91
<i>MobileNetV2</i>	86.86	87	87	87
<i>MobileNetV3</i>	94.57	95	95	95

Berdasarkan hasil evaluasi yang terlihat pada Tabel 7, model *MobileNetV3* memiliki performa paling unggul dibandingkan dengan *MobileNetV1* dan *MobileNetV2*. Hal ini terlihat dari nilai *accuracy* yang lebih tinggi berserta nilai *precision*, *recall*, dan *F1-score* yang lebih unggul dalam proses klasifikasi. Dengan demikian, arsitektur *MobileNetV3* dapat dikatakan lebih efektif dalam mengidentifikasi dan mengklasifikasikan jenis jerawat pada dataset yang dipakai dalam penelitian ini.

Selain menghasilkan nilai evaluasi tertinggi, *MobileNetV3* juga menunjukkan stabilitas yang lebih baik selama tahap pelatihan dan validasi. Hal ini mengindikasikan bahwa penyempurnaan arsitektur pada *MobileNetV3* mampu meningkatkan kemampuan model dalam mengekstraksi fitur jerawat yang lebih diskriminatif tanpa mengurangi efisiensi parameter. Implikasi praktis dari temuan ini adalah bahwa *MobileNetV3* berpotensi menjadi arsitektur yang tepat untuk diterapkan pada sistem pendukung analisis awal kondisi kulit berbasis komputer, terutama pada aplikasi mobile dan web yang memerlukan model ringan dengan performa akurasi yang tinggi.

3.4. Diskusi

Hasil penelitian mengindikasikan bahwa ketiga arsitektur *MobileNet* yang dipakai dalam studi ini berhasil mengklasifikasikan citra jerawat dengan performa yang memadai. Hal tersebut terlihat dari kenaikan nilai *accuracy* selama proses pelatihan, serta hasil evaluasi melalui *confusion matrix* dan *classification report*. Ketiga model tersebut mampu mengenali pola fitur dari citra jerawat sehingga dapat mengklasifikasikan lima jenis jerawat, yaitu *blackhead*, *whitehead*, *papules*, *pustules*, dan *cyst*.

Berdasarkan hasil evaluasi, model *MobileNetV3* menunjukkan performa terbaik dibandingkan dengan *MobileNetV1* dan *MobileNetV2*. Model *MobileNetV3* memperoleh nilai akurasi sebesar 94.57%, sedangkan *MobileNetV1* memperoleh akurasi sebesar 90.86% dan *MobileNetV2* sebesar 86.86%. Performa yang lebih baik pada *MobileNetV3* diduga karena arsitektur ini memiliki struktur jaringan yang lebih optimal melalui penggunaan teknik *Neural Architecture Search* (NAS) dan modul *Squeeze-and-Excitation* (SE) yang dapat meningkatkan performa model dalam menangkap fitur-fitur krusial dari citra.

Dibandingkan dengan penelitian terdahulu, penelitian ini memberikan kontribusi yang lebih terarah melalui perbandingan tiga varian *MobileNet* dalam pengaturan eksperimen yang seragam. Dengan desain tersebut, perbedaan performa yang diperoleh dapat diinterpretasikan secara lebih objektif sebagai pengaruh dari perbedaan arsitektur, bukan akibat variasi dataset atau preprocessing. Hal ini memperkuat validitas kesimpulan bahwa *MobileNetV3* merupakan arsitektur yang paling sesuai di antara model yang diuji untuk klasifikasi jenis jerawat.

Secara analitis, kesalahan klasifikasi yang masih ditemukan antara kelas *papules* dan *pustules* merupakan hal yang dapat dipahami, mengingat kedua kelas tersebut memiliki kemiripan karakteristik visual, terutama pada aspek kemerahan, ukuran lesi, dan tekstur lokal. Selain itu, ketidakseimbangan distribusi data antarkelas juga berpotensi memengaruhi performa model, terutama karena jumlah sampel *whitehead* lebih sedikit dibandingkan kelas lainnya. Oleh karena itu, pembahasan hasil penelitian tidak hanya perlu menonjolkan capaian akurasi tertinggi pada *MobileNetV3*, tetapi juga harus mengakui bahwa kemiripan visual antarkelas dan ketimpangan distribusi data masih menjadi keterbatasan utama dalam penelitian ini.

Selain itu, tahap *pre-processing* yang meliputi *image resizing*, normalisasi citra, dan augmentasi data berperan penting dalam meningkatkan kinerja model. Augmentasi data berperan dalam menambah keragaman pada dataset pelatihan sehingga model mampu mempelajari pola fitur secara lebih optimal serta meningkatkan kemampuan generalisasi terhadap data yang belum pernah dilihat. Namun demikian, kesalahan klasifikasi masih ditemukan pada kelas-kelas yang memiliki karakteristik visual yang mirip, yang mengindikasikan bahwa kesamaan tekstur dan bentuk jerawat masih menjadi tantangan dalam proses klasifikasi.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa penerapan metode *Deep Learning* berbasis arsitektur *MobileNet* dapat dimanfaatkan secara efektif untuk mengklasifikasikan jenis jerawat berdasarkan citra. Model dikembangkan menggunakan pendekatan *transfer learning* dengan tiga varian arsitektur, yaitu *MobileNetV1*, *MobileNetV2*, dan *MobileNetV3*, yang dilatih menggunakan dataset citra jerawat yang telah melalui tahap *pre-processing* berupa *image resizing*, normalisasi citra, serta augmentasi data.

Hasil evaluasi menunjukkan bahwa model *MobileNetV3* menghasilkan performa terbaik dengan tingkat akurasi sebesar 94.57%, diikuti oleh *MobileNetV1* sebesar 90.86% dan *MobileNetV2* sebesar 86.86%. Hal ini mengindikasikan bahwa arsitektur *MobileNetV3* memiliki kemampuan yang lebih optimal dalam mengekstraksi fitur citra jerawat dibandingkan dengan model lainnya.

Berdasarkan hasil tersebut, penelitian ini memberikan implikasi bahwa model *transfer learning* dengan arsitektur ringan memiliki potensi yang baik untuk mendukung pengembangan sistem klasifikasi citra jerawat yang praktis, akurat, dan efisien secara komputasional. Untuk penelitian selanjutnya, disarankan penggunaan dataset yang lebih besar dan seimbang, penerapan teknik *cross-validation*, eksplorasi metode

explainability seperti *Grad-CAM* [22], serta evaluasi performa model pada implementasi perangkat mobile maupun sistem pendukung keputusan klinis berbasis web.

REFERENSI

- [1] V. Marshalia, “Perancangan Pemanfaatan IoT dan Machine Learning Untuk Sistem Monitoring Kesehatan Kulit Wajah Berjerawat,” vol. 3, no. 1, pp. 135–145, 2024.
- [2] N. U. R. Sifatullah, “Jerawat (*Acne vulgaris*): Review Penyakit Infeksi Pada Kulit,” no. November, pp. 19–23, 2021.
- [3] M. I. Anshori, R. A. Yaqin, M. Ali, Z. Sidiq, and W. P. Agung, “Klasifikasi Jenis Jerawat Secara Otomatis Dengan Convolutional Neural Network Menggunakan Arsitektur Resnet-50 Automatic Acne Type Classification with Convolutional Neural Network Using Resnet-50 Architecture,” vol. 15, no. April, pp. 73–84, 2025.
- [4] V. Dalira, B. Sebayang, I. G. Ngurah, and L. Wijaya, “Klasifikasi Jenis Jerawat Berdasarkan Citra Menggunakan Convolutional Neural Network dengan Arsitektur *MobileNetV2*,” vol. 14, no. 3, pp. 766–774, 2024.
- [5] S. I. Ali and A. M. Abdulazeez, “Machine Learning Classification Algorithms-Based Smart Cities Applications: A Review,” *Indones. J. Comput. Sci.*, vol. 12, no. 2, pp. 284–301, 2023, [Online]. Available: <http://ijcs.stmikindonesia.ac.id/ijcs/index.php/ijcs/article/view/3135>
- [6] F. Ramadhani, S. Rahardianto, and M. Masjkur, “Acne Severity Classification Study Using Convolutional Neural Network Algorithm with *MobileNetV2* Architecture: Kajian Klasifikasi Tingkat Keparahan Jerawat Menggunakan Algoritma Convolutional Neural Network,” *Indones. J. Stat. Its Appl.*, vol. 8, no. 2, pp. 112–128, 2024, [Online]. Available: <https://app.dimensions.ai/details/publication/pub.1184056910%0Ahttps://doi.org/10.29244/ijsa.v8i2p112-128>
- [7] F. N. Cahya, N. Hardi, D. Riana, and S. Hadiani, “Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN),” vol. 10, pp. 618–626, 2021.
- [8] H. A. Faudyta, J. T. Sinaga, and E. R. Subhiyakt, “Implementation of *MobileNet* Architecture for Skin Cancer Disease Classification,” *J. Appl. Informatics Comput.*, vol. 8, no. 2, pp. 589–597, 2024, doi: 10.30871/jaic.v8i2.8771.
- [9] A. R. Andini and I. Yuadi, “Classification of Acne Type Based on Convolutional Neural Network Klasifikasi Jenis Jerawat Berdasarkan Convolutional Neural Network,” vol. 5, no. January, pp. 301–308, 2025.
- [10] A. G. Howard and W. Wang, “*MobileNets*: Efficient Convolutional Neural Networks for Mobile Vision Applications,” 2017.
- [11] A. Howard *et al.*, “Searching for *MobileNetV3*,” *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1314–1324, 2019, doi: 10.1109/ICCV.2019.00140.
- [12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “*MobileNetV2*: Inverted Residuals and Linear Bottlenecks,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.
- [13] N. Rochmawati, H. B. Hidayati, Y. Yamasari, H. P. A. Tjahyaningtijas, W. Yustanti, and A. Prihanto, “Analisa Learning Rate dan Batch Size pada Klasifikasi Covid Menggunakan Deep Learning dengan Optimizer Adam,” *J. Inf. Eng. Educ. Technol.*, vol. 5, no. 2, pp. 44–48, 2021, doi: 10.26740/jieet.v5n2.p44-48.
- [14] J. Zi, W. Hu, G. Fan, F. Chen, and Y. Chen, “SFL-*MobileNetV3*: A lightweight network for weed recognition in tropical cassava fields in China,” *Expert Syst. Appl.*, vol. 277, no. December 2024, p. 127196, 2025, doi: 10.1016/j.eswa.2025.127196.
- [15] M. A. Hakiki and N. Rachmat, “Klasifikasi Spesies Jamur Menggunakan Convolutional Neural Network dengan Arsitektur *MobileNetV2*,” vol. 6, no. 1, pp. 11–24, 2025, doi: 10.35957/algoritme.v6i1.11077.
- [16] S. Ramdani and A. Rahmatulloh, “Implementasi *MobileNet* untuk Klasifikasi Gambar dan Deteksi Emosi Menggunakan KERAS *MobileNet* Implementation for Image Classification and Emotion Detection Using KERAS,” vol. 12, no. 2, pp. 259–264, 2024, doi: 10.26418/justin.v12i2.73389.
- [17] D. Aura, A. Sujjada, and I. Lucia, “Implementasi *MobileNet* dalam Klasifikasi Jenis Kulit Wajah untuk Penentuan Kandungan Skincare,” vol. 7, no. 1, pp. 17–24, 2025.
- [18] A. Kahfi, A. Shiddiq, and T. Dzikrullah, “Deteksi Penyakit pada Tanaman Padi Menggunakan *MobileNet* Transfer Learning Berbasis Android,” vol. 2, no. 2, 2022.
- [19] N. S. Suriani, S. S. A. Tarmizi, M. N. H. Mohd, and S. M. Shah, “Acne Severity Classification on Mobile Devices using Lighweight Deep Learning Approach,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 6, pp. 680–687, 2024, doi: 10.14569/IJACSA.2024.0150668.
- [20] W. Bismi, D. Novianti, and M. Qomaruddin, “Analisis Perbandingan Klasifikasi Citra Genus *Panthera* dengan Pendekatan Deep learning Model *MobileNet*,” *J. Inform. dan Rekayasa Perangkat Lunak*, vol.

- 6, no. 1, pp. 1–9, 2024.
- [21] J. Arifianto, “Aplikasi Web Pendeteksi Jerawat Pada Wajah Menggunakan Algoritma Deep Learning dengan TensorFlow,” 2024.
- [22] S. Suara, A. Jha, P. Sinha, and A. A. Sekh, “Is Grad-CAM Explainable in Medical Images?,” *Commun. Comput. Inf. Sci.*, vol. 2009 CCIS, pp. 124–135, 2024, doi: 10.1007/978-3-031-58181-6_11.