

TESIS
ANALISIS PERFORMA VGG16 DAN MOBILENETV2
DALAM KLASIFIKASI PENYAKIT DAUN PISANG
MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK
(CNN)



disusun oleh:

Nama : Nita Helmawati
NIM : 24.51.1630
Konsentrasi : Business Intelligence

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2025

TESIS
ANALISIS PERFORMA VGG16 DAN MOBILENETV2
DALAM KLASIFIKASI PENYAKIT DAUN PISANG
MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK
(CNN)

PERFORMANCE ANALYSIS OF VGG16 AND
MOBILENETV2 IN BANANA LEAF DISEASE
CLASSIFICATION USING CONVOLUTIONAL NEURAL
NETWORK (CNN)

Diajukan untuk memenuhi salah satu syarat mencapai derajat Pascasarjana
Program Studi S2 Informatika



disusun oleh:

Nama : Nita Helmawati
NIM : 24.51.1630
Konsentrasi : Business Intelligence

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2025

HALAMAN PERSETUJUAN
ANALISIS PERFORMA VGG16 DAN MOBILENETV2 DALAM
KLASIFIKASI PENYAKIT DAUN PISANG MENGGUNAKAN
CONVOLUTIONAL NEURAL NETWORK (CNN)

PERFORMANCE ANALYSIS OF VGG16 AND MOBILENETV2 IN
BANANA LEAF DISEASE CLASSIFICATION USING
CONVOLUTIONAL NEURAL NETWORK (CNN)

yang disusun dan diajukan oleh

Nita Helmawati

24.51.1630

telah disetujui oleh Dosen Pembimbing Tesis
pada tanggal 3 November 2025

Dosen Pembimbing,



Prof. Dr. Ema Utami, S.Si., M.Kom
NIK. 190302037

HALAMAN PENGESAHAN

**ANALISIS PERFORMA VGG16 DAN MOBILENETV2 DALAM
KLASIFIKASI PENYAKIT DAUN PISANG MENGGUNAKAN
CONVOLUTIONAL NEURAL NETWORK (CNN)**

**PERFORMANCE ANALYSIS OF VGG16 AND MOBILENETV2 IN
BANANA LEAF DISEASE CLASSIFICATION USING
CONVOLUTIONAL NEURAL NETWORK (CNN)**

yang disusun dan diajukan oleh

Nita Helmawati

24.51.1630

Telah dipertahankan di depan Dewan Penguji
pada tanggal 3 November 2025

Susunan Dewan Penguji

Nama Penguji

Dr. Andi Sunvoto, M.Kom

NIK. 190302052

Hanif Al Fatta, S.Kom., M.Kom., Ph.D

NIK. 190302096

Prof. Dr. Ema Utami, S.Si., M.Kom

NIK. 190302037

Tanda Tangan



Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer
Tanggal 3 November 2025

DEKAN FAKULTAS ILMU KOMPUTER



Prof. Dr. Kusriani, M.Kom,

NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Nita Helmawati
NIM : 24.51.1630
Konsentrasi : *Business Intelligence*

Menyatakan bahwa Tesis dengan judul berikut:
Analisis Performa VGG16 dan MobileNetV2 dalam Klasifikasi Penyakit Daun Pisang Menggunakan Convolutional Neural Network (CNN)

Dosen Pembimbing : Prof. Dr. Ema Utami, S.Si., M.Kom

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 3 November 2025

Yang Menyatakan,



Nita Helmawati

HALAMAN PERSEMBAHAN

Alhamdulillahirabbil 'aalamiin. Segala puji bagi Allah SWT atas limpahan nikmat, karunia, dan pertolongan-Nya yang tak pernah putus hingga detik ini. Karya ini lahir sebagai bentuk rasa syukur yang dalam atas semua kemudahan dan kekuatan yang telah Allah berikan kepada penulis.

Dengan penuh cinta, karya ini penulis persembahkan untuk kedua orang tua tercinta dan seluruh keluarga yang telah mewarnai hidup penulis dengan kasih sayang dan kebahagiaan. Terima kasih atas cinta, doa, nasihat, dan dukungan yang tiada henti, baik secara materi maupun moral. Tanpa itu semua, penulis tidak akan sampai di titik ini.

Karya ini juga penulis persembahkan untuk teman-teman dan sahabat-sahabat terbaik. Terima kasih telah menjadi bagian dari perjalanan hidup penulis, dalam suka maupun duka. Kehadiran kalian sungguh berarti, dan penulis bersyukur pernah dipertemukan dengan orang-orang sebaik kalian.

Untuk Prof Ema, terima kasih atas segala bimbingan, masukan, dan ilmu yang telah diberikan selama ini. Alhamdulillah, anak wedok kini telah menyelesaikan jenjang Magister. Tentu perjalanan masih panjang dan banyak tantangan yang menanti. Semoga semua bekal ilmu ini bisa menjadi pijakan untuk melangkah ke masa depan. Doa terbaik untuk Prof sekeluarga, semoga senantiasa dalam lindungan dan keberkahan Allah SWT.

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas segala limpahan rahmat, karunia dan kasih sayang-Nya, sehingga penulis diberikan kekuatan, kesehatan dan kesempatan untuk menyelesaikan penyusunan tesis ini dengan baik.

Proses penyusunan tesis ini tentu tidak selalu mudah. Ada banyak cerita, perjuangan, tantangan, serta pembelajaran yang mewarnai setiap prosesnya. Namun, penulis meyakini bahwa tesis yang baik adalah tesis yang selesai. Maka dari itu, penyelesaian tesis ini tidak lepas dari peran, bantuan, doa serta dukungan dari banyak pihak yang dengan tulus memberikan semangat, motivasi dan arahan.

Dengan segala kerendahan hati, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Cinta Pertama penulis, Bapak Abd. Lazim. Ucapan terima kasih yang sebesar-besarnya penulis sampaikan sebagai teladan utama dalam keteguhan, kesabaran, dan ketulusan doa yang senantiasa mengiringi setiap langkah kehidupan penulis. Terima kasih atas segala bentuk pengorbanan, kasih sayang, serta tanggung jawab yang Bapak berikan dalam memastikan kehidupan yang layak bagi penulis hingga saat ini. Terima kasih juga karena telah menjadi sumber inspirasi dan motivasi yang mendorong penulis untuk berjuang meraih gelar hingga level Magister, sebagaimana yang Bapak harapkan. Semoga Bapak senantiasa dikaruniai umur panjang, kesehatan,

dan kebahagiaan sehingga dapat terus menyaksikan serta mendampingi berbagai fase kehidupan yang akan datang.

2. Kepada Pintu Surga penulis, Ibu Aminah. Dengan penuh hormat penulis menyampaikan penghargaan dan terima kasih yang mendalam kepada beliau yang setiap doa dan pelukannya menjadi sumber kekuatan terbesar dalam setiap langkah perjalanan hidup penulis. Tiada rangkaian kata yang sepenuhnya mampu menggambarkan rasa syukur ini. Terima kasih atas anugerah kehidupan melalui kelahiran, perawatan, dan kasih sayang yang tulus, atas doa yang senantiasa menyertai, serta dukungan yang tiada henti dalam setiap fase kehidupan penulis. Terima kasih juga atas pengorbanan waktu, tenaga, dan upaya yang Ibu curahkan hingga penulis dapat mencapai posisi saat ini. Menjadi suatu kebanggaan tersendiri memiliki sosok ibu yang luar biasa, yang senantiasa mendukung pencapaian cita-cita. Semoga Allah SWT senantiasa melimpahkan kesehatan, kebahagiaan, dan umur panjang kepada Ibu.
3. Kepada Kakak-Kakak Tercinta, Ayu Astuti dan Usman Jayadi. Penulis menyampaikan terima kasih atas segala bentuk dukungan, motivasi, dan semangat yang senantiasa diberikan selama proses perjalanan ini. Kehadiran serta dorongan yang diberikan telah menjadi salah satu faktor penting yang menguatkan penulis dalam menyelesaikan setiap tahap pencapaian hingga berada pada titik ini.
4. Abdul Azis, adik bungsu penulis. Tumbuhlah menjadi generasi yang cerdas, berakhlak mulia, berjiwa mandiri, serta mampu memberikan kontribusi

positif bagi keluarga, masyarakat, dan bangsa. Semoga setiap langkahmu senantiasa disertai keberkahan, kebahagiaan, dan kesuksesan.

5. Yudika Ilpan, Revan Ramdani, Karisa Muliani Putri, dan Wais Alkaroni, keponakan-keponakan tercinta, yang selalu membawa warna dan kebahagiaan tersendiri dalam hari-hari penulis. Tumbuhlah menjadi pribadi yang tangguh, berilmu, berhati baik dan mampu meraih impian serta memberikan manfaat bagi sesama. Semoga perjalanan hidup kalian dipenuhi kesempatan yang baik, langkah yang dimudahkan, dan kebahagiaan yang tak terputus.
6. Seluruh keluarga besar yang tidak dapat disebutkan satu per satu, yang selalu menjadi sumber doa dan dukungan moril bagi penulis.
7. Prof. Dr. Ema Utami, S.Si., M.Kom selaku dosen pembimbing, terima kasih atas segala bimbingan, kesabaran, dan arahan yang sangat berarti bagi kelancaran penyusunan tesis ini. Terima kasih juga karena senantiasa meluangkan waktu, bahkan di luar jam kerja untuk tetap responsif dalam memberikan masukan dan dukungan, sehingga penulis dapat menyelesaikan setiap tahap penelitian dengan baik. Keteladanan, dedikasi, dan perhatian yang ditunjukkan menjadi sumber motivasi yang tak ternilai bagi penulis. Semoga segala kebaikan dan keikhlasannya senantiasa mendapatkan balasan terbaik dari Allah SWT.
8. Bp. Dr. Andi Sunyoto, M.Kom sebagai penguji pertama dan Bp. Hanif Al Fatta, M.Kom., Ph.D sebagai penguji kedua, terima kasih telah meluangkan waktu serta menyampaikan berbagai masukan, saran, dan kritik yang

membangun selama proses ujian tesis. Setiap arahan dan evaluasi yang diberikan menjadi pembelajaran berharga bagi penulis dalam memperbaiki, menyempurnakan, dan memperdalam pemahaman terhadap penelitian ini.

9. Keluarga besar Direktorat BPC, terima kasih atas dukungan, semangat, dan kebersamaan yang begitu berharga selama ini. Kehangatan dan rasa saling mendukung yang terjalin telah menjadi salah satu kekuatan yang membantu penulis melewati berbagai tantangan hingga dapat menyelesaikan penelitian ini. Terima kasih atas dua tahun kebersamaan yang penuh kesan, yang akan selalu menjadi bagian berharga dalam perjalanan hidup penulis.
10. Teman-teman sekelas Angkatan 2024 Informatika Reguler, terima kasih telah menemani perjalanan panjang ini dengan canda tawa, semangat, dan kebersamaan yang tak ternilai.
11. Terakhir, terima kasih untuk perempuan mungil yang memiliki impian besar namun terkadang sulit dimengerti isi kepalanya, yaitu diri sendiri. Seorang anak perempuan berumur 23 tahun, sangat keras kepala tapi penuh ambisi, terima kasih telah berusaha keras untuk meyakini dan menguatkan diri sendiri bahwa kamu dapat menyelesaikan studi ini sampai selesai. Berbahagialah selalu dan rayakan bahagiamu sebagai berkah dimanapun kamu menginjakkan kaki. Jangan sia-siakan doa yang selalu kamu langitkan, Allah sudah merencanakan dan memberikan porsi terbaik untuk perjalanan hidupmu. Semoga langkah kebaikan selalu menyertaimu dan semoga Allah selalui meridhai serta menjagamu dalam lindungan-Nya. Terima kasih telah membuktikan pada orang lain bahwa studi ini akan

selesai sekaligus mengukir pencapaian sebagai Magister pertama di keluarga besar.

Penulis menyadari bahwa tesis ini masih jauh dari kata sempurna, mengingat keterbatasan pengetahuan dan pengalaman yang dimiliki. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun demi kesempurnaan karya ini di masa yang akan datang.

Yogyakarta, 3 November 2025

Penulis



DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN KEASLIAN TESIS.....	iv
HALAMAN PERSEMBAHAN.....	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN.....	xvi
DAFTAR LAMBANG DAN SINGKATAN.....	xvii
DAFTAR ISTILAH.....	xviii
INTISARI.....	xix
<i>ABSTRACT</i>	xx
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	6
1.3. Batasan Masalah.....	7
1.4. Tujuan Penelitian.....	7
1.5. Manfaat Penelitian.....	8
BAB II TINJAUAN PUSTAKA.....	10
2.1. Tinjauan Pustaka.....	10

2.2. Keaslian Penelitian	16
2.3. Landasan Teori	24
2.3.1 <i>Convolutional Neural Network (CNN)</i>	24
2.3.2 VGG16	25
2.3.3 MobileNetV2	27
2.3.4 Klasifikasi Penyakit Daun Pisang dengan CNN	29
2.3.5 Klasifikasi Penyakit Daun Pisang dengan CNN	30
2.3.6 Evaluasi Performa Model	31
2.3.7 Parameter Setting	32
BAB III METODE PENELITIAN	36
3.1. Jenis, Sifat, dan Pendekatan Penelitian	36
3.2. Metode Pengumpulan Data	37
3.3. Metode Analisis Data	38
3.4. Alur Penelitian	38
3.4.1 Pengambilan Data	39
3.4.2 Augmentasi Data	40
3.4.3 Split Data	44
3.4.4 Model VGG16	45
3.4.5 Model MobileNetV2	47
3.4.6 Evaluasi Model	48
BAB IV HASIL PENELITIAN DAN PEMBAHASAN	51
4.1. Pelatihan Menggunakan VGG16	51
4.1.1 Hasil akurasi	51

4.1.2 Hasil Presisi	52
4.1.3 Hasil Recall	53
4.1.4 Hasil F1-Score	54
4.1.5 Hasil <i>Training Time</i>	55
4.1.6 Hasil <i>Confusion Matriks</i> dan <i>Loss</i>	56
4.2. Pelatihan Menggunakan MobileNetV2	59
4.2.6 Hasil <i>Confusion Matriks</i> dan <i>Loss</i>	64
4.3. Evaluasi Model	68
4.3.1 Hasil Evaluasi VGG16	68
4.3.2 Skenario Terbaik VGG16	69
4.3.3 Faktor yang Mempengaruhi Skenario Terbaik pada VGG16	71
4.3.4 Hasil Evaluasi MobileNetV2	75
4.3.5 Skenario Terbaik MobileNetV2	77
4.3.6 Faktor yang Mempengaruhi Skenario Terbaik pada MobileNetV2	78
4.3.7 Perbandingan <i>Loop</i>	83
4.3.8 Perbandingan Penelitian	84
BAB V PENUTUP	88
5.1. Kesimpulan	88
5.2. Saran	89
DAFTAR PUSTAKA	91
LAMPIRAN	97

DAFTAR TABEL

Tabel 2.1 Matriks Literatur Review dan Posisi Penelitian	19
Tabel 3.1 Sebelum dan Sesudah <i>Gaussian Blur</i>	41
Tabel 3.2 Sebelum dan Sesudah <i>Cropping</i>	42
Tabel 3.3 Sebelum dan Sesudah <i>Flipping</i>	43
Tabel 3.4 Sebelum dan Sesudah Rotasi	44
Tabel 3.5 Pembagian Dataset	45
Tabel 3.6 Pelatihan VGG16	46
Tabel 3.7 Pelatihan MobileNetV2	47
Tabel 4.1 Skenario Terbaik Pada VGG16	69
Tabel 4.2 Hasil Pengujian Optimizer yang Berbeda dan LR yang Sama	71
Tabel 4.3 Hasil Pengujian Optimizer yang Sama dan Learning Rate Berbeda	73
Tabel 4.4 Skenario Terbaik Pada MobileNetV2	77
Tabel 4.5 Hasil Pengujian Optimizer Yang Beda dan Learning Rate Yang Sama	78
Tabel 4.6 Hasil Pengujian Optimizer Sama dan Learning Rate Berbeda	80
Tabel 4.7 Perbandingan Loop dan Tanpa Loop	83
Tabel 4.8 Perbandingan Penelitian	84

DAFTAR GAMBAR

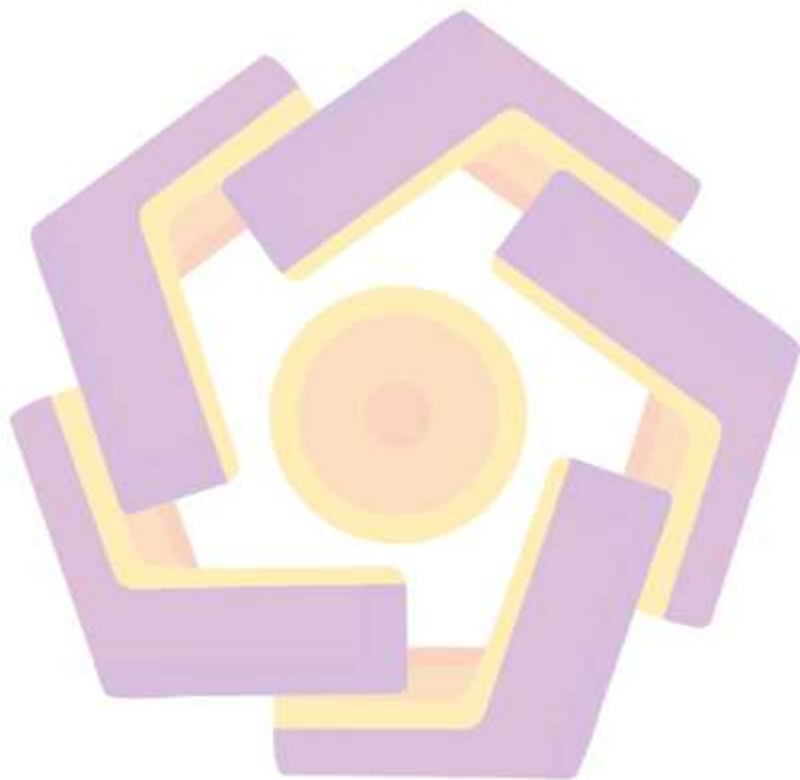
Gambar 2.1 Perbandingan Penelitian Sebelumnya dan Penelitian Saat Ini	16
Gambar 2.2 Arsitektur VGG16	26
Gambar 2.3 Arsitektur MobileNetV2	28
Gambar 3.1 Alur Penelitian	38
Gambar 3.2 Kelas pada Dataset	40



DAFTAR LAMPIRAN

Lampiran 1 Source Code

97



DAFTAR LAMBANG DAN SINGKATAN

TP	<i>True Positive</i>
TN	<i>True Negative</i>
FP	<i>False Positive</i>
FN	<i>False Negative</i>
CNN	<i>Convolutional Neural Network</i>
VGG16	<i>Visual Geometry Group</i>
Adam	<i>Adaptive Moment Estimation</i>
AdamW	<i>Adaptive with Weight Decay</i>
SGD	<i>Stochastic Gradient Descent</i>



DAFTAR ISTILAH

Deteksi Dini	Identifikasi penyakit awal.
Klasifikasi	Pengelompokan citra daun pisang berdasarkan kategori.
VGG16	CNN dengan fitur detail.
MobileNetV2	CNN efisien berukuran kecil.
Citra (<i>Image</i>)	Data visual yang digunakan dalam model <i>deep learning</i> .
<i>Deep Learning</i>	Pembelajaran jaringan mendalam.
<i>Epoch</i>	Satu siklus pelatihan.
<i>Optimizer</i> (Adam)	Pengatur pembaruan bobot.
<i>Learning Rate</i>	Kecepatan belajar model.
Augmentasi Data	Penambahan variasi data.
<i>Transfer Learning</i>	Gunakan model pre-trained.
<i>Overfitting</i>	Model terlalu menyesuaikan data.
Akurasi	Tingkat prediksi benar.
Presisi	Ketepatan prediksi positif.
Recall	Kemampuan menemukan positif.
F1-Score	Gabungan presisi dan recall.

INTISARI

Indonesia merupakan salah satu produsen utama pisang di dunia dengan potensi ekspor yang besar, namun produksinya kerap terancam oleh serangan penyakit daun yang menurunkan kualitas dan kuantitas hasil panen. Deteksi dini dan akurat terhadap penyakit daun pisang menjadi tantangan penting, khususnya dalam kondisi lapangan yang bervariasi. Penelitian ini bertujuan untuk mengevaluasi dan membandingkan performa dua arsitektur *Convolutional Neural Network* (CNN), yaitu VGG16 dan MobileNetV2, dalam mengklasifikasikan penyakit daun pisang secara multi-kelas berdasarkan citra. Metode yang digunakan meliputi pelatihan dan pengujian model dengan dataset citra penyakit daun pisang yang telah dikurasi dan ditingkatkan kualitasnya melalui augmentasi. Evaluasi dilakukan berdasarkan metrik akurasi, presisi, recall, F1-score, dan efisiensi komputasi.

Hasil pengujian menunjukkan bahwa model VGG16 dengan konfigurasi optimal (15 epoch, optimizer Adam, learning rate 0,01) memberikan performa terbaik dengan akurasi sebesar 98,12%, presisi 98,17%, recall 98,12%, dan F1-score 98,12%, serta waktu pelatihan selama 79,92 detik. Sementara itu, MobileNetV2 menunjukkan performa yang cukup baik namun sedikit lebih rendah, dengan keunggulan pada efisiensi komputasi. Pemilihan model optimal sangat bergantung pada kebutuhan sistem, VGG16 direkomendasikan untuk skenario yang menekankan akurasi tinggi, sedangkan MobileNetV2 lebih sesuai untuk implementasi pada perangkat dengan sumber daya terbatas. Penelitian ini memberikan kontribusi terhadap pengembangan untuk klasifikasi penyakit tanaman berbasis deep learning yang praktis dan adaptif terhadap kondisi lapangan.

Kata kunci: klasifikasi penyakit daun pisang, VGG16, MobileNetV2, CNN, deteksi citra daun.

ABSTRACT

Indonesia is one of the world's leading banana producers with significant export potential, but its production is often threatened by leaf diseases that reduce the quality and quantity of the harvest. Early and accurate detection of banana leaf diseases is a critical challenge, especially under varying field conditions. This study aims to evaluate and compare the performance of two Convolutional Neural Network (CNN) architectures, VGG16 and MobileNetV2, in classifying multi-class banana leaf diseases based on images. The methods used include training and testing the model with a curated and enhanced dataset of banana leaf disease images through augmentation. Evaluation is based on accuracy, precision, recall, F1-score, and computational efficiency metrics.

Testing results show that the VGG16 model with optimal configuration (15 epochs, Adam optimizer, learning rate 0.01) provides the best performance with an accuracy of 98.12%, precision of 98.17%, recall of 98.12%, and F1-score of 98.12%, as well as a training time of 79.92 seconds. Meanwhile, MobileNetV2 showed fairly good performance but was slightly lower, with an advantage in computational efficiency. The selection of the optimal model depends heavily on system requirements; VGG16 is recommended for scenarios emphasizing high accuracy, while MobileNetV2 is more suitable for implementation on devices with limited resources. This research contributes to the development of practical and field-adaptive deep learning-based plant disease classification.

Keywords: *banana leaf disease classification, VGG16, MobileNetV2, CNN, leaf image detection.*

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Indonesia adalah salah satu produsen utama pisang di dunia dengan produksi mencapai 8.741.146,7 ton per tahun dari area penanaman seluas 145.401 hektar, yang menghasilkan rata-rata 60.117,4 kg per hektar (Mukhtar Amin, 2023). Tingginya produksi pisang per kapita di Indonesia, mencapai 32,984 kg per orang, tidak hanya memenuhi kebutuhan pangan domestik tetapi juga membuka peluang ekspor yang menjanjikan (Mukhtar Amin, 2023). Pertanian merupakan sektor yang sangat penting dalam kehidupan sosial-ekonomi masyarakat, terutama di negara agraris seperti Indonesia. Pisang adalah salah satu tanaman buah yang memiliki nilai ekonomi tinggi dan permintaan yang terus meningkat. Namun, produksi pisang seringkali terancam oleh berbagai penyakit yang dapat menurunkan kualitas dan kuantitas hasil panen (Md Ripon Sheikh, 2024). Salah satu tantangan utama yang dihadapi para petani adalah identifikasi dan pengendalian penyakit pada daun pisang, yang jika tidak ditangani dengan baik, dapat menyebabkan kerugian yang signifikan.

Dalam penerapan teknologi berbasis kecerdasan buatan, model pembelajaran mesin seperti VGG16 dan MobileNetV2 telah menunjukkan kemajuan signifikan dalam berbagai bidang (Hindarto, 2023; Liao et al., 2023). Untuk deteksi suara pernapasan abnormal menggunakan stetoskop elektronik dan ponsel pintar, penelitian oleh (Liao et al., 2023) mengoptimalkan model

MobileNetV2 untuk mengatasi tantangan komputasi kompleks yang seringkali menyebabkan waktu pemrosesan yang lama dan konsumsi daya yang tinggi. Hal ini sejalan dengan penelitian yang menunjukkan keterbatasan pada basis data yang ada, yang seringkali hanya mencakup sampel kecil dari populasi tertentu, seperti populasi Kaukasia. Selain itu, dalam bidang pengelolaan sumber daya perikanan yang berkelanjutan, penelitian oleh (Hindarto, 2023) mengidentifikasi penggunaan model VGG16 dan MobileNet untuk klasifikasi ikan, yang semakin penting mengingat ancaman penangkapan ikan berlebihan dan perubahan iklim. Di sisi lain, penelitian yang dilakukan oleh (S. Kumar & Kumar, 2024) mengusulkan Efficient-VGG16, sebuah metode ensemble inovatif yang menggabungkan arsitektur VGG16 dan EfficientNet-B0 untuk klasifikasi gambar sinar-X dada COVID-19. Model ini menunjukkan kinerja yang cukup baik, dengan akurasi dan F1-Score yang lebih tinggi dibandingkan dengan metode *transfer learning* lainnya, seperti EfficientNet-B0 dan XgBoost, dengan mencapai akurasi 99,46% dan skor F1 98,41%, yang menggarisbawahi keunggulannya dalam menghadapi tantangan diagnosis cepat dan akurat.

Berdasarkan tinjauan literatur terhadap penelitian-penelitian sebelumnya, teridentifikasi beberapa masalah kritis yang melatarbelakangi pemilihan arsitektur VGG16 dan MobileNetV2 dalam penelitian ini. Masalah kompleksitas komputasi yang tinggi pada model sebelumnya seperti pada penelitian oleh (Chaudhari & Patil, 2023) dengan pendekatan ensemble dan (Seetharaman & Mahendran, 2022) dengan CRNN-RCNN, yang meskipun mencapai akurasi di atas 92%, memerlukan

waktu komputasi yang lama dan sumber daya yang besar (Prasetyo & Utami, 2024; Shetty & Mahesh, 2024).

Penelitian selanjutnya terdapat ketidakmampuan model-model sebelumnya dalam menghadapi kondisi lapangan nyata, sebagaimana ditunjukkan oleh (Andreanov Ridhovan et al., 2022; Samridhi et al., 2023), dimana performa model menurun signifikan pada variasi pencahayaan dan terbatas pada deteksi satu jenis penyakit tertentu.

Permasalahan ketiga adalah trade-off antara akurasi dan efisiensi yang belum terpecahkan. Penelitian oleh (Fitriah et al., 2022; Hindarto, 2023) mengungkapkan bahwa VGG16 memang menawarkan akurasi tinggi (98.07%) namun dengan komputasi yang lambat, sementara MobileNetV2 lebih efisien tetapi dengan akurasi yang lebih rendah (85-90%).

Pemilihan VGG16 dan MobileNetV2 dalam penelitian ini didasarkan pada kebutuhan untuk mengatasi keterbatasan-keterbatasan tersebut secara komprehensif. VGG16 dipilih karena kemampuannya dalam ekstraksi fitur kompleks yang dibutuhkan untuk identifikasi penyakit daun (Senthil Pandi et al., 2024), sementara MobileNetV2 dipilih karena desainnya yang khusus untuk kondisi bervariasi (Deng et al., 2024). Kedua arsitektur ini menawarkan solusi yang lebih seimbang antara akurasi dan efisiensi komputasi dibandingkan pendekatan-pendekatan sebelumnya. Secara khusus penelitian ini dirancang untuk melakukan analisis mendalam terhadap performa kedua arsitektur dalam hal akurasi klasifikasi multi-penyakit dan kebutuhan komputasi.

Penelitian ini memfokuskan pada akurasi dan efisiensi komputasi karena temuan terkini menunjukkan keterbatasan model *deep learning* dalam klasifikasi penyakit daun pisang pada kondisi lapangan yang bervariasi. Meskipun studi sebelumnya (Andreanov Ridhovan et al., 2022; Samridhi et al., 2023) menghasilkan akurasi >90% pada kondisi ideal, performa model sering menurun signifikan ketika menghadapi variasi pencahayaan, kotoran daun, atau sudut pengambilan gambar yang berbeda. Penelitian ini secara komprehensif mengevaluasi tidak hanya akurasi tetapi juga aspek komputasi.

Penelitian ini memberikan kontribusi signifikan terhadap pengembangan model *deep learning* untuk klasifikasi penyakit daun pisang dengan memanfaatkan arsitektur VGG16 dan MobileNetV2, sekaligus mengatasi beberapa keterbatasan yang ditemukan dalam penelitian sebelumnya. Pemilihan VGG16 dan MobileNetV2 dalam penelitian ini dirancang untuk mengatasi masalah-masalah tersebut melalui pendekatan yang lebih komprehensif dan terukur.

Pemilihan arsitektur VGG16 dan MobileNetV2 dalam penelitian ini didasarkan pada pertimbangan mendalam mengenai performa, efisiensi komputasi, dan relevansi dengan karakteristik dataset penyakit daun pisang. VGG16 dipilih karena kemampuannya dalam mengekstraksi fitur yang mendetail melalui 16 lapisan konvolusi, sehingga cocok untuk mendeteksi pola kompleks seperti bercak daun, perubahan warna, dan tekstur tidak normal yang menjadi indikator penyakit. Beberapa penelitian terdahulu, seperti yang dilakukan oleh (Senthil Pandi et al., 2024), menunjukkan bahwa VGG16 mampu mencapai akurasi hingga 97% dalam klasifikasi penyakit daun tanaman, sementara Hindarto (2023) membuktikan

keunggulannya dalam identifikasi objek biologis dengan akurasi mendekati 99%. Kelebihan ini menjadikan VGG16 sebagai kandidat kuat untuk tugas klasifikasi yang memerlukan presisi tinggi.

Di sisi lain, MobileNetV2 dipilih karena efisiensi komputasinya yang unggul, berkat penggunaan *depthwise separable convolution* yang mengurangi parameter dan beban pemrosesan tanpa mengorbankan akurasi secara signifikan. Arsitektur ini telah terbukti efektif dalam berbagai aplikasi berbasis citra, termasuk deteksi penyakit tanaman, seperti yang dilaporkan oleh (Tanwar et al., 2023) dengan akurasi 90% serta oleh (Liao et al., 2023) yang menyoroti kecepatan inferensinya yang tinggi. Selain itu, penelitian (Shetty & Mahesh, 2024) menunjukkan bahwa MobileNetV2 tetap menjaga akurasi di atas 90% bahkan pada dataset berskala besar, menjadikannya solusi ideal untuk implementasi di perangkat dengan sumber daya terbatas, seperti sistem berbasis *edge computing*.

Kombinasi kedua arsitektur ini memungkinkan penelitian untuk melakukan analisis komparatif yang komprehensif, mencakup aspek akurasi, kecepatan pelatihan, dan efisiensi sumber daya. Studi sebelumnya, seperti yang dilakukan oleh (Shi, 2024), telah membandingkan VGG16 dan MobileNetV2 dalam konteks klasifikasi citra medis, menyimpulkan bahwa VGG16 cenderung lebih akurat, sementara MobileNetV2 lebih efisien. Meskipun kedua model tersebut memiliki karakteristik yang berbeda, performa dari VGG16 dan MobileNetV2 dalam klasifikasi penyakit daun pisang sangat bergantung pada pengaturan parameter pelatihan, khususnya pemilihan *optimizer* dan *learning rate* yang digunakan (Shi, 2024). *Optimizer* berperan penting dalam menentukan arah dan kecepatan hasil

selama proses pelatihan, sedangkan *learning rate* memengaruhi stabilitas serta kecepatan pelatihan. Pemilihan kombinasi yang tidak tepat dapat menyebabkan model gagal mencapai hasil pelatihan atau mengalami *overfitting*, sehingga menurunkan hasil akurasi klasifikasi (Prasetyo & Utami, 2024).

Dengan mempertimbangkan temuan tersebut, penelitian ini bertujuan untuk mengevaluasi model mana yang lebih optimal untuk klasifikasi penyakit daun pisang, baik dalam skenario komputasi tinggi maupun terbatas. Selain itu, pemilihan kedua model ini juga didukung oleh penelitian (Prasetyo & Utami, 2024) yang menyatakan bahwa VGG16 dan MobileNetV2 mampu memberikan solusi klasifikasi dengan akurasi tinggi tanpa kompleksitas komputasi yang berlebihan, sehingga sesuai untuk pengembangan sistem deteksi penyakit tanaman yang praktis dan skalabel.

Analisis performa VGG16 dan MobileNetV2 dalam klasifikasi penyakit daun pisang diharapkan dapat memberikan kontribusi signifikan terhadap pengembangan deteksi penyakit tanaman yang lebih akurat dan efisien. Penelitian ini juga bertujuan untuk memberikan pemahaman yang lebih mendalam mengenai penerapan teknologi CNN, khususnya VGG16 dan MobileNetV2 dalam meningkatkan efektivitas deteksi penyakit tanaman, serta memberikan wawasan terkait implementasinya dalam bidang pertanian.

1.2. Rumusan Masalah

Rumusan masalah dari analisis performa MobileNetV2 dalam klasifikasi penyakit daun pisang menggunakan CNN adalah sebagai berikut:

1. Bagaimana performa MobileNetV2 dan VGG16 dalam klasifikasi penyakit daun pisang?
2. Apakah penggunaan MobileNetV2 dan VGG16 dapat meningkatkan akurasi pada klasifikasi penyakit daun pisang?
3. Bagaimana kombinasi *optimizer* dan *learning rate* memengaruhi performa VGG16 dan MobileNetV2 dalam klasifikasi penyakit daun pisang?

1.3. Batasan Masalah

Berikut ini adalah rincian batasan masalah yang terdapat pada analisis performa VGG16 dan MobileNetV2 dalam klasifikasi penyakit daun pisang menggunakan CNN:

1. Penelitian ini berfokus pada klasifikasi penyakit pada daun pisang yang terinfeksi dan daun pisang sehat.
2. Dataset diambil dari sumber terbuka, seperti Kaggle yang sudah diorganisir ke dalam folder sesuai kategorinya.
3. Solusi yang diusulkan dalam penelitian ini terbatas pada pengembangan dan pelatihan model VGG16 dan MobileNetV2 untuk tujuan klasifikasi. Hal ini termasuk pengoptimalan model dalam hal akurasi dan waktu pemrosesan.
4. Analisis hasil terdapat pada matrik evaluasi yang umum digunakan dalam klasifikasi, seperti akurasi, F1-Score, presisi, recall dan *training time*.

1.4. Tujuan Penelitian

Penelitian ini akan mencapai titik terminasi ketika seluruh tujuan utama telah terpenuhi melalui pencapaian indikator-indikator yang terukur, yang meliputi terlaksananya seluruh proses pelatihan dan evaluasi model secara komprehensif

dengan diperolehnya metrik performa yang mencakup akurasi, presisi, recall, F1-score, serta *training time*.

Selanjutnya yaitu telah mendapatkan hasil akurasi terhadap *baseline* model sebelumnya beserta identifikasi faktor-faktor kritis yang mempengaruhi akurasi, termasuk efek augmentasi data dan *transfer learning*, sehingga dapat memberikan gambaran menyeluruh tentang efektivitas kedua arsitektur *deep learning* dalam klasifikasi penyakit daun pisang.

Berikut ini adalah tujuan penelitian analisis performa VGG16 dan MobileNetV2 untuk klasifikasi penyakit daun pisang:

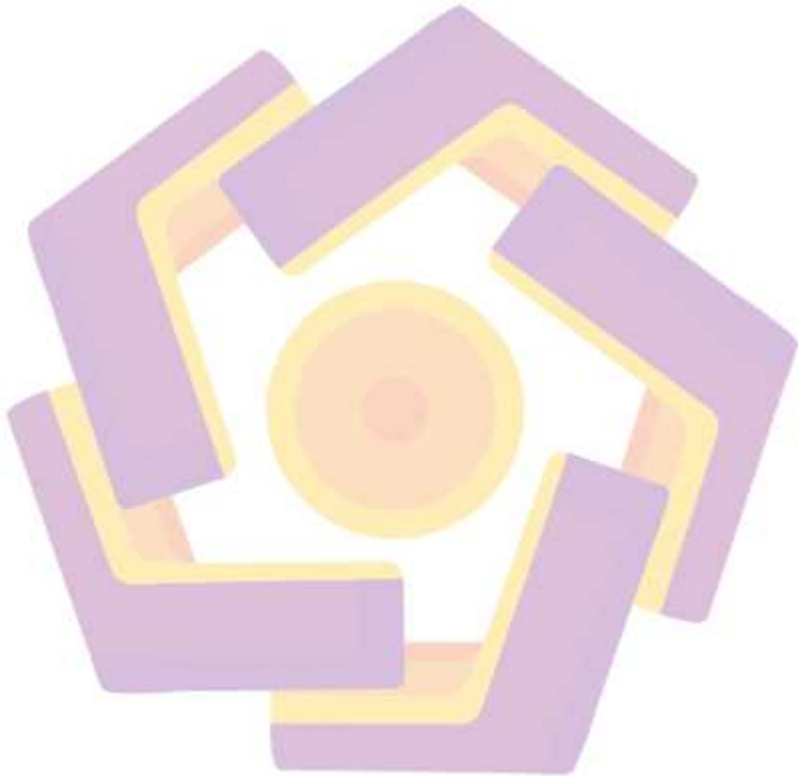
1. Menganalisis performa VGG16 dan MobileNetV2 dalam klasifikasi penyakit daun pisang.
2. Mengukur dampak penggunaan VGG16 dan MobileNetV2 terhadap akurasi klasifikasi penyakit daun pisang

1.5. Manfaat Penelitian

Manfaat dari penelitian terkait analisis performa VGG16 dan MobileNetV2 dalam klasifikasi penyakit daun pisang sebagai berikut:

1. Dengan membandingkan dua arsitektur, penelitian ini berpotensi meningkatkan akurasi dalam mendeteksi berbagai jenis penyakit pada daun pisang.
2. Dengan kemampuan mendeteksi penyakit secara akurat dan cepat, petani bisa mendapatkan informasi yang lebih baik dalam pengambilan keputusan terkait perawatan tanaman, termasuk pemilihan pestisida dan strategi pengendalian hama yang lebih efektif.

3. Hasil analisis performa ini dapat dijadikan basis data untuk penelitian lebih lanjut mengenai pengembangan model CNN lainnya serta aplikasi kecerdasan buatan dalam bidang pertanian.



BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Berdasarkan sistematik literatur review yang dilakukan oleh (Prasetyo & Utami, 2024), hasil penelitian ini menjelaskan tentang penggunaan dataset dan algoritma yang populer di bidang penyakit daun pisang. Penggunaan dataset 1.600 gambar yang telah melalui proses augmentasi. Penelitian ini menjelaskan ada 7 jenis penyakit populer pada daun pisang seperti Pestalotiopsis 5%, Cucumber Mosaic Virus 5%, Cordana 8%, Fusarium Wilt 15%, Bunchy Top 13%, Bacterial Wilt 18%, dan Sigatoka 36%. Sementara itu, penelitian SLR menjelaskan algoritma populer dibidang ini yaitu CNN, Artificial Neural Network (ANN), K-Means Clustering, Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN) dan Ensemble Approach.

Berikut ini beberapa penelitian eksperimen yang sudah dilakukan pada bidang deteksi penyakit pada daun pisang menggunakan pendekatan *machine learning* maupun *deep learning*.

Penelitian oleh (M. Kumar & Kumar, 2024) mengembangkan model hybrid CNN-SVM untuk menilai keparahan penyakit layu daun pisang, dengan akurasi 94,77%. Kelebihan dari pendekatan ini adalah kemampuannya untuk menggabungkan fitur ekstraksi dan klasifikasi yang meningkatkan akurasi. Sebaliknya, kelemahannya terletak pada penggunaan arsitektur model terbatas yang dapat mempengaruhi presisi dan kepraktisan evaluasi tingkat keparahan penyakit.

Penelitian oleh (Tanwar et al., 2023) menerapkan model hybrid CNN-SVM untuk mengidentifikasi berbagai penyakit daun pisang, mencapai akurasi 90%. Kelebihan dari penelitian ini adalah kemampuannya untuk mendeteksi beberapa jenis penyakit, namun kelemahan terletak pada jumlah gambar yang terbatas, yang berpotensi memengaruhi keakuratan pada kondisi nyata di lapangan.

Penelitian oleh (Shetty & Mahesh, 2024) mengembangkan metode segmentasi berbasis densitas gravitasi kernel spasial (SKGDC) untuk mendeteksi dan memprediksi penyakit pada daun, buah, dan batang pisang. Dengan menggunakan model hybrid HCA-YOLOv8, penelitian ini mencapai akurasi tinggi, yaitu 98.12% untuk BananaLSD, 99% untuk Banana Leaf, dan 98.5% untuk PSFD-Musa. Kelebihan utama dari penelitian ini adalah akurasi yang sangat baik dalam klasifikasi penyakit pada tanaman pisang. Namun, tantangan utama dari metode ini adalah kompleksitas implementasi yang tinggi, mengingat banyaknya metode preprocessing yang digunakan, seperti Up-SGF dan IWCF.

Penelitian oleh (Liao et al., 2023) berhasil mengembangkan basis data suara pemapasan orang dewasa China (LD-DF RSdb) dan menggunakan MobileNetV2 untuk deteksi suara pernapasan abnormal. Dengan mengumpulkan 9.584 rekaman suara dari 145 peserta, penelitian ini menunjukkan performa yang baik dengan area under the curve (AUC) sebesar 0.8923. Meskipun MobileNetV2 dikenal efisien dan ringan, kelemahan dari penelitian ini adalah penggunaan satu model yang mungkin tidak mampu menangani variabilitas kompleksitas suara pernapasan yang lebih luas.

Penelitian (S. Kumar & Kumar, 2024) mengembangkan metode baru untuk klasifikasi gambar X-ray dada menggunakan Machine Learning dan Transfer Learning dengan Efficient-VGG16 sebagai pendekatan ensemble yang menggabungkan arsitektur EfficientNet-B0 dan VGG16. Model ini berhasil mencapai akurasi tertinggi 99.46% dan F1 score 98.41%, mengungguli pendekatan lain seperti EfficientNet-B0 dan XgBoost. Meskipun efisien dalam meningkatkan akurasi, kelemahan dari metode ini adalah peningkatan kompleksitas dan kebutuhan waktu komputasi yang lebih besar, yang dapat membatasi penerapan dalam skala besar.

Penelitian yang dilakukan oleh (Samridhi et al., 2023) menggunakan CNN untuk mendeteksi penyakit Sigatoka pada daun pisang dan mencapai akurasi 96,41%. Kelebihan dari penelitian ini adalah fokus pada deteksi dini, yang penting untuk mengurangi kerugian hasil panen. Namun, kelemahannya adalah hanya berfokus pada satu jenis penyakit, sehingga mempersempit aplikabilitas model di lapangan.

Penelitian oleh (Shifat E. Arman, 2024) menggunakan Banana SqueezeNet dalam mengklasifikasi penyakit daun pisang dengan akurasi yang memadai. Meski demikian, tantangan dalam hal generalisasi hasil untuk kondisi yang lebih kompleks menunjukkan perlunya penyempurnaan teknik dalam pemrosesan citra dan ekstraksi fitur.

Penelitian (Chaudhari & Patil, 2023) menggunakan teknik klasifikasi ensemble dengan hasil akurasi 92%. Kelebihan dari pendekatan ini adalah kemampuannya dalam menangani berbagai jenis patogen, tetapi kelemahannya

adalah penggunaan sumber daya komputasi yang lebih besar, yang dapat menghambat implementasi praktis.

Penelitian yang dilakukan oleh (Deng et al., 2024) mengembangkan metode berbasis KVA dan GR-ARNet, mencapai akurasi diatas 95% untuk tiga jenis penyakit. Kelebihan dari penelitian ini adalah penggunaan teknik pengolahan citra yang canggih, tetapi kelemahannya terdapat pada kompleksitas model yang lebih tinggi, yang dapat menyulitkan implementasinya di lapangan.

Penelitian yang dilakukan oleh (Andreanov Ridhovan et al., 2022b) menunjukkan hasil yang mengesankan dengan menerapkan DenseNet dan Inception dan mencapai akurasi 84,73% pada dataset yang lebih kecil, penelitian tersebut juga mengakui adanya keterbatasan data sebagai tantangan utama. Kelemahan pada penelitian ini belum secara eksplisit menguji model pada data dengan kondisi pencahayaan yang beragam, yang bisa mempengaruhi generalisasi model.

Penelitian oleh (Seetharaman & Mahendran, 2022) menggunakan teknik Gabor Extraction dan RCNN untuk mendeteksi penyakit pada daun pisang dan mencapai akurasi 98%. Kelebihan dari penelitian ini adalah akurasinya yang tinggi. Namun, kelemahan terletak pada waktu komputasi yang lebih lama, yang bisa menjadi masalah dalam konteks penerapan di lapangan.

Penelitian oleh (Hindarto, 2023) menunjukkan perkembangan signifikan dalam identifikasi ikan melalui evaluasi komparatif dua arsitektur deep learning, yaitu VGG16 dan MobileNetV2. Kelebihan penelitian ini adalah penyediaan wawasan dalam pemilihan model deep learning yang sesuai dengan kebutuhan

spesifik, seperti memprioritaskan akurasi atau efisiensi. Namun, terdapat kekurangan yang perlu diperhatikan, yaitu keterbatasan dataset yang digunakan, yang belum mencakup variasi spesies ikan secara luas. Hal ini dapat mempersulit generalisasi hasil penelitian ke skenario dunia nyata dengan jenis ikan yang lebih beragam.

Penelitian yang dilakukan oleh (Shi, 2024) membandingkan performa MobileNetV2, VGG16, dan ResNet50 untuk klasifikasi tumor otak pada citra MRI, dimana ResNet50 menunjukkan akurasi 92.05%, diikuti VGG16 91.13% dan MobileNetV2 85.32%. Kelebihan penelitian ini terletak pada komparasi menyeluruh antar arsitektur CNN, namun kelemahannya adalah hanya menggunakan satu model untuk setiap arsitektur sehingga hasil yang diperoleh kurang optimal untuk generalisasi.

Penelitian (Senthil Pandi et al., 2024) berfokus pada peningkatan akurasi deteksi penyakit daun apel menggunakan VGG16 yang mencapai akurasi tinggi 97% melalui pendekatan hybrid. Keunggulan penelitian ini adalah penggunaan kombinasi CNN, LBP, dan *ensemble optimizer*, namun memiliki keterbatasan dalam kompleksitas model yang tinggi dan tidak dilakukan perbandingan dengan arsitektur lain.

Penelitian (Alex David et al., 2024) memanfaatkan VGG16 dan MobileNetV2 untuk klasifikasi tumor otak, dimana VGG16 menunjukkan performa lebih akurat dibanding MobileNetV2. Penelitian ini memberikan kontribusi dalam pengembangan diagnostik MRI berbasis *deep learning*, meskipun masih

menghadapi tantangan dalam interpretabilitas model terhadap data MRI yang kompleks.

Penelitian yang dilakukan (Haris Rangkuti et al., 2023) mengembangkan sistem retrieval gambar pisang berbasis CNN dengan akurasi mencapai 95% menggunakan EfficientNetV2. Kelebihan penelitian ini terletak pada analisis perbandingan metode pengukuran jarak gambar, namun perlu pengembangan lebih lanjut untuk variasi buah dan kondisi gambar yang lebih beragam.

Penelitian oleh (Fitriana et al., 2022) melakukan fine-tuning MobileNetV2 dan VGG16 untuk klasifikasi gambar ikan, dengan hasil VGG16 memberikan akurasi lebih tinggi 98,07% sementara MobileNetV2 lebih efisien. Penelitian ini memperkuat temuan sebelumnya tentang trade-off antara akurasi dan efisiensi pada arsitektur CNN, meskipun masih terdapat kendala dalam optimalisasi parameter dan segmentasi gambar.

Dari beberapa penelitian tersebut, meskipun terdapat kemajuan yang signifikan dalam penggunaan *deep learning* untuk deteksi penyakit daun pisang, tantangan dalam hal efisiensi, generalisasi model, dan kecepatan deteksi masih perlu ditangani. Oleh karena itu, penelitian ini bertujuan untuk menyempurnakan metode yang ada dengan menerapkan arsitektur CNN yang lebih efisien dan teknik augmentasi data yang optimal.

2.2. Keaslian Penelitian

Dibawah ini merupakan diagram alir yang menjelaskan perbandingan antara penelitian sebelumnya dengan penelitian saat ini terkait klasifikasi penyakit daun pisang menggunakan arsitektur CNN yang dapat dilihat pada Gambar 2.1.



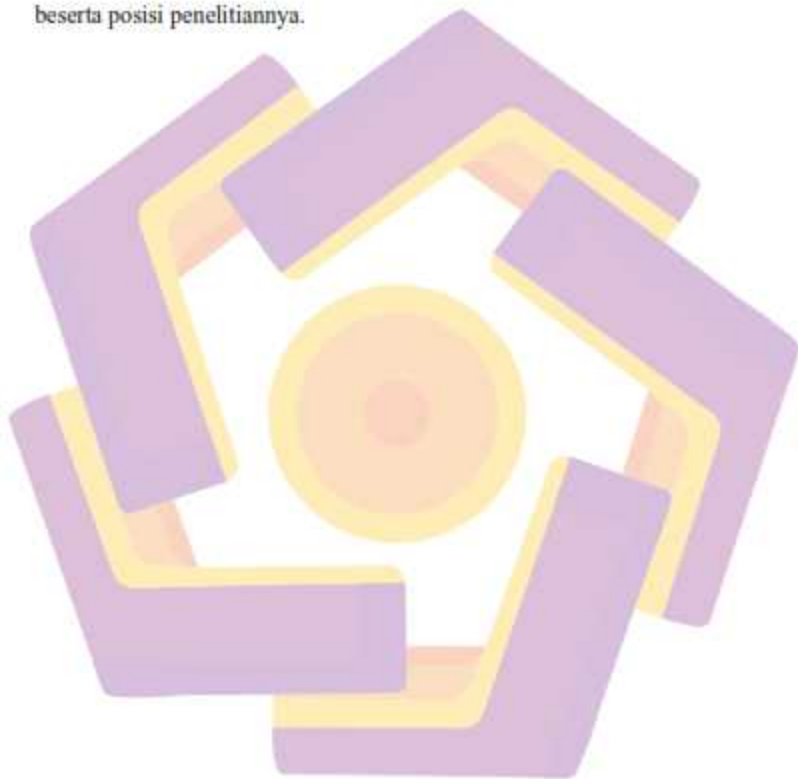
Gambar 2.1 Perbandingan Penelitian Sebelumnya dan Penelitian Saat Ini

Secara umum, penelitian sebelumnya menggunakan berbagai metode seperti CNN (LeNet, AlexNet, VGG, MobileNetV2), *transfer learning* (dengan arsitektur ResNet50 dan EfficientNet), serta pendekatan machine learning konvensional seperti SVM dan Random Forest yang mengandalkan ekstraksi fitur manual. Selain itu, metode deteksi objek seperti YOLO dan pendekatan hibrida CNN + SVM juga telah diterapkan (Ibarra et al., 2023; Senthil Pandi et al., 2024).

Jenis penelitian sebelumnya umumnya terbatas pada klasifikasi satu jenis penyakit, segmentasi area daun, deteksi objek, dan pengembangan aplikasi mobile yang tidak menitikberatkan pada akurasi. Penelitian tersebut juga cenderung dilakukan tanpa evaluasi efisiensi waktu. Meskipun beberapa penelitian menunjukkan tingkat akurasi yang tinggi (>90%), namun terdapat beberapa kelemahan seperti penggunaan dataset yang kecil dan kurang bervariasi, hanya memanfaatkan satu model tanpa perbandingan, serta fokus semata pada akurasi tanpa mempertimbangkan efisiensi dan keberagaman data. Selain itu, sebagian besar penelitian sebelumnya belum mempertimbangkan daun sehat sebagai kontrol dalam klasifikasi (Fitriana et al., 2022).

Sebagai perbandingan, penelitian saat ini menggunakan pendekatan dengan model yang ringan dan membandingkan performa dua arsitektur CNN yaitu VGG16 dan MobileNetV2. Evaluasi dilakukan secara menyeluruh menggunakan metrik evaluasi serta memanfaatkan dataset yang telah diperluas (augmented) untuk mencakup variasi yang lebih beragam (Hadi et al., 2024). Fokus utama dari penelitian ini adalah pada efisiensi dan akurasi klasifikasi, dengan

mempertimbangkan daun sehat sebagai kontrol untuk meningkatkan validitas hasil klasifikasi penyakit daun pisang. Berdasarkan tinjauan terhadap penelitian-penelitian terkait, Tabel 2.1 menyajikan matriks literatur review yang mengkaji berbagai pendekatan dalam deteksi penyakit tanaman, khususnya daun pisang, beserta posisi penelitiannya.



Tabel 2. 1 Matriks Literatur Review dan Posisi Penelitian

No	Judul	Peneliti, Tahun, Media Publikasi	Metode Penelitian	Keunggulan dan Kelemahan	Perbandingan
1	Deep Learning Meets Support Vector Machines: An Effective Hybrid Model for Banana Leaf Wilt Disease Severity Assessment	(M. Kumar & Kumar, 2024) / ICDT	CNN dan SVM	Penggunaan arsitektur model yang terbatas dapat mempengaruhi presisi dan kepraktisan evaluasi tingkat keparahan penyakit.	Penelitian ini membandingkan performa CNN (VGG16 & MobileNetV2) sebagai alternatif yang lebih sederhana dan fokus pada efisiensi.
2	Implementing a Hybrid CNN-SVM Model for Banana Leaf Disease Classification	(Tanwar et al., 2023)	Algoritma <i>deep learning</i> .	Dataset yang lebih kecil dapat mempengaruhi generalisasi model	MobileNetV2 lebih efisien untuk dataset besar, dengan tetap mempertahankan akurasi tinggi.
3	Identification of Sigatoka Leaf Spot Disease in Banana Using Convolutional Neural Network (CNN)	(Samridhi et al., 2023)	<i>Deep learning</i> berbasis CNN.	Kelemahannya adalah penelitian ini hanya berfokus pada satu jenis penyakit, sehingga mempersempit aplikabilitas model.	Penelitian ini memperluas cakupan klasifikasi dengan lebih banyak jenis penyakit, termasuk Sigatoka, untuk pengujian lebih menyeluruh.
4	BananaLSD: A banana leaf images dataset for classification of banana leaf diseases	(Shifat E. Arman, 2024) / Elsevier	<i>Machine learning</i>	Kelemahan dari penelitian ini terletak pada tantangan dalam menggeneralisasi hasil untuk kondisi yang lebih kompleks, yang menunjukkan perlunya	Penelitian ini menggunakan dataset serupa dengan fokus pada perbandingan VGG16 dan MobileNetV2 untuk melihat efektivitas pada kondisi kompleks.

No	Judul	Peneliti, Tahun, Media Publikasi	Metode Penelitian	Keunggulan dan Kelemahan	Perbandingan
	using machine learning			penyempurnaan lebih lanjut dalam teknik pemrosesan citra dan ekstraksi fitur.	
5	Detection and Classification of Banana Leaf Disease Using Novel Segmentation and Ensemble Machine Learning Approach	(Chaudhari & Patil, 2023b)	Machine learning	Kelemahannya adalah penggunaan sumber daya komputasi yang lebih besar, yang dapat menghambat implementasi praktis.	MobileNetV2 menawarkan solusi dengan akurasi sebanding tetapi lebih hemat sumber daya.
6	Identification of banana leaf disease based on KVA and GR-ARNet	(Deng et al., 2024) / Elsevier	KVA dan GR-ARNet	Kelemahannya terdapat pada kompleksitas model yang lebih tinggi, yang dapat menyulitkan implementasinya di lapangan.	MobileNetV2 menawarkan keseimbangan antara akurasi dan efisiensi tanpa memerlukan arsitektur yang lebih rumit.
7	Disease Detection in Banana Leaf Plants using DenseNet and Inception Method	(Andranov Ridhovan et al., 2022) / RESTI	Metode DenseNet dan Inception	Penelitian ini tidak diuji pada kondisi pencahayaan yang beragam.	MobileNetV2 didesain untuk menangani berbagai kondisi pencahayaan dengan efisiensi lebih tinggi.
8	Leaf Disease Detection in Banana Plant using Gabor Extraction and Region-Based Convolution Neural Network (RCNN)	(Seetharaman & Mahendran, 2022) / Springer	Gabor Extraction dan Region-Based CNN	Kelemahan terletak pada waktu komputasi yang lebih lama, yang bisa menjadi masalah dalam konteks penerapan di lapangan.	Penelitian ini membandingkan VGG16 dan MobileNetV2, yang menawarkan waktu komputasi lebih cepat.

No	Judul	Peneliti, Tahun, Media Publikasi	Metode Penelitian	Keunggulan dan Kelemahan	Perbandingan
9	SKGDC: Effective Segmentation Based Deep Learning Methodology for Banana Leaf, Fruit, and Stem Disease Prediction	(Shetty & Mahesh, 2024)	Metode segmentasi berbasis densitas gravitasi kernel spasial (SKGDC).	Kompleksitas implementasi tinggi karena preprocessing melibatkan banyak metode, seperti Up-SGF dan IWCF.	Penelitian ini memiliki kompleksitas model yang lebih moderat, karena hanya menggunakan dua arsitektur yang lebih sederhana dan efisien, yaitu VGG16 dan MobileNetV2, yang dirancang untuk aplikasi yang lebih ringan dengan fokus pada klasifikasi penyakit daun pisang.
10	Automated detection of abnormal respiratory sound from electronic stethoscope and mobile phone using MobileNetV2	(Liao et al., 2023)	MobileNetV2	Meskipun MobileNetV2 adalah model yang ringan dan efisien, namun penggunaan satu model saja mungkin tidak mencakup kompleksitas variabilitas suara permapasan yang lebih luas.	VGG16 dan MobileNetV2 dibandingkan untuk melihat apakah model yang lebih ringan seperti MobileNetV2 dapat memberikan hasil yang setara atau lebih baik dibandingkan dengan model yang lebih besar dan lebih kompleks seperti VGG16, terutama dalam hal efisiensi dan waktu proses.
11	Efficient-VGG16: A Novel Ensemble Method for the Classification of COVID-19 X-ray Images in Contrast to Machine and Transfer Learning	(S. Kumar & Kumar, 2024) / Elsevier	VGG16 (<i>Machine learning dan transfer learning</i>)	Model ensemble seperti Efficient-VGG16 dapat meningkatkan akurasi, namun juga menambah kompleksitas dan memerlukan lebih banyak waktu komputasi, yang dapat membatasi kecepatan penerapan dalam skala besar.	MobileNetV2 adalah arsitektur yang lebih ringan, efisien dalam penggunaan daya komputasi dan waktu, serta cocok dalam penerapan skala yang besar.

No	Judul	Peneliti, Tahun, Media Publikasi	Metode Penelitian	Keunggulan dan Kelemahan	Perbandingan
12	Comparative Analysis VGG16 Vs MobileNet Performance for Fish Identification	(Hindarto, 2023) / JTIE	VGG16 dan MobileNetV2	Dataset gambar ikan yang digunakan belum mencakup variasi spesies ikan yang cukup luas, sehingga hasil penelitian bisa sulit untuk digeneralisasikan ke skenario dunia nyata dengan jenis ikan yang lebih beragam.	Dataset penyakit daun pisang mencakup beberapa kategori. Hal ini memungkinkan pengujian yang lebih spesifik dengan hasil yang lebih relevan untuk skenario tertentu.
13	A Comparison of MobileNetV2, VGG16, and ResNet50 for Classifying Brain Tumors	(Shi, 2024) / IEEE ICEACE	Model CNN (MobileNetV2, VGG16, ResNet50)	Hanya menggunakan satu model, sehingga menghasilkan hasil yang kurang akurat. Disarankan menggunakan lebih dari satu model untuk eksplorasi lebih lanjut untuk optimasi	Penelitian ini menggunakan dua arsitektur, yaitu VGG16 dan MobileNetV2
14	Accuracy Enhancement in Apple Leaf Diseases Detection and Classification Using VGG16	(Senthil Pandi et al., 2024) / IEEE EDKCON	VGG16	Kompleksitas model cukup tinggi, dan tidak dibandingkan dengan model atau arsitektur lain	Penelitian ini tidak hanya berfokus pada VGG16 tetapi menggunakan dua arsitektur, yaitu VGG16 dan MobileNetV2
15	Deep Learning Advances in Brain Tumor Classification: Leveraging VGG16 and MobileNetV2 for	(Alex David et al., 2024) / IEEE ICPECTS	VGG16 dan MobileNetV2	Terdapat kelemahan pada interpretabilitas AI masih menjadi tantangan data MRI kompleks	Model dan perbandingan sama, namun konteks berbeda dan validasi hasil serupa dengan objek berbeda

No	Judul	Peneliti, Tahun, Media Publikasi	Metode Penelitian	Keunggulan dan Kelemahan	Perbandingan
	Accurate MRI Diagnostics				
16	Development of Banana Image Retrieval Based on its Class Using the CNN Model and Image Distance	(Haris Rangkuti et al., 2023) / IEEE ICOCO	<i>Convolution Neural Network (CNN)</i>	Perlu eksplorasi lebih lanjut pada buah lain dengan penggunaan gambar, rotasi dan skala bervariasi	Sama-sama menggunakan citra pisang, tetapi fokus penelitian berbeda (retrieval -klasifikasi penyakit)
17	Fine-Tuned MobileNetV2 and VGG16 Algorithm for Fish Image Classification	(Fitriah et al., 2022) / IEEE ICIMCIS	MobileNetV2 dan VGG16	Masalah penelitian terdapat pada segmentasi gambar ikan dan tuning parameter yang optimal	Penelitian ini mendukung temuan bahwa VGG16 unggul dalam akurasi, MobileNetV2 unggul dalam efisiensi tetapi dengan dataset yang berbeda

2.3. Landasan Teori

Berikut penjelasan detail terkait dengan landasan teori yang terdapat pada penelitian ini.

2.3.1 *Convolutional Neural Network (CNN)*

CNN merupakan salah satu jenis jaringan saraf tiruan yang dirancang khusus untuk mengolah data berupa gambar. CNN bekerja dengan cara mengaplikasikan operasi konvolusi pada gambar untuk mengekstrak fitur-fitur penting (Rifkie Primartha, 2021). Operasi ini dilakukan melalui lapisan-lapisan konvolusi, diikuti dengan lapisan pooling untuk mengurangi dimensi data, dan lapisan *fully connected* untuk klasifikasi. CNN dikenal efektif dalam menangani masalah klasifikasi gambar karena kemampuannya dalam mempelajari fitur hierarkis, dari fitur lokal hingga fitur global (Suri & Cabri, 2014). Pada dasarnya, CNN terdiri dari tiga jenis lapisan utama:

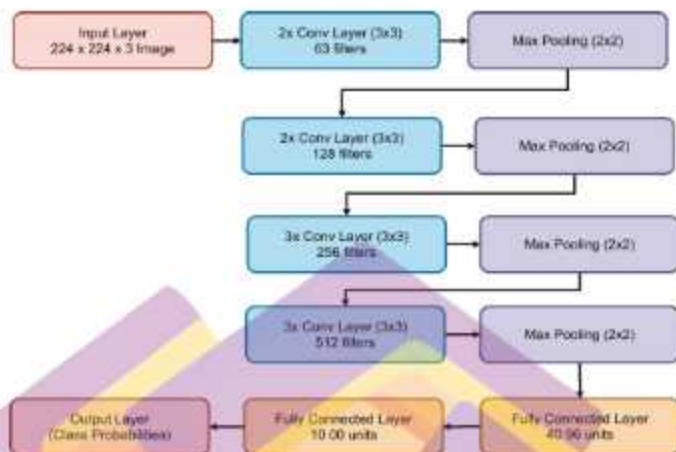
1. Lapisan Konvolusi berfungsi untuk mengekstraksi fitur dari gambar input dengan menggunakan filter atau kernel yang digeser melalui gambar.
2. Lapisan Pooling berfungsi untuk mereduksi dimensi dan mengurangi kompleksitas model, dengan cara mengambil nilai maksimum atau rata-rata dari sekelompok pixel dalam gambar.
3. Lapisan Fully Connected untuk menyambungkan semua neuron di lapisan sebelumnya dengan neuron pada lapisan keluaran, yang digunakan untuk menentukan kelas dari gambar yang diklasifikasikan.

Kelebihan CNN dalam klasifikasi gambar terletak pada kemampuannya untuk melakukan ekstraksi fitur secara otomatis, tanpa memerlukan proses pra-pemrosesan yang kompleks, sehingga sangat efektif dalam berbagai aplikasi pengenalan pola (Raharjo, 2022).

2.3.2 VGG16

VGG16 adalah salah satu arsitektur CNN yang dikembangkan oleh *Visual Geometry Group* di *University of Oxford* pada tahun 2014. VGG16 memiliki kedalaman 16 lapisan, yang terdiri dari 13 lapisan konvolusi dan 3 lapisan fully connected. Ciri khas dari arsitektur ini adalah penggunaan filter berukuran kecil (3x3) dalam setiap lapisan konvolusi, yang memungkinkan jaringan untuk menangkap berbagai tingkat fitur dengan resolusi tinggi (Suri & Cabri, 2014).

VGG16 telah terbukti sangat efektif dalam berbagai tugas klasifikasi gambar, termasuk dalam kompetisi ImageNet (Vakalopoulou et al., 2023). Namun, meskipun performanya sangat baik, VGG16 memiliki jumlah parameter yang sangat besar, sehingga membutuhkan waktu pelatihan yang cukup lama dan memerlukan sumber daya komputasi yang tinggi. Oleh karena itu, VGG16 lebih cocok digunakan pada perangkat dengan kapasitas komputasi yang memadai. Berikut adalah gambaran arsitektur VGG16 yang dapat dilihat pada Gambar 2.2.



Gambar 2.2 Arsitektur VGG16

Gambar 2.2 menunjukkan arsitektur VGG16 yang digunakan untuk klasifikasi citra. Berikut adalah penjelasan gambaran arsitekturnya:

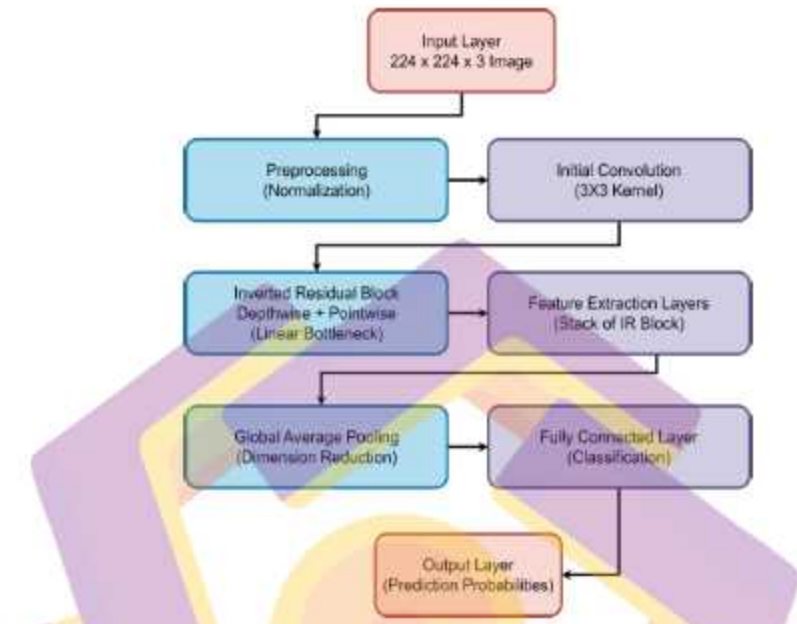
1. Input Layer: Memasukkan citra dengan ukuran $224 \times 224 \times 3$ (RGB image).
2. Conv Layer: Menggunakan beberapa lapisan konvolusi 3×3 untuk mengekstraksi fitur dari citra. Jumlah filter bertambah secara bertahap ($63 \rightarrow 128 \rightarrow 256 \rightarrow 512$) untuk menangkap pola yang lebih kompleks.
3. Max Pooling Layer: Digunakan setelah beberapa lapisan konvolusi untuk mengurangi dimensi data (*down-sampling*) tanpa kehilangan fitur penting.
4. Fully Connected Layer: Terdiri dari dua lapisan dengan masing-masing 10.000 dan 4.096 unit, yang bertugas melakukan klasifikasi berdasarkan fitur yang telah diekstraksi.
5. Output Layer: Memberikan probabilitas untuk setiap kelas yang akan diprediksi.

2.3.3 MobileNetV2

MobileNetV2 adalah arsitektur CNN yang dirancang untuk perangkat dengan sumber daya terbatas, seperti perangkat mobile atau embedded systems. Arsitektur ini dikembangkan oleh Google pada tahun 2018 dan dikenal karena efisiensinya dalam hal ukuran model dan kecepatan komputasi (Rifkie Primartha, 2018). MobileNetV2 menggunakan konsep inverted residuals dan depthwise separable convolutions, yang memungkinkan pengurangan jumlah parameter dan komputasi, tetapi tetap mempertahankan akurasi yang tinggi (Vakalopoulou et al., 2023).

MobileNetV2 mengadopsi lapisan konvolusi dengan pembagian fungsi konvolusi menjadi dua tahap: pertama, menggunakan konvolusi depthwise untuk mengurangi dimensi, kemudian diikuti dengan konvolusi pointwise untuk meningkatkan dimensi. Pendekatan ini menghasilkan model yang lebih ringan namun tetap cukup akurat untuk klasifikasi gambar (Vakalopoulou et al., 2023).

Karena ukurannya yang lebih kecil dan kecepatan komputasi yang lebih tinggi, MobileNetV2 sangat cocok untuk aplikasi yang memerlukan pengolahan gambar secara real-time pada perangkat dengan kapasitas komputasi terbatas, seperti smartphone atau perangkat IoT (Rifkie Primartha, 2018). Berikut adalah gambar arsitektur MobileNetV2 yang dapat dilihat pada Gambar 2.3.



Gambar 2.3 Arsitektur MobileNetV2

Gambar 2.3 menunjukkan arsitektur MobileNetV2 yang digunakan untuk klasifikasi penyakit daun pisang. Berikut adalah penjelasan gambaran dari arsitektur MobileNetV2:

1. Input Layer: Menerima citra dengan dimensi $224 \times 224 \times 3$.
2. Preprocessing (Normalization): Normalisasi data sebelum pemrosesan untuk mempercepat pelatihan dan meningkatkan performa.
3. Initial Convolution: Lapisan konvolusi awal dengan kernel 3×3 untuk menangkap fitur dasar.
4. Inverted Residual Block: Blok utama arsitektur MobileNetV2 yang menggabungkan *depthwise convolution* (memproses setiap channel secara

terpisah) dan *pointwise convolution* (menggabungkan informasi antar-channel). Blok ini membuat model lebih ringan dan efisien.

5. Global Average Pooling: Mengurangi dimensi fitur menjadi vektor satu dimensi untuk mempermudah proses klasifikasi.
6. Fully Connected Layer: Melakukan klasifikasi berdasarkan fitur akhir.
7. Output Layer: Memberikan hasil probabilitas untuk setiap kelas.

2.3.4 Klasifikasi Penyakit Daun Pisang dengan CNN

Penyakit pada tanaman pisang, seperti penyakit daun pisang, dapat dikenali melalui tanda-tanda visual pada daun yang terinfeksi. Pengenalan otomatis penyakit daun pisang menggunakan CNN dapat membantu dalam mendeteksi penyakit dengan cepat dan akurat, sehingga memudahkan petani dalam menangani penyakit secara efisien (Vakalopoulou et al., 2023). Untuk melakukan klasifikasi, dataset gambar daun pisang yang mengandung berbagai jenis penyakit dan kondisi daun sehat diperlukan sebagai input ke dalam model CNN.

Model CNN yang telah dilatih untuk mengenali fitur-fitur khas dari daun pisang yang terinfeksi dapat menghasilkan prediksi yang cepat dan akurat. Arsitektur seperti VGG16 dan MobileNetV2 sering digunakan dalam penelitian ini untuk membandingkan kinerja antara model yang lebih berat (VGG16) dengan model yang lebih ringan dan efisien (MobileNetV2), untuk mengetahui kelebihan dan kekurangan masing-masing arsitektur dalam konteks klasifikasi penyakit daun pisang (Rifkie Primartha, 2021). Berikut adalah penjelasan mengenai tiga jenis penyakit yang sering ditemukan pada daun pisang:

2.3.5 Klasifikasi Penyakit Daun Pisang dengan CNN

Pada klasifikasi penyakit daun pisang terdiri dari tiga jenis penyakit yang terinfeksi, sebagai berikut:

2.3.5.1 Cordana

Penyakit Cordana disebabkan oleh jamur *Cordana* yang menginfeksi daun pisang. Penyakit ini biasanya dikenal dengan gejala bercak-bercak kecil berwarna coklat atau hitam pada permukaan daun. Bercak-bercak ini bisa berkembang dan menyebabkan daun mengering atau bahkan gugur jika infeksi cukup parah. Cordana biasanya terjadi pada musim hujan, di mana kelembapan yang tinggi mendukung pertumbuhan jamur ini (Andreanov Ridhovan et al., 2022a). Pengendalian dilakukan dengan pemangkasan daun yang terinfeksi dan aplikasi fungisida yang tepat.

2.3.5.2 Pestalotiopsis

Penyakit Pestalotiopsis disebabkan oleh jamur yang menyerang daun pisang. Gejalanya berupa bercak kecil yang berwarna coklat dengan batas yang jelas, dan seiring waktu bercak ini bisa membesar. Pada kondisi yang lebih parah, penyakit ini dapat menyebabkan kerusakan serius pada daun pisang, mengganggu fotosintesis dan memperburuk pertumbuhan tanaman. Pestalotiopsis sering kali ditemukan pada kondisi cuaca lembap (Andreanov Ridhovan et al., 2022a). Pengendalian umumnya dilakukan dengan fungisida dan perbaikan sistem drainase untuk mengurangi kelembapan berlebih.

2.3.5.3 Sigatoka

Sigatoka merupakan salah satu penyakit yang paling merusak pada tanaman pisang, disebabkan oleh jamur *Mycosphaerella fijiensis*. Penyakit ini ditandai dengan munculnya bercak kuning pada daun yang berkembang menjadi bercak coklat atau hitam. Dengan berjalannya waktu, daun pisang akan mengalami kerusakan parah yang menghambat proses fotosintesis, mengurangi hasil produksi pisang. Ada dua jenis Sigatoka yang umum ditemukan yaitu *Sigatoka kuning* yang disebabkan oleh *M. musicola* dan *Sigatoka hitam* yang disebabkan oleh *M. fijiensis*. Sigatoka hitam lebih berbahaya dan lebih sering ditemukan di daerah tropis. Pengendalian Sigatoka mencakup penggunaan varietas pisang yang tahan terhadap penyakit ini, pemangkasan daun yang terinfeksi, serta aplikasi fungisida secara teratur (Andreanov Ridhovan et al., 2022). Ketiga penyakit ini sangat merugikan tanaman pisang, terutama jika tidak dikendalikan dengan baik. Penanganan yang tepat melalui pencegahan, pengendalian kimia, dan perawatan kebun yang baik sangat penting untuk menjaga kesehatan tanaman pisang dan meminimalkan kerugian hasil.

2.3.6 Evaluasi Performa Model

Untuk mengevaluasi performa model CNN dalam klasifikasi penyakit daun pisang, beberapa metrik evaluasi yang sering digunakan antara lain akurasi, F1 Score, presisi, dan recall (Prasetyo & Utami, 2024). Akurasi mengukur proporsi prediksi yang benar, sedangkan presisi dan recall memberikan gambaran tentang kemampuan model dalam mendeteksi kelas. F1-score adalah kombinasi dari presisi dan recall yang memberikan gambaran keseluruhan tentang keseimbangan antara keduanya.

Evaluasi ini penting untuk memastikan bahwa model tidak hanya akurat, tetapi juga sensitif terhadap kelas yang lebih sedikit ditemukan atau lebih sulit dikenali.

2.3.7 Parameter Setting

Pada bagian ini disajikan pengaturan parameter yang terdiri atas tiga komponen utama, yaitu optimizer, learning rate, dan epoch. Ketiga parameter ini memiliki peran penting dalam proses pelatihan model, karena secara langsung memengaruhi kecepatan konvergensi, kestabilan pelatihan, serta kualitas hasil akhir model. Oleh karena itu, pemilihan dan pengaturan nilai dari masing-masing parameter dilakukan secara cermat untuk memperoleh performa model yang optimal. Penjelasan detail terkait ketiga parameter tersebut sebagai berikut.

2.3.7.1 Optimizer

Optimizer merupakan algoritma yang berfungsi menyesuaikan parameter model seperti bobot dan bias selama proses pelatihan, dengan tujuan meminimalkan loss function, mempercepat konvergensi, serta meningkatkan akurasi model. Setiap jenis optimizer memiliki karakteristik dan keunggulan tersendiri dalam hal kecepatan, stabilitas, serta efisiensi komputasi (Alex David et al., 2024). Optimizer yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Adam (*Adaptive Moment Estimation*)

Adam adalah optimizer adaptif yang secara otomatis menyesuaikan *learning rate* untuk setiap parameter berdasarkan rata-rata gradien (momentum) dan kuadrat gradien sebelumnya. Keunggulannya terletak pada konvergensi yang cepat dan efisien pada dataset besar, serta kemampuannya menangani gradien yang

tidak stabil. Namun, Adam cenderung berisiko *overfitting* dan membutuhkan memori lebih besar, karena harus menyimpan dua momen (kecepatan dan percepatan gradien).

2. AdamW (*Adaptive with Weight Decay*)

AdamW merupakan pengembangan dari Adam yang memperbaiki kelemahannya dalam hal regularisasi. Berbeda dari Adam, AdamW memisahkan proses penyesuaian bobot (*weight decay*) dari pembaruan learning rate. Hal ini membuatnya lebih stabil dan efektif dalam mencegah *overfitting*, terutama saat fine-tuning model deep learning. Meskipun demikian, AdamW sedikit lebih kompleks secara perhitungan dibanding Adam.

3. SGD (*Stochastic Gradient Descent*)

SGD merupakan optimizer klasik yang menggunakan subset kecil data (*mini-batch*) untuk memperbarui bobot setiap iterasi. Dengan menambahkan momentum, SGD dapat mempertahankan arah pembaruan parameter agar lebih stabil. Keunggulannya adalah lebih sederhana, hemat memori, dan menghasilkan model yang lebih generalis, terutama pada dataset kecil. Namun, konvergensinya lebih lambat dan memerlukan penyesuaian learning rate secara manual, sehingga tidak seadaptif Adam atau AdamW.

2.3.7.2 Learning Rate

Learning rate adalah parameter yang mengatur seberapa besar perubahan nilai bobot pada setiap proses pembelajaran model. Nilai ini sangat penting karena menentukan kecepatan model dalam menemukan pola terbaik dari data pelatihan. Jika nilai learning rate terlalu besar, model bisa kesulitan mencapai hasil yang stabil

karena langkah-langkah perubahannya terlalu besar dan dapat melewati nilai optimal (Andreanov Ridhovan et al., 2022; Vakalopoulou et al., 2023). Sebaliknya, jika nilainya terlalu kecil, proses pelatihan akan berlangsung sangat lambat dan bisa jadi tidak mencapai hasil yang maksimal karena terjebak di titik yang kurang optimal. Oleh karena itu, pemilihan nilai learning rate yang sesuai menjadi hal yang penting untuk mempercepat proses pelatihan sekaligus menjaga akurasi model.

2.3.7.3 *Epoch*

Epoch adalah satuan dalam pelatihan model yang menunjukkan berapa kali seluruh data pelatihan digunakan untuk melatih model secara penuh. Dengan setiap *epoch*, model mempelajari data lebih dalam dan memperbarui bobot-bobotnya untuk meningkatkan akurasi prediksi. Jumlah *epoch* yang terlalu sedikit bisa membuat model belum cukup belajar dari data, sedangkan jumlah *epoch* yang terlalu banyak dapat menyebabkan model terlalu menghafal data pelatihan (*overfitting*) dan tidak mampu mengenali data baru dengan baik (Tian et al., 2024). Oleh karena itu, jumlah *epoch* yang digunakan harus dipertimbangkan secara cermat agar model dapat mencapai keseimbangan antara kemampuan belajar dan kemampuan generalisasi.

Pada Tabel 2.2 disajikan penjelasan mengenai perbedaan yang terdapat pada masing-masing parameter setting, meliputi kelebihan dan kelemahan pada setiap konfigurasi yang digunakan.

Tabel 2.2 Perbedaan Masing-masing Parameter *Setting*

Parameter	Kelebihan	Kekurangan
Epoch	Memberikan kesempatan model belajar lebih lama sehingga mampu mencapai konvergensi yang lebih baik dan akurasi lebih tinggi.	Jika jumlah epoch terlalu banyak, model dapat mengalami overfitting dan waktu pelatihan menjadi lebih lama.
Learning rate	Mengatur kecepatan pembaruan bobot selama pelatihan, learning rate yang optimal mempercepat konvergensi dan meningkatkan akurasi.	Nilai terlalu besar menyebabkan model tidak stabil (<i>diverge</i>), sedangkan terlalu kecil membuat proses pelatihan lambat dan akurasi rendah.
Optimizer	Mengoptimalkan proses pembaruan bobot agar model mencapai hasil terbaik. Optimizer seperti Adam, AdamW, dan SGD mampu beradaptasi dengan perubahan gradien secara dinamis.	Setiap optimizer memiliki karakteristik berbeda dengan hasil tidak stabil pada dataset tertentu dan memerlukan penyesuaian parameter tambahan.

Secara umum, ketiga parameter pada Tabel 2.2 tersebut saling berkaitan dalam menentukan performa model. Epoch mengatur durasi pembelajaran, learning rate mengendalikan kecepatan konvergensi, sedangkan optimizer menentukan mekanisme pembaruan bobot. Kombinasi yang tepat dari ketiganya akan menghasilkan kinerja model yang optimal, sementara pengaturan yang tidak seimbang dapat menyebabkan hasil pelatihan kurang maksimal (Prasetyo & Utami, 2024).

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Penelitian ini merupakan jenis penelitian komparatif yang bertujuan untuk menganalisis dan membandingkan performa model VGG16 dan MobileNetV2 dalam klasifikasi penyakit daun pisang. Fokus utama penelitian ini adalah pada peningkatan akurasi dan efisiensi model melalui eksperimen terstruktur dan sistematis. Sifat penelitian ini bersifat kuantitatif, dengan penekanan pada pengumpulan dan analisis data untuk mengevaluasi hasil klasifikasi.

Pendekatan yang digunakan dalam penelitian ini adalah eksperimental, di mana peneliti mengatur kondisi kontrol untuk mengamati efek dari penggunaan dua arsitektur model yang berbeda, yaitu VGG16 dan MobileNetV2 dalam klasifikasi citra. Metodologi penelitian melibatkan teknik supervised learning, masing-masing model dilatih dengan dataset berlabel yang terdiri dari gambar daun pisang sehat dan terinfeksi.

Evaluasi performa model dilakukan dengan menggunakan metrik seperti akurasi, presisi, recall, dan F1-score, serta visualisasi confusion matrix. Hal ini memberikan wawasan komprehensif mengenai kinerja kedua model dalam klasifikasi penyakit daun pisang. Penelitian ini diharapkan dapat memberikan kontribusi yang signifikan terhadap penerapan teknologi deep learning di bidang pertanian, khususnya dalam diagnosis penyakit tanaman.

3.2. Metode Pengumpulan Data

Penelitian ini menggunakan dataset hasil augmentasi sebanyak 1.600 citra daun pisang yang berasal dari dataset asli berjumlah 937 gambar yang diperoleh dari platform Kaggle pada link berikut <https://ungu.in/KaggleNitaCerdas> (Shifat E. Arman, 2024). Dataset awal menunjukkan distribusi tidak seimbang dengan komposisi daun sehat (129 gambar), Cordana (162 gambar), Pestalotiopsis (173 gambar), dan Sigatoka (473 gambar). Ketidakseimbangan distribusi kelas ini berpotensi menyebabkan bias model terhadap kelas mayoritas (Sigatoka) dalam proses klasifikasi.

Untuk mengatasi masalah tersebut, dilakukan teknik augmentasi data yang meliputi *gaussian blur*, *cropping*, *flipping*, dan *rotation*. Proses augmentasi dilakukan secara selektif dengan intensitas berbeda untuk setiap kelas - kelas minor (daun sehat, Cordana, Pestalotiopsis) diaugmentasi hingga mencapai 400 gambar per kelas, sedangkan kelas mayor (Sigatoka) diaugmentasi secara minimal untuk mempertahankan karakteristik aslinya.

Dataset hasil augmentasi kemudian dibagi dengan rasio 70:20:10 menjadi data pelatihan (1.120 gambar), validasi (160 gambar), dan pengujian (320 gambar). Pembagian ini didasarkan pada pertimbangan kebutuhan jumlah sampel yang memadai untuk proses pembelajaran model, optimalisasi generalisasi model melalui validasi, dan efisiensi komputasi. Seluruh proses augmentasi hanya diterapkan pada data pelatihan untuk mencegah kebocoran data (data leakage), sementara data validasi dan pengujian menggunakan gambar asli untuk memastikan evaluasi yang objektif terhadap performa model.

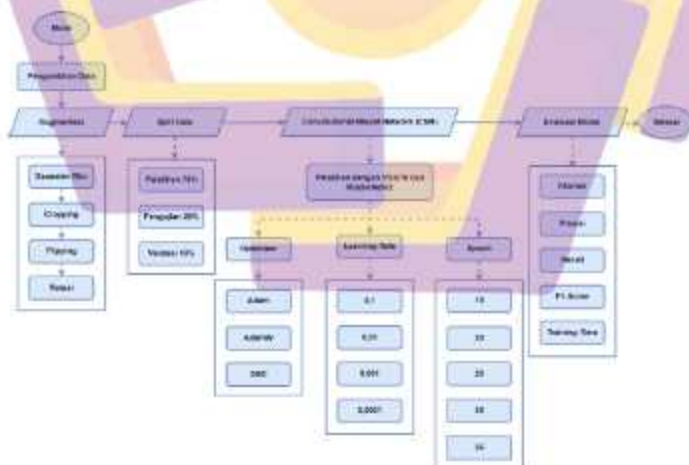
3.3. Metode Analisis Data

Proses analisis dimulai dengan pelatihan kedua model menggunakan data latih yang telah diproses sebelumnya. Selanjutnya, performa model dievaluasi dengan menggunakan metrik kinerja yaitu akurasi, presisi, recall, dan F1-score, untuk mengukur seberapa baik model dapat mengklasifikasikan gambar sesuai dengan kategori penyakitnya.

Confusion matrix digunakan untuk memberikan gambaran lebih mendetail mengenai distribusi prediksi yang benar dan salah pada setiap kelas, sehingga dapat membantu mengidentifikasi kelemahan model dalam mengklasifikasikan kelas tertentu.

3.4. Alur Penelitian

Diagram alur penelitian pada klasifikasi penyakit daun pisang memiliki beberapa tahapan yang saling terintegrasi yaitu pengumpulan data hingga evaluasi metrix. Alur penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

Berdasarkan alur penelitian yang ditampilkan pada Gambar 3.1, berikut disajikan penjelasan secara rinci mengenai tahapan-tahapan dalam alur tersebut.

3.4.1 Pengambilan Data

Pengambilan data dalam penelitian ini dilakukan dengan memanfaatkan citra daun pisang yang diperoleh dari dataset publik "*Banana Leaf Spot Diseases (BananaLSD)*". Dataset tersebut dikembangkan melalui eksperimen yang dilaksanakan oleh Universitas Pertanian Bangabandhu Sheikh Mujibur Rahman (BSMRAU). Seluruh gambar daun pisang dalam dataset ini diambil menggunakan berbagai jenis perangkat telepon pintar, dan diperoleh di lingkungan alami dengan kondisi pencahayaan serta atmosfer yang bervariasi. Variasi kondisi tersebut menghasilkan keragaman visual yang tinggi, sehingga memperkaya kualitas dan cakupan data (Shifat E. Arman, 2024).

Keragaman data ini memiliki nilai strategis dalam meningkatkan ketepatan proses klasifikasi penyakit pada daun pisang. Dataset ini tidak hanya memberikan manfaat bagi para peneliti di bidang patologi tanaman, tetapi juga berguna bagi praktisi pertanian dalam penerapan teknologi di lapangan. Selain itu, keberadaan dataset BananaLSD berkontribusi secara signifikan terhadap pengembangan sistem deteksi dini penyakit bercak daun, yang pada akhirnya dapat membantu petani dalam mengambil keputusan secara tepat dan responsif. Oleh karena itu, pemanfaatan dataset ini turut mendorong terciptanya solusi deteksi penyakit tanaman yang bersifat efisien, terjangkau, dan mendukung keberlanjutan dalam praktik pertanian modern. Dataset yang digunakan dalam penelitian ini disajikan pada Gambar 3.2.



Gambar 3.2 Kelas pada Dataset

3.4.2 Augmentasi Data

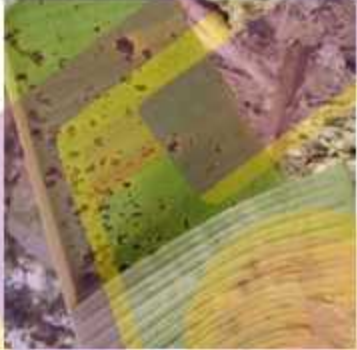

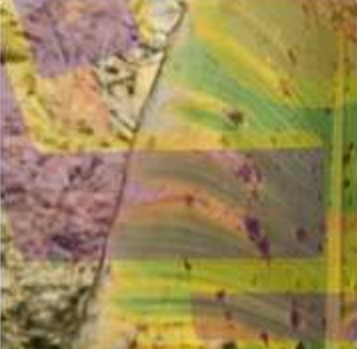
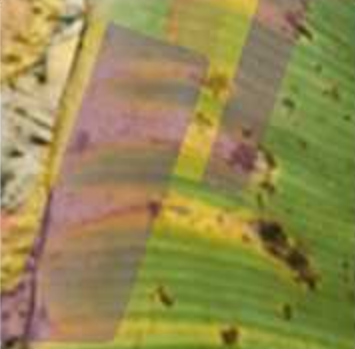
Dalam penelitian ini, proses augmentasi data dilakukan untuk meningkatkan jumlah dan keragaman citra dalam dataset, sehingga model dapat dilatih secara lebih optimal dan mampu mengenali pola secara lebih general. Augmentasi data dilakukan terhadap citra daun pisang, tujuan dari augmentasi ini adalah untuk memperluas variasi visual pada dataset tanpa perlu melakukan pengambilan gambar tambahan di lapangan. Berikut adalah teknik augmentasi yang digunakan dalam penelitian ini meliputi:

3.4.2.1 Gaussian Blur

Teknik augmentasi dengan *gaussian blur* ini menerapkan efek kabur (blur) pada citra menggunakan fungsi distribusi Gaussian. Teknik ini digunakan untuk

mensimulasikan kondisi pencitraan yang tidak fokus atau kabur, seperti yang terjadi pada kamera dengan getaran atau lensa berkabut. Dengan memberikan efek ini, model diharapkan mampu mengenali objek meskipun dalam kondisi gambar yang tidak tajam. Perbandingan citra sebelum dan sesudah diterapkan Gaussian Blur ditampilkan pada Tabel 3.1

Tabel 3.1 Sebelum dan Sesudah *Gaussian Blur*



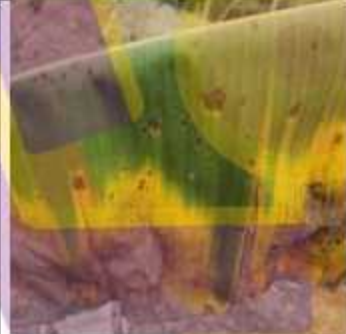
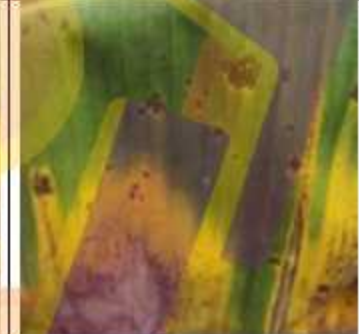
Jenis Penyakit	Gambar Sebelum Gaussian Blur	Gambar Sesudah Gaussian Blur
Pestalotiopsis		
Pestalotiopsis		

3.4.2.2 Cropping

Cropping dilakukan dengan teknik memotong sebagian area dari citra untuk menghasilkan gambar baru. Teknik ini membantu model mengenali bagian tertentu

dari daun yang mungkin menjadi fokus utama dalam mendeteksi gejala penyakit. Cropping juga berguna untuk melatih model dalam mengidentifikasi penyakit meskipun hanya sebagian daun yang terlihat. Perbandingan citra sebelum dan sesudah diterapkan *cropping* ditampilkan pada Tabel 3.2.

Tabel 3.2 Sebelum dan Sesudah *Cropping*


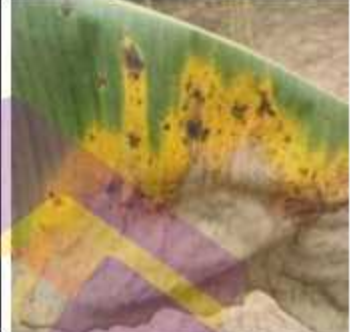
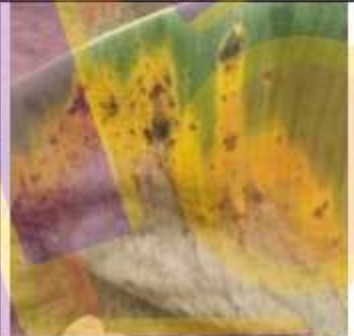

Jenis Penyakit	Gambar Sebelum Cropping	Gambar Sesudah Cropping
Pestalotiopsis		
Pestalotiopsis		

3.4.2.3 Flipping

Flipping atau pencerminan dilakukan dengan proses membalik citra secara horizontal, vertikal, atau keduanya. Teknik ini bertujuan untuk meningkatkan keragaman posisi dan orientasi objek dalam citra, sehingga model tidak hanya

mengenali gejala penyakit dari satu arah pandang saja. Perbandingan citra sebelum dan sesudah diterapkan *flipping* ditampilkan pada Tabel 3.3.



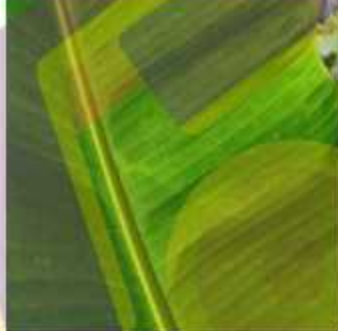

Tabel 3.3 Sebelum dan Sesudah *Flipping*

Jenis Penyakit	Gambar Sebelum <i>flipping</i>	Gambar Sesudah <i>flipping</i>
Pestalotiopsis		
Pestalotiopsis		

3.4.2.4 Rotasi

Rotasi dilakukan dengan proses memutar citra dengan sudut tertentu, misalnya 15° , 30° , atau 45° . Tujuan dari augmentasi ini adalah agar model mampu mengenali pola penyakit meskipun letak daun mengalami pergeseran sudut atau orientasi yang tidak standar. Perbandingan citra sebelum dan sesudah diterapkan rotasi ditampilkan pada Tabel 3.4.

Tabel 3.4 Sebelum dan Sesudah Rotasi

Jenis Penyakit	Gambar Sebelum Rotasi	Gambar Sesudah Rotasi
Healthy		
Healthy		

3.4.3 Split Data

Pada tahap ini, proses *split data* dilakukan dengan tujuan untuk membagi dataset menjadi tiga subset utama. Pembagian data ini sangat penting guna mengukur kemampuan *generalisasi model* dalam memprediksi data baru secara akurat. Dengan demikian, proses *split data* menjadi langkah krusial dalam memastikan bahwa model yang dikembangkan tidak hanya bekerja baik pada data latih, tetapi juga memiliki kinerja yang andal saat diterapkan di dunia nyata. Rincian pembagian jumlah data per kelas dapat dilihat pada Tabel 3.5 berikut.

Tabel 3.5 Pembagian Dataset

No	Kelas	Jumlah Dataset	Augmentasi	Pelatihan (70%)	Validasi (10%)	Pengujian (20%)
1	<i>Healthy</i>	129	400	280	40	80
2	<i>Cordana</i>	162	400	280	40	80
3	<i>Sigatoka</i>	473	400	280	40	80
4	<i>Pestalotiopsis</i>	173	400	280	40	80
	Total	937	1600	1120	160	320

Pada Tabel 3.5, pembagian dataset dilakukan dengan proporsi 70% untuk data pelatihan, 20% untuk data pengujian, serta 10% untuk data validasi. Proporsi ini dipilih supaya model yang dikembangkan dapat dilatih secara optimal, diuji secara menyeluruh, dan divalidasi secara objektif untuk memastikan performa yang akurat dan dapat diandalkan.

Jumlah data yang digunakan untuk pelatihan mencapai 1.120 gambar, sedangkan data pengujian dan validasi masing-masing terdiri atas 320 gambar dan 160 gambar. Jumlah tersebut merupakan hasil dari proses augmentasi terhadap dataset asli, sehingga setiap kelas memiliki total 400 gambar yang seimbang dan merata.

3.4.4 Model VGG16

Model VGG16 merupakan salah satu arsitektur CNN yang terkenal karena kedalamannya (16 lapisan dengan bobot terlatih) dan kesederhanaan desainnya (mengggunakan hanya convolution layer 3x3 dan max pooling). Pada penelitian ini, VGG16 digunakan sebagai arsitektur untuk melakukan pelatihan data citra yang sudah melalui proses augmentasi dan splitting (pelatihan 70%, validasi 10%, pengujian 20%). Konfigurasi VGG16 dapat dilihat pada Tabel 3.6.

Tabel 3.6 Pelatihan VGG16

No	Konfigurasi	Keterangan
	Model	VGG16
1	Optimizer	Adam, AdamW, SGD
2	Learning rate	0.1, 0.01, 0.001, 0.0001
3	Epoch	15, 20, 25, 30, 35
4	Loop	<i>for optimizer in optimizers / for lr in learning_rates</i>

Tabel 3.6 menampilkan konfigurasi *hyperparameter* yang digunakan dalam proses pelatihan model VGG16 pada penelitian ini. Pemilihan model VGG16 didasarkan pada kemampuannya dalam menangani klasifikasi citra secara efektif melalui arsitektur jaringan konvolusional yang mendalam. Untuk proses optimasi, digunakan tiga jenis optimizer, yaitu Adam, AdamW, dan SGD, yang masing-masing memiliki keunggulan tersendiri dalam memperbarui bobot jaringan selama pelatihan. Selain itu, variasi nilai *learning rate* sebesar 0.1, 0.01, 0.001, dan 0.0001 diterapkan untuk mengetahui pengaruh kecepatan pembelajaran terhadap konvergensi model. Jumlah epoch juga divariasikan mulai dari 15 hingga 35 guna mengamati sejauh mana model dapat belajar secara optimal dari data yang tersedia. Kombinasi dari konfigurasi tersebut dirancang untuk mengevaluasi performa model secara menyeluruh dan menemukan pengaturan terbaik yang menghasilkan akurasi tinggi serta kestabilan dalam proses pelatihan.

Sebagai mekanisme otomatisasi, penelitian ini menggunakan *loop* untuk mengiterasi setiap kombinasi *hyperparameter* tersebut. Dengan menggunakan *loop* ini, proses pelatihan dapat dijalankan berulang kali untuk setiap pasangan optimizer, learning rate, dan epoch tanpa harus menulis ulang kode pelatihan secara manual.

3.4.5 Model MobileNetV2

Model MobileNetV2 adalah arsitektur CNN yang lebih ringan dan efisien dibandingkan VGG16, dirancang khusus untuk perangkat dengan sumber daya terbatas. Model ini memanfaatkan digunakan untuk mengurangi jumlah parameter dan komputasi. Konfigurasi VGG16 dapat dilihat pada Tabel 3.7.

Tabel 3.7 Pelatihan MobileNetV2

No	Konfigurasi	Keterangan
	Model	MobileNetV2
1	Optimizer	Adam, AdamW, SGD
2	Learning rate	0.1, 0.01, 0.001, 0.0001
3	Epoch	15, 20, 25, 30, 35
4	Loop	<i>for optimizer in optimizers / for lr in learning_rates</i>

Tabel 3.7 menyajikan konfigurasi pelatihan yang digunakan untuk model MobileNetV2 dalam penelitian ini. Untuk proses optimasi model, digunakan tiga jenis optimizer, yaitu Adam, AdamW, dan SGD, yang masing-masing memiliki metode perhitungan gradien yang berbeda untuk mempercepat konvergensi dan meningkatkan akurasi. Variasi *learning rate* sebesar 0.1, 0.01, 0.001, dan 0.0001 diterapkan untuk mengamati pengaruhnya terhadap proses pembelajaran model. Selain itu, jumlah epoch divariasikan dari 15 hingga 35 guna mengevaluasi performa model dalam berbagai durasi pelatihan dan menghindari risiko *overfitting* maupun *underfitting*. Konfigurasi ini dirancang secara sistematis untuk mengidentifikasi kombinasi *hyperparameter* terbaik dalam mengoptimalkan kinerja MobileNetV2 pada tugas klasifikasi citra daun pisang (Prasetyo & Utami, 2024; Shifat E. Arman, 2024).

Sebagai langkah otomatisasi, loop digunakan untuk mengiterasi setiap kombinasi hyperparameter, melalui perintah *for optimizer in optimizers* dan *for lr*

in *learning_rates*. Dengan menggunakan fungsi ini, seluruh kombinasi *optimizer*, *learning rate*, dan *epoch* dapat dijalankan secara sistematis tanpa penulisan kode berulang, sehingga mempermudah proses evaluasi dan mempercepat pencarian konfigurasi terbaik.

3.4.6 Evaluasi Model

Untuk mendapatkan hasil akhir pada penelitian ini menggunakan akurasi, presisi, recall dan f1-score.

3.4.6.1 Akurasi

Accuracy (akurasi) merupakan metrik yang menunjukkan seberapa sering model membuat prediksi yang benar dibandingkan dengan seluruh jumlah data (Prasetyo & Utami, 2024). Rumusnya adalah sebagai berikut:

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

TP adalah True Positive, TN adalah True Negative, FP adalah False Positive, dan FN adalah False Negative. Akurasi sangat berguna ketika data seimbang, artinya jumlah data dari setiap kelas hampir sama. Namun, jika data tidak seimbang, akurasi bisa menyesatkan karena model bisa terlihat baik hanya dengan memprediksi kelas mayoritas (Prasetyo & Utami, 2024).

3.4.6.2 Presisi

Presisi adalah metrik yang mengukur seberapa tepat model dalam memprediksi kelas positif. Presisi menjawab pertanyaan dari semua prediksi positif yang dibuat oleh model (Prasetyo & Utami, 2024). Rumusnya adalah sebagai berikut:

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

Presisi menjadi penting dalam situasi di mana kesalahan positif palsu (*false positive*) sangat merugikan.

3.4.6.3 Recall

Recall atau sensitivitas menunjukkan seberapa baik model dapat menemukan semua kasus positif yang sebenarnya ada di dalam data (Prasetyo & Utami, 2024).

Rumus recall adalah sebagai berikut:

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

Recall menjadi penting ketika kesalahan negatif palsu (*false negative*) harus diminimalkan.

3.4.6.4 F1-Score

F1-Score adalah metrik gabungan dari *precision* dan *recall* yang mempertimbangkan keduanya secara seimbang (Prasetyo & Utami, 2024). F1-score dihitung dengan rumus sebagai berikut:

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

F1-score sangat berguna ketika ingin keseimbangan antara *precision* dan *recall*, terutama dalam kasus data tidak seimbang. Skor ini memberikan gambaran yang lebih adil terhadap performa model ketika tidak bisa hanya mengandalkan akurasi.

3.4.6.5 Training Time

Training time merupakan durasi waktu yang dibutuhkan oleh sebuah model *deep learning* untuk menyelesaikan proses pelatihan terhadap seluruh dataset. Waktu ini mencakup seluruh proses iteratif dalam mengoptimalkan bobot dan bias model melalui serangkaian *epoch*, yang masing-masing terdiri atas sejumlah *batch*.

Lama atau singkatnya *training time* dipengaruhi oleh berbagai faktor, seperti ukuran dataset, arsitektur model yang digunakan, jumlah *epoch*, jenis *optimizer*, serta spesifikasi perangkat keras seperti CPU atau GPU. Dalam penelitian ini, analisis terhadap *training time* menjadi hal penting untuk mengevaluasi efisiensi model, khususnya saat membandingkan dua arsitektur berbeda seperti VGG16 dan MobileNetV2 (Shi, 2024). Model yang memiliki waktu pelatihan lebih singkat cenderung lebih efisien secara komputasi, namun harus tetap diperhatikan bahwa efisiensi waktu harus seimbang dengan performa akurasi yang dihasilkan.



BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Pelatihan Menggunakan VGG16

Berikut ini merupakan penjelasan detail terkait hasil pelatihan menggunakan VGG16.

4.1.1 Hasil akurasi

Berikut ini merupakan hasil evaluasi akurasi model VGG16 berdasarkan variasi jumlah epoch selama proses pelatihan yang dapat dilihat pada Gambar 4.1.



Gambar 4.1 Hasil Akurasi VGG16

Gambar 4.1 menunjukkan bahwa akurasi model VGG16 berada pada kisaran tinggi dan relatif stabil di atas 96%, dengan nilai tertinggi sebesar 98,12% yang dicapai pada epoch 15, 30, dan 35. Hal ini menunjukkan bahwa model telah mencapai konvergensi lebih awal, sehingga penambahan epoch setelah titik tersebut tidak lagi memberikan peningkatan akurasi yang signifikan.

Dengan demikian, gambar ini memberikan pengetahuan bahwa model VGG16 memiliki kemampuan generalisasi yang baik dan efisien dalam proses

pelatihan, di mana peningkatan jumlah epoch tidak selalu berbanding lurus dengan peningkatan akurasi.

4.1.2 Hasil Presisi

Berikut ini merupakan hasil evaluasi presisi model VGG16 berdasarkan variasi jumlah epoch selama proses pelatihan yang dapat dilihat pada Gambar 4.2.



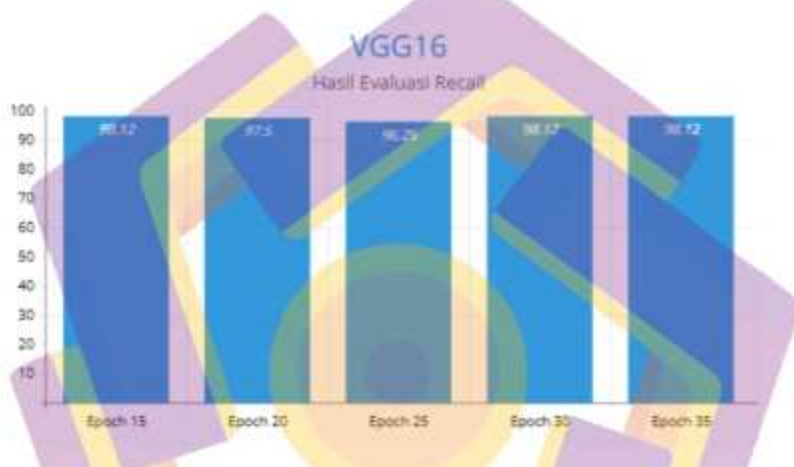
Gambar 4.2 Hasil Presisi VGG16

Gambar 4.2 menunjukkan hasil evaluasi presisi dari model VGG16 pada berbagai epoch, yaitu epoch ke-15, 20, 25, 30, dan 35, presisi tertinggi dicapai pada epoch ke-15 dengan nilai 98,17%, diikuti oleh epoch ke-30 dengan nilai 98,16% dan epoch ke-35 dengan nilai 98,12%. Sementara itu, presisi terendah terjadi pada epoch ke-25 dengan nilai 96,37%. Hasil ini menunjukkan bahwa model VGG16 relatif stabil dalam menghasilkan nilai presisi yang tinggi pada setiap interval epoch yang diuji, dengan fluktuasi yang tidak terlalu signifikan. Nilai presisi yang konsisten di atas 96% mengindikasikan bahwa model mampu mengklasifikasikan data dengan tingkat ketepatan yang sangat baik, meskipun terdapat sedikit

penurunan presisi pada epoch tertentu. Evaluasi ini penting untuk menentukan epoch optimal yang dapat digunakan dalam proses pelatihan model guna mencapai performa terbaik.

4.1.3 Hasil Recall

Berikut ini merupakan hasil evaluasi recall model VGG16 berdasarkan variasi jumlah epoch selama proses pelatihan yang dapat dilihat pada Gambar 4.3.



Gambar 4.3 Hasil Recall VGG16

Gambar 4.3 menyajikan hasil evaluasi metrik recall dari VGG16 pada lima epoch yang berbeda. Hasil recall tertinggi tercatat pada epoch ke-15, 30, dan 35 dengan nilai yang sama, yaitu 98,12%, menunjukkan bahwa pada epoch-epoch ini model mampu mengidentifikasi sebagian besar sampel positif secara benar. Sementara itu, nilai recall terendah terjadi pada epoch ke-25 dengan 96,25%, sedangkan pada epoch ke-20 tercatat 97,5%.

Secara keseluruhan, hasil ini memperlihatkan bahwa performa recall dari model VGG16 cukup stabil dan tinggi, dengan nilai selalu berada di atas 96%. Hal

ini menunjukkan bahwa model memiliki kemampuan yang baik dalam mendeteksi seluruh data positif tanpa terlalu banyak yang terlewat (*false negatives*).

4.1.4 Hasil F1-Score

Berikut ini merupakan hasil evaluasi F1-score model VGG16 berdasarkan variasi jumlah epoch selama proses pelatihan yang dapat dilihat pada Gambar 4.4.



Gambar 4.4 Hasil F1-Score VGG16

Gambar 4.4 merupakan hasil evaluasi F1-Score dari VGG16 pada beberapa epoch, yaitu epoch ke-15, 20, 25, 30, dan 35. Metrik F1-Score merupakan gabungan dari presisi dan recall, yang memberikan gambaran seimbang tentang performa model, terutama pada kasus data yang tidak seimbang.

Berdasarkan hasil eksperimen yang dilakukan, F1-Score tertinggi diperoleh pada epoch ke-30 dengan nilai 98,13%, diikuti sangat dekat oleh epoch ke-15 dan ke-35 yang sama-sama mencatat nilai 98,12%. F1-Score terendah terjadi pada epoch ke-25, yaitu 96,28%, sementara pada epoch ke-20 tercatat 97,51%.

Nilai F1-Score yang relatif tinggi dan konsisten menunjukkan bahwa model VGG16 memiliki kinerja yang baik dalam menyeimbangkan antara kemampuan mengidentifikasi data positif secara benar (*recall*) dan menghindari kesalahan klasifikasi positif yang salah (*presisi*). Meskipun terdapat sedikit fluktuasi antar *epoch*, perbedaannya sangat kecil dan cenderung stabil, yang mengindikasikan bahwa VGG16 dapat diandalkan untuk klasifikasi dengan performa yang optimal pada hampir semua *epoch* yang diuji.

4.1.5 Hasil *Training Time*

Berikut ini merupakan hasil evaluasi *training time* model VGG16 berdasarkan variasi jumlah *epoch* selama proses pelatihan yang dapat dilihat pada Gambar 4.5.



Gambar 4.5 *Training Time* VGG16

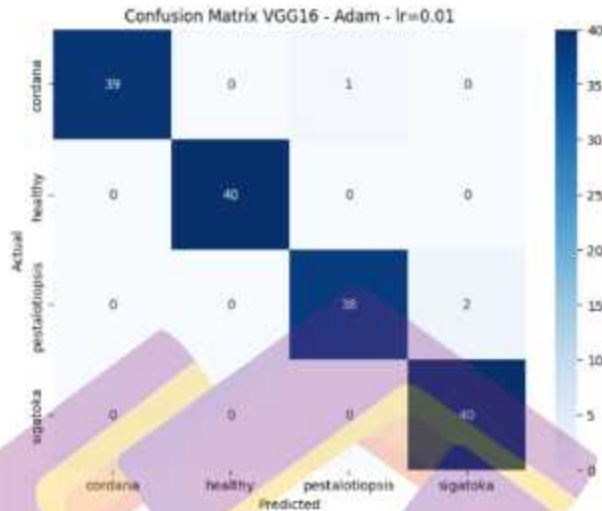
Gambar 4.5 menunjukkan waktu pelatihan (*training time*) model VGG16 pada berbagai jumlah *epoch*, yaitu 15, 20, 25, 30, dan 35. Secara umum, hasil menunjukkan adanya peningkatan waktu pelatihan seiring dengan bertambahnya jumlah *epoch*. Pada *epoch* ke-15, waktu pelatihan tercatat sebesar 79,92 detik. Nilai

ini meningkat menjadi 106,79 detik pada epoch ke-20, dan terus bertambah menjadi 130,38 detik pada epoch ke-25. Peningkatan signifikan terjadi pada epoch ke-30 dengan waktu pelatihan sebesar 262,54 detik, dan mencapai puncaknya pada epoch ke-35 dengan waktu pelatihan 275,26 detik.

Kenaikan waktu pelatihan ini mencerminkan beban komputasi yang semakin besar seiring bertambahnya epoch, di mana model melakukan proses *forward* dan secara berulang dalam jumlah yang lebih banyak. Hal ini sesuai dengan karakteristik umum dari pelatihan model *deep learning*, di mana kompleksitas dan kebutuhan sumber daya meningkat proporsional terhadap jumlah epoch. Meskipun waktu pelatihan meningkat secara signifikan, terutama setelah epoch ke-25, pertimbangan terhadap peningkatan akurasi atau performa model menjadi aspek penting dalam menentukan jumlah epoch yang optimal, agar tidak terjadi *overfitting* maupun inefisiensi dalam penggunaan sumber daya komputasi.

4.1.6 Hasil Confusion Matriks dan Loss

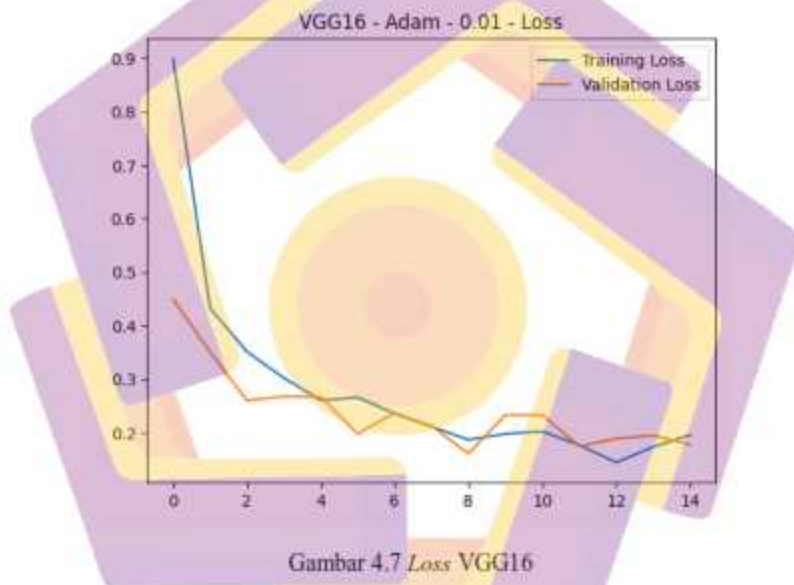
Berikut ini merupakan hasil *confusion matrix* dan *loss* dari VGG16 berdasarkan variasi jumlah epoch selama proses pelatihan yang dapat dilihat pada Gambar 4.5 dan Gambar 4.6. *Confusion matrix* ini penting untuk memberikan gambaran performa klasifikasi model terhadap masing-masing kelas, sedangkan grafik *loss* digunakan untuk mengevaluasi efektivitas proses pelatihan dan kestabilan generalisasi model.



Gambar 4.6 *Confusion Matriks* VGG16

Gambar 4.6 merupakan *confusion matrix* dari hasil klasifikasi menggunakan model VGG16 dengan *optimizer* Adam dan *learning rate* sebesar 0.01. *Confusion matrix* ini digunakan untuk mengevaluasi performa model dalam mengklasifikasikan empat kelas, yaitu cordana, healthy, pestalotiopsis, dan sigatoka. Berdasarkan matriks tersebut, dapat diketahui bahwa kelas cordana diprediksi dengan sangat baik, di mana 39 dari 40 data diklasifikasikan dengan benar, dan hanya 1 data yang salah diklasifikasikan ke kelas pestalotiopsis. Sementara itu, kelas healthy berhasil diklasifikasikan dengan sempurna, yakni seluruh 40 data dikenali dengan benar tanpa adanya kesalahan. Untuk kelas pestalotiopsis, performa model sedikit lebih rendah dibandingkan dua kelas sebelumnya, dengan 38 data diklasifikasikan dengan benar dan 2 data mengalami kesalahan klasifikasi ke kelas sigatoka. Adapun kelas sigatoka juga menunjukkan

hasil klasifikasi yang sangat baik, dengan seluruh 40 data diklasifikasikan dengan benar. Secara keseluruhan, model menunjukkan akurasi yang sangat tinggi, yang ditandai dengan dominasi warna biru gelap pada bagian diagonal confusion matrix (menunjukkan prediksi benar) dan sangat sedikit kesalahan klasifikasi pada bagian off-diagonal. Hal ini menunjukkan bahwa model VGG16 dengan konfigurasi yang digunakan memiliki kemampuan yang sangat baik dalam mengenali masing-masing kelas citra daun pisang pada dataset yang digunakan.



Gambar 4.7 Loss VGG16

Gambar 4.7 merupakan grafik *loss function* dari proses pelatihan model VGG16 yang menggunakan *optimizer* Adam dengan *learning rate* sebesar 0.01. Grafik ini menampilkan perbandingan antara *training loss* (garis biru) dan *validation loss* (garis orange) selama proses pelatihan berlangsung selama 14 epoch.

Pada awal pelatihan, nilai loss untuk data pelatihan cukup tinggi (mendekati 0.9), namun mengalami penurunan dalam beberapa epoch pertama. Hal ini menunjukkan bahwa model dengan cepat belajar mengenali pola dari data pelatihan. Nilai training loss terus menurun di bawah nilai 0.2 pada akhir epoch.

Sementara itu, *validation loss* juga menunjukkan tren penurunan yang stabil dan konsisten dengan *training loss*, tanpa terjadi lonjakan drastis atau *overfitting* yang signifikan. *Validation loss* berada dalam kisaran yang serupa dengan *training loss*, bahkan lebih rendah pada beberapa epoch, yang merupakan indikasi bahwa model tidak hanya belajar dari data pelatihan, tetapi juga mampu melakukan generalisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

Secara keseluruhan, grafik ini menunjukkan bahwa model VGG16 dengan konfigurasi tersebut berhasil dilatih secara efektif, dengan proses pembelajaran yang stabil dan kemampuan generalisasi yang baik, sebagaimana ditunjukkan oleh *tren loss* yang menurun dan konsisten antara data pelatihan dan data validasi.

4.2. Pelatihan Menggunakan MobileNetV2

Berikut ini merupakan penjelasan detail terkait hasil penelitian menggunakan MobileNetV2.

4.2.1 Hasil Akurasi

Berikut ini merupakan hasil evaluasi akurasi model MobileNetV2 berdasarkan variasi jumlah epoch selama proses pelatihan yang dapat dilihat pada Gambar 4.8.



Gambar 4.8 Hasil Akurasi MobileNetV2

Gambar 4.8 menunjukkan bahwa akurasi MobileNetV2 cenderung stabil di 96,25% hingga 96,88%, dengan peningkatan kecil pada epoch ke-30. Hal ini menunjukkan bahwa model sudah mencapai konvergensi lebih awal, dan penambahan epoch berikutnya tidak memberikan peningkatan yang berarti terhadap akurasi.

Temuan ini mengindikasikan bahwa MobileNetV2 mampu mencapai performa yang konsisten dengan jumlah epoch relatif sedikit, menandakan efisiensi dalam proses pelatihan. Hal ini juga memberikan pengetahuan bahwa peningkatan jumlah epoch tidak selalu berbanding lurus dengan peningkatan akurasi, karena model sudah stabil dan tidak memerlukan pelatihan lebih lama untuk mencapai hasil optimal.

4.2.2 Hasil Prestasi

Berikut ini merupakan hasil evaluasi presisi model MobileNetV2 berdasarkan variasi jumlah epoch selama proses pelatihan yang dapat dilihat pada Gambar 4.9.



Gambar 4.9 Hasil Presisi MobileNetV2

Gambar 4.9 menampilkan hasil evaluasi presisi dari model MobileNetV2 pada berbagai epoch. Nilai presisi tertinggi diperoleh pada epoch ke-30 dengan nilai 96,88%, mengungguli nilai pada epoch lainnya yang berada dalam rentang 96,25%–96,33%. Hal ini menunjukkan bahwa model memiliki kemampuan paling optimal dalam mengklasifikasikan data positif secara benar pada epoch ke-30. Dengan demikian, epoch ke-30 dapat dianggap sebagai konfigurasi paling efektif dalam aspek presisi untuk model MobileNetV2.

4.2.3 Hasil Recall

Berikut ini merupakan hasil evaluasi *recall* model MobileNetV2 berdasarkan variasi jumlah epoch selama proses pelatihan yang dapat dilihat pada Gambar 4.10.



Gambar 4.10 Hasil Recall MobileNetV2

Gambar 4.10 menunjukkan hasil evaluasi nilai recall dari model MobileNetV2 pada berbagai jumlah epoch. Nilai recall tertinggi dicapai pada epoch 30 dengan angka 96,89%, sementara pada epoch 15, 20, 25, dan 35 nilai recall tercatat konstan sebesar 96,25%. Hal ini menunjukkan bahwa peningkatan jumlah epoch hingga 30 memberikan dampak positif terhadap kemampuan model dalam mengidentifikasi seluruh kelas positif secara benar. Namun, setelah melewati epoch 30, tidak terjadi peningkatan yang signifikan, menandakan potensi stabilisasi performa model.

4.2.4 Hasil F1-Score

Berikut ini merupakan hasil evaluasi F1-Score model MobileNetV2 berdasarkan variasi jumlah epoch selama proses pelatihan yang dapat dilihat pada Gambar 4.11.



Gambar 4.11 Hasil F1-Score MobileNetV2

Gambar 4.11 menyajikan hasil evaluasi F1-Score dari model MobileNetV2 dengan variasi jumlah epoch. Nilai F1-Score tertinggi dicapai pada epoch ke-30 sebesar 96,87%, menunjukkan keseimbangan optimal antara precision dan recall pada konfigurasi tersebut. Pada epoch lainnya, nilai F1-Score berada dalam rentang 96,22% hingga 96,24%, dengan perbedaan yang sangat tipis, mencerminkan performa model yang stabil dan andal. Hasil ini menegaskan bahwa penggunaan 30 epoch merupakan konfigurasi paling efektif untuk memperoleh performa klasifikasi terbaik dalam hal keseimbangan deteksi positif dan penghindaran kesalahan prediksi.

4.2.5 Hasil *Training Time*

Berikut ini merupakan hasil evaluasi *training time* model MobileNetV2 berdasarkan variasi jumlah epoch selama proses pelatihan yang dapat dilihat pada Gambar 4.12.



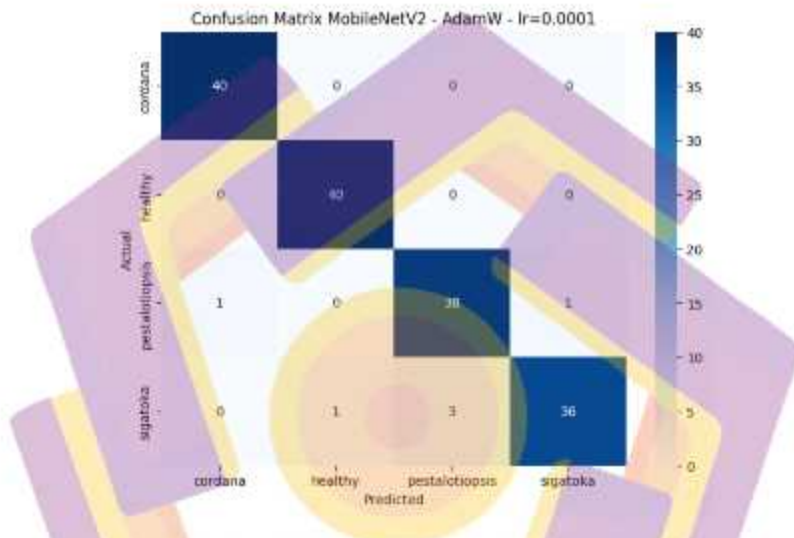
Gambar 4.12 *Training Time* MobileNetV2

Gambar 4.12 menunjukkan durasi waktu pelatihan (*training time*) model MobileNetV2 pada berbagai jumlah epoch. Waktu pelatihan terlama terjadi pada epoch ke-25 dengan durasi 141,77 detik, sedangkan waktu pelatihan tercepat terdapat pada epoch ke-35, yaitu 92,5 detik. Secara umum, peningkatan jumlah epoch tidak selalu berkorelasi langsung dengan peningkatan waktu pelatihan, yang kemungkinan dipengaruhi oleh kombinasi optimizer, learning rate, serta efisiensi komputasi selama proses pelatihan. Hasil ini penting sebagai pertimbangan dalam memilih konfigurasi model yang optimal antara akurasi dan efisiensi waktu.

4.2.6 Hasil *Confusion Matriks* dan *Loss*

Hasil evaluasi performa model MobileNetV2 berdasarkan variasi jumlah epoch selama proses pelatihan disajikan melalui *confusion matrix* dan *loss curve*, masing-masing ditampilkan pada Gambar 4.13 dan Gambar 4.14. *Confusion matrix* tersebut menunjukkan tingkat akurasi klasifikasi terhadap masing-masing kelas

target, sedangkan *loss curve* menggambarkan perkembangan nilai *training loss* dan *validation loss* pada setiap epoch. Kedua visualisasi tersebut memberikan gambaran menyeluruh mengenai efektivitas proses pelatihan model serta kemampuannya dalam melakukan generalisasi terhadap data yang belum pernah dilihat sebelumnya.



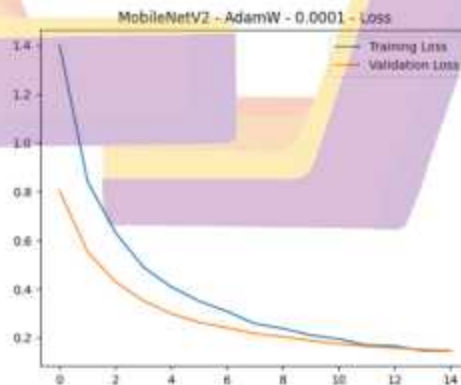
Gambar 4.13 Confusion Matrix MobileNetV2

Gambar 4.13 merupakan representasi confusion matrix yang digunakan untuk mengevaluasi kinerja model MobileNetV2 dalam melakukan klasifikasi citra penyakit daun pisang. Model ini dilatih menggunakan algoritma optimisasi AdamW dengan nilai *learning rate* sebesar 0.0001. Proses klasifikasi melibatkan empat kelas, yaitu cordana, healthy, pestalotiopsis, dan sigatoka dengan masing-masing kelas terdiri dari 40 citra uji.

Pada hasil *confusion matrix*, diketahui bahwa model mampu mengklasifikasikan kelas cordana dan healthy secara sempurna, dengan masing-

masing memperoleh 40 prediksi yang benar dari total 40 data uji (tingkat akurasi 100%). Untuk kelas pestalotiopsis, model mengklasifikasikan 38 data secara tepat, sementara dua sisanya mengalami kesalahan klasifikasi, masing-masing satu data diprediksi sebagai cordana dan satu data sebagai sigatoka. Adapun pada kelas sigatoka, model berhasil mengklasifikasikan 36 data secara benar, dengan tiga data salah diklasifikasikan sebagai pestalotiopsis dan satu data sebagai healthy.

Secara umum, model menunjukkan performa klasifikasi yang sangat baik, terutama dalam membedakan citra daun yang sehat dan yang terinfeksi penyakit cordana. Kesalahan klasifikasi yang terjadi mayoritas ditemukan antara kelas pestalotiopsis dan sigatoka, yang dimungkinkan disebabkan oleh kemiripan ciri visual antara kedua kelas tersebut. Temuan ini dapat menjadi bahan pertimbangan dalam pengembangan lebih lanjut, terutama dalam pemilihan atau penguatan fitur visual serta penerapan teknik augmentasi data untuk meningkatkan akurasi klasifikasi pada kelas-kelas yang memiliki karakteristik serupa. Hasil *training time* dapat dilihat pada Gambar 4.14.



Gambar 4.14 Hasil *Training Time* MobileNetV2

Gambar 4.14 merupakan *grafik loss curve* yang menggambarkan nilai *training loss* dan *validation loss* dari model MobileNetV2 yang dilatih menggunakan algoritma optimisasi AdamW dengan nilai *learning rate* sebesar 0.0001 selama 15 epoch. Hasil tersebut berfungsi untuk mengevaluasi proses selama pelatihan dan validasi, serta mengidentifikasi potensi masalah seperti *overfitting* atau *underfitting*.

Pada *training loss* dan *validation loss* mengalami penurunan yang signifikan seiring bertambahnya epoch, yang menunjukkan bahwa model berhasil mempelajari pola data dengan baik dari waktu ke waktu. Pada awal pelatihan, *training loss* dimulai di atas angka 1.4, kemudian secara konsisten menurun hingga mendekati nilai 0.2 pada epoch ke-14. Sementara itu, *validation loss* juga menunjukkan tren penurunan yang serupa, bahkan berada sedikit di bawah *training loss* pada sebagian besar epoch, yang mengindikasikan bahwa model tidak mengalami *overfitting* dan memiliki kemampuan generalisasi yang baik terhadap data yang tidak dilatih.

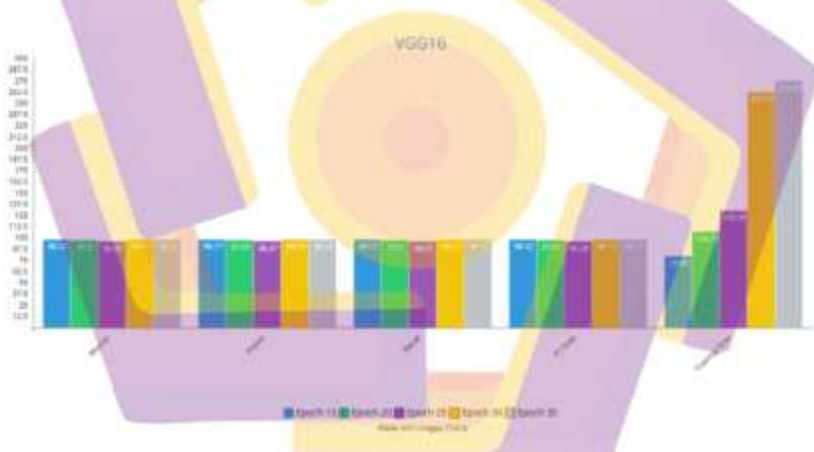
Stabilitas dan konvergensi dari kedua kurva loss ini menunjukkan bahwa parameter pelatihan, termasuk arsitektur model, algoritma optimisasi, serta nilai *learning rate* yang digunakan, telah dipilih dengan tepat. Hasil ini memberikan indikasi bahwa model MobileNetV2 yang digunakan memiliki performa yang optimal dalam menyelesaikan tugas klasifikasi citra pada dataset yang digunakan, baik pada data pelatihan maupun data validasi.

4.3. Evaluasi Model

Hasil evaluasi performa VGG16 dan MobileNetV2 disajikan berdasarkan metrik evaluasi yang meliputi akurasi, presisi, recall, F1-score, serta training time. Evaluasi ini bertujuan untuk menentukan konfigurasi terbaik dari model dalam mengklasifikasi penyakit daun pisang. Setiap model diuji menggunakan kombinasi jumlah epoch, jenis optimizer, dan nilai learning rate. Rincian hasil evaluasi disajikan pada penjelasan berikut.

4.3.1 Hasil Evaluasi VGG16

Dibawah ini merupakan hasil evaluasi VGG16 yang dapat dilihat pada Gambar 4.15.



Gambar 4.15 Hasil Evaluasi VGG16

Gambar 4.15 menunjukkan hasil evaluasi performa model VGG16 berdasarkan metrik akurasi, presisi, recall, F1-score, dan waktu pelatihan pada setiap epoch. Performa model tergolong sangat tinggi dan stabil, dengan semua metrik evaluasi (akurasi, presisi, recall, dan F1-score) berada di 97% hingga 98%,

yang menunjukkan bahwa model mampu mengenali pola data dengan sangat baik secara konsisten pada setiap epoch.

Namun, terdapat peningkatan signifikan pada waktu pelatihan seiring bertambahnya epoch, dengan waktu terlama tercatat pada epoch ke-35 (275,26 detik). Hal ini menandakan bahwa meskipun akurasi meningkat secara marginal, maka komputasi juga semakin besar, sehingga perlu pertimbangan efisiensi dalam menentukan jumlah epoch optimal. Dapat disimpulkan bahwa VGG16 sangat akurat tetapi membutuhkan sumber daya yang jauh lebih besar dibanding model lain seperti MobileNetV2.

Tabel 3.5 diperlukan untuk menampilkan perbandingan skenario terbaik dari berbagai kombinasi parameter pelatihan (jumlah epoch, optimizer, dan learning rate) terhadap performa model VGG16. Evaluasi ini bertujuan untuk mengidentifikasi konfigurasi paling optimal yang menghasilkan performa tertinggi dengan efisiensi waktu pelatihan yang seimbang.

4.3.2 Skenario Terbaik VGG16

Skenario terbaik untuk model VGG16 diperoleh pada saat model dilatih menggunakan *optimizer*, *learning rate* dan *epoch* jumlah epoch. Hasil skenario terbaik pada VGG16 dapat dilihat pada Tabel 4.1.

Tabel 4.1 Skenario Terbaik Pada VGG16

Jumlah Epoch	Optimizer	Learning Rate	Confusion Matrix & Training Time	Hasil
Epoch 15	Adam	0.01	Akurasi	98,12%
			Presisi	98,17%
			Recall	98,12%
			F1-Score	98,12%
			Training Time	79,92 seconds
Epoch 20	AdamW	0.01	Akurasi	97,5%
			Presisi	97,57%

			Recall	97,5%
			F1-Score	97,51%
			Training Time	106,79 seconds
Epoch 25	Adam	0.001	Akurasi	96,25%
			Presisi	96,37%
			Recall	96,25%
			F1-Score	96,28%
			Training Time	130,38 seconds
Epoch 30	Adam	0.01	Akurasi	98,12%
			Presisi	98,16%
			Recall	98,12%
			F1-Score	98,13%
			Training Time	262,54 seconds
Epoch 35	AdamW	0.001	Akurasi	98,12%
			Presisi	98,12%
			Recall	98,12%
			F1-Score	98,12%
			Training Time	275,26 seconds

Berdasarkan hasil pada Tabel 4.1, skenario terbaik dicapai pada epoch ke-30 dengan optimizer Adam dan learning rate 0.01, menghasilkan akurasi tertinggi sebesar 98.12% serta nilai presisi, recall, dan F1-score yang juga tinggi (>98%). Meskipun demikian, skenario ini memiliki waktu pelatihan tertinggi (262.54 detik), sehingga menunjukkan adanya trade-off antara akurasi maksimal dan efisiensi waktu.

Sementara itu, skenario dengan epoch ke-15 dan optimizer Adam menunjukkan hasil yang cukup tinggi (98.12%) namun dengan waktu pelatihan paling singkat (79.92 detik), menjadikannya pilihan efisien jika waktu pelatihan menjadi pertimbangan utama. Dengan demikian, tabel ini penting dalam menentukan parameter pelatihan terbaik berdasarkan kebutuhan spesifik: akurasi maksimum atau efisiensi komputasi.

4.3.3 Faktor yang Mempengaruhi Skenario Terbaik pada VGG16

Berikut ini merupakan faktor-faktor yang memengaruhi terbentuknya skenario terbaik pada proses pengujian model.

Tabel 4.2 Hasil Pengujian Optimizer yang Berbeda dan LR yang Sama

Optimizer	Learning Rate	Akurasi
Adam	0,1	50%
AdamW	0,1	49,38%
SGD	0,1	25%
Adam	0,01	98,12%
AdamW	0,01	97,50%
SGD	0,01	85%
Adam	0,001	97,50%
AdamW	0,001	96,88%
SGD	0,001	82,50%
Adam	0,0001	81,87%
AdamW	0,0001	82,50%
SGD	0,0001	73,12%

Hasil pengujian pada Tabel 4.2 memperlihatkan bahwa pemilihan *learning rate* yang tepat memiliki pengaruh dominan terhadap kinerja model dibandingkan dengan perbedaan jenis optimizer. Meskipun Adam dan AdamW menunjukkan performa yang sangat baik dan stabil di berbagai *learning rate*, pencapaian akurasi optimal tetap bergantung pada nilai *learning rate* yang digunakan selama proses pelatihan.

Visualisasi hasil pengujian kombinasi *optimizer* dan *learning rate* pada model VGG16 disajikan pada Gambar 4.16, yang memperlihatkan perbandingan tingkat akurasi antar optimizer berdasarkan variasi nilai *learning rate*.



Gambar 4.16 Perbandingan Akurasi VGG16 (Optimizer dan LR)

Gambar 4.16 menunjukkan perbandingan akurasi model VGG16 berdasarkan variasi optimizer dan learning rate. Pada sumbu horizontal (X) ditampilkan nilai learning rate, sedangkan sumbu vertikal (Y) menunjukkan nilai akurasi dalam persen (%). Tiga jenis optimizer yang digunakan adalah Adam, AdamW, dan SGD, yang masing-masing diwakili oleh garis berwarna biru, oranye, dan hijau.

Dari grafik terdapat pola perubahan akurasi yang dipengaruhi oleh learning rate. Baik pada optimizer Adam maupun AdamW, akurasi tertinggi dicapai saat learning rate berada di kisaran 0.01, dengan nilai mendekati 98%. Ketika learning rate terlalu tinggi (0.1) atau terlalu rendah (0.0001), akurasi justru menurun cukup signifikan. Hal ini menunjukkan bahwa *learning rate* sangat penting untuk mencapai kinerja terbaik model.

Sementara itu, jika dibandingkan antar optimizer, perbedaan akurasi tidak terlalu besar pada learning rate yang sama. Pada learning rate 0.01, baik Adam maupun AdamW memberikan hasil yang hampir sama, sedangkan optimizer SGD selalu menghasilkan akurasi yang lebih rendah di setiap learning rate. Dengan demikian, dapat disimpulkan bahwa faktor yang paling berpengaruh terhadap peningkatan akurasi adalah pemilihan learning rate yang tepat, sedangkan jenis optimizer hanya memberikan pengaruh tambahan yang relatif kecil terhadap hasil akhir.

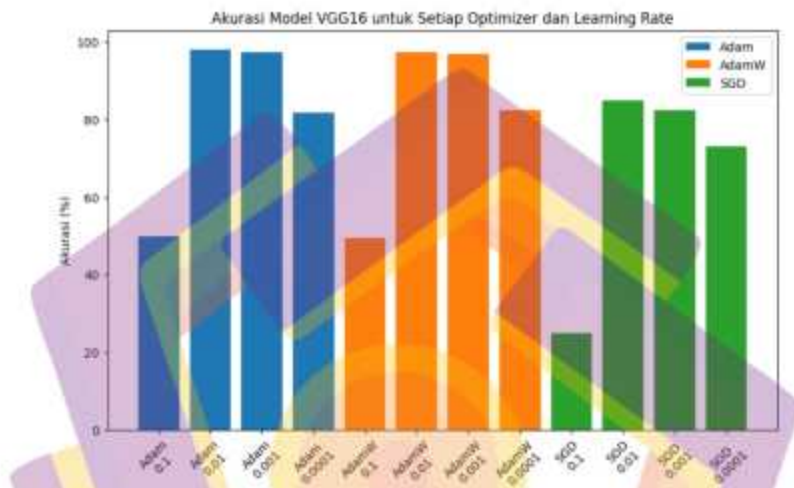
Selanjutnya, dilakukan pengujian terhadap variasi learning rate dengan menggunakan optimizer yang sama untuk mengetahui pengaruhnya secara lebih spesifik terhadap akurasi model. Hasil pengujian tersebut ditampilkan pada Tabel 4.3.

Tabel 4.3 Hasil Pengujian Optimizer yang Sama dan Learning Rate Berbeda

Optimizer	Learning Rate	Akurasi
Adam	0,1	50%
Adam	0,01	98,12%
Adam	0,001	97,50%
Adam	0,0001	81,87%
AdamW	0,1	49,38%
AdamW	0,01	97,50%
AdamW	0,001	96,88%
AdamW	0,0001	82,50%
SGD	0,1	25%
SGD	0,01	85%
SGD	0,001	82,50%
SGD	0,0001	73,12%

Tabel 4.3 menampilkan hasil pengujian model VGG16 dengan menggunakan optimizer yang sama namun menerapkan variasi learning rate yang berbeda. Tujuan pengujian ini adalah untuk melihat pengaruh perubahan learning

rate terhadap nilai akurasi yang dihasilkan oleh masing-masing optimizer. Hasil pada Tabel 4.3 kemudian divisualisasikan dalam Gambar 4.17 untuk memperlihatkan hubungan antara variasi learning rate dan tingkat akurasi pada setiap optimizer yang diuji.



Gambar 4.17 Akurasi VGG16 untuk Setiap Optimizer dan LR

Berdasarkan hasil pengujian yang dilakukan pada Gambar 4.17 terhadap model VGG16 dengan berbagai kombinasi *optimizer* dan *learning rate*, faktor yang paling berpengaruh terhadap peningkatan akurasi adalah *learning rate*. Hal ini terlihat dari perbedaan nilai akurasi yang signifikan ketika *learning rate* diubah, meskipun *optimizer* yang digunakan tetap sama. Pada *optimizer* Adam, perubahan *learning rate* dari 0.1 menjadi 0.01 menyebabkan peningkatan akurasi yang sangat besar, yaitu dari 50% menjadi 98,12%. Sementara itu, variasi *optimizer* pada *learning rate* yang sama tidak memberikan perubahan akurasi yang terlalu

signifikan. Pada *learning rate* 0.01, akurasi Adam sebesar 98,12%, AdamW sebesar 97,50%, dan SGD sebesar 85%.

Hal ini disebabkan karena *learning rate* berperan langsung dalam menentukan seberapa besar langkah pembaruan bobot pada setiap iterasi pelatihan. Jika nilainya terlalu tinggi, model cenderung tidak stabil dan sulit mencapai titik konvergensi optimal, sedangkan jika terlalu rendah proses pelatihan menjadi lambat. Oleh karena itu, *learning rate* yang seimbang memungkinkan model melakukan proses pembelajaran yang efisien dan stabil, sehingga menghasilkan akurasi yang lebih tinggi secara konsisten.

Berdasarkan hasil tersebut, dapat disimpulkan bahwa pemilihan *learning rate* yang tepat memiliki pengaruh yang lebih dominan terhadap kinerja model dibandingkan dengan jenis *optimizer* yang digunakan.

4.3.4 Hasil Evaluasi MobileNetV2

Gambar 4.17 merupakan visualisasi performa model MobileNetV2 berdasarkan lima skenario pelatihan dengan variasi jumlah epoch (15, 20, 25, 30, dan 35).



Gambar 4.18 Hasil Evaluasi MobileNetV2

Berdasarkan hasil eksperimen, epoch 30 menunjukkan performa tertinggi pada seluruh metrik evaluasi, yaitu akurasi 96,88%, presisi 96,89%, recall 96,89%, dan F1-score 96,87%, menjadikannya konfigurasi paling optimal dalam eksperimen ini. Di sisi lain, skenario lainnya (epoch 15, 20, 25, dan 35) menunjukkan hasil yang identik atau sangat berdekatan, dengan nilai akurasi dan recall stabil pada angka 96,25% serta F1-score pada kisaran 96,22%-96,24%.

Namun demikian, dari segi efisiensi waktu, epoch 35 dan epoch 25 memiliki waktu pelatihan tercepat, yaitu 92,5 detik, sedangkan epoch 20 memerlukan waktu terlama, yaitu 141,77 detik. Hal ini mengindikasikan bahwa peningkatan performa model pada epoch 30 tidak dibarengi dengan peningkatan signifikan pada waktu pelatihan, karena tetap berada dalam rentang waktu yang wajar (105,1 detik). Oleh karena itu, konfigurasi pada epoch 30 dengan optimizer Adam dan learning rate 0.01 direkomendasikan sebagai skenario terbaik karena memberikan keseimbangan antara akurasi tinggi dan waktu pelatihan yang efisien.

4.3.5 Skenario Terbaik MobileNetV2

Tabel 4.2 menyajikan berbagai skenario pelatihan MobileNetV2 untuk menentukan kombinasi terbaik antara jumlah epoch, optimizer, dan learning rate terhadap performa model. Evaluasi ini penting dilakukan untuk mengoptimalkan akurasi model sekaligus efisiensi waktu pelatihan.

Tabel 4.4 Skenario Terbaik Pada MobileNetV2

Jumlah Epoch	Optimizer	Learning Rate	Confusion Matrix & Training Time	Hasil
Epoch 15	AdamW	0.0001	Akurasi	96,25%
			Presisi	96,28%
			Recall	96,25%
			F1-Score	96,22%
			Training Time	94,12 seconds
Epoch 20	SGD	0.01	Akurasi	96,25%
			Presisi	96,25%
			Recall	96,25%
			F1-Score	96,23%
			Training Time	120,71 seconds
Epoch 25	AdamW	0.001	Akurasi	96,25%
			Presisi	96,26%
			Recall	96,25%
			F1-Score	96,22%
			Training Time	92,50 seconds
Epoch 30	Adam	0.01	Akurasi	96,88%
			Presisi	96,89%
			Recall	96,88%
			F1-Score	96,87%
			Training Time	105,10 seconds
Epoch 35	AdamW	0.001	Akurasi	96,25%
			Presisi	96,26%
			Recall	96,25%
			F1-Score	96,22%
			Training Time	92,50 seconds

Tabel 4.2 menunjukkan hasil skenario terbaik dari model MobileNetV2 berdasarkan variasi jumlah epoch, optimizer, dan learning rate. Dari lima skenario yang diuji, performa terbaik dicapai pada epoch 30 dengan menggunakan optimizer Adam dan learning rate 0.01. Pada skenario ini, model mencatatkan nilai akurasi

96,88%, presisi 96,89%, recall 96,88%, dan F1-score 96,87%, yang merupakan nilai tertinggi dibandingkan epoch lainnya.

Sementara itu, pada epoch 15, 20, 25, dan 35, hasil evaluasi model menunjukkan konsistensi performa dengan akurasi dan recall yang sama yaitu sebesar 96,25%, serta F1-score berkisar antara 96,22% hingga 96,23%, namun tidak mampu melampaui kinerja pada epoch 30. Hal ini menunjukkan bahwa peningkatan epoch hingga titik tertentu (dalam hal ini epoch 30) memberikan dampak positif terhadap kemampuan generalisasi model. Namun, setelahnya (epoch 35), tidak terjadi peningkatan performa, bahkan cenderung stagnan. Kombinasi parameter terbaik pada MobileNetV2 untuk hal klasifikasi dalam eksperimen ini diperoleh pada epoch 30 dengan *optimizer* Adam dan *learning rate* 0,01, yang memberikan performa paling optimal secara keseluruhan.

4.3.6 Faktor yang Mempengaruhi Skenario Terbaik pada MobileNetV2

Beberapa faktor berperan dalam menentukan skenario terbaik pada MobileNetV2, terutama yang berkaitan dengan konfigurasi parameter pelatihan serta hasil pengujian yang diperoleh. Faktor-faktor tersebut berkontribusi terhadap performa akhir model dalam mencapai tingkat akurasi yang optimal. Berikut penjelasan mengenai faktor-faktor yang memengaruhi hasil tersebut.

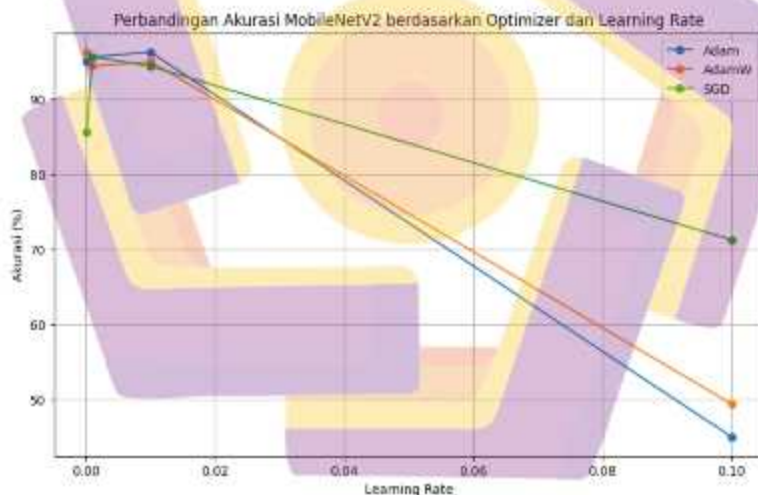
Tabel 4.5 Hasil Pengujian Optimizer Berbeda dan LR Sama

Optimizer	Learning Rate	Akurasi
Adam	0,1	45%
AdamW	0,1	49,38%
SGD	0,1	71,25%
Adam	0,01	96,25%
AdamW	0,01	95%
SGD	0,01	94,37%
Adam	0,001	95,63%

AdamW	0,001	94,37%
SGD	0,001	95,63%
Adam	0,0001	95%
AdamW	0,0001	96,25%
SGD	0,0001	85,62%

Tabel 4.5 menampilkan hasil pengujian model MobileNetV2 dengan menggunakan learning rate yang sama namun menerapkan jenis optimizer yang berbeda. Pengujian ini dilakukan untuk melihat seberapa besar pengaruh variasi optimizer terhadap tingkat akurasi model.

Hasil pengujian pada tabel sebelumnya divisualisasikan dalam Gambar 4.19 untuk menunjukkan perbandingan akurasi model MobileNetV2 pada setiap kombinasi optimizer dan learning rate.



Gambar 4.19 Perbandingan Akurasi MobileNetV2 (Optimizer dan LR)

Berdasarkan hasil pengujian model MobileNetV2 pada Gambar 4.19 dengan berbagai kombinasi optimizer dan learning rate, dapat disimpulkan bahwa faktor yang paling berpengaruh terhadap perubahan nilai akurasi adalah learning rate. Hal ini terlihat dari perbedaan nilai akurasi yang signifikan ketika learning rate diubah, meskipun jenis optimizer yang digunakan tetap sama. Misalnya, pada optimizer Adam, akurasi meningkat drastis dari 45% pada learning rate 0.1 menjadi 96,25% pada learning rate 0.01, lalu sedikit menurun pada nilai learning rate yang lebih kecil. Pola yang sama juga terlihat pada optimizer AdamW dan SGD, di mana learning rate 0.01 menghasilkan performa terbaik dibandingkan dengan nilai lainnya.

Sementara itu, jika dibandingkan antar optimizer pada learning rate yang sama, perbedaan akurasinya tidak terlalu besar. Sebagai contoh, pada learning rate 0.01, Adam mencapai akurasi 96,25%, AdamW 95%, dan SGD 94,37%. Hal ini menunjukkan bahwa meskipun jenis optimizer berperan dalam proses pembaruan bobot model, nilai learning rate yang digunakan menjadi faktor penentu utama dalam mencapai tingkat akurasi yang optimal. Dengan demikian, pemilihan learning rate yang tepat memiliki pengaruh yang lebih dominan terhadap hasil akhir performa model dibandingkan dengan pemilihan jenis optimizer.

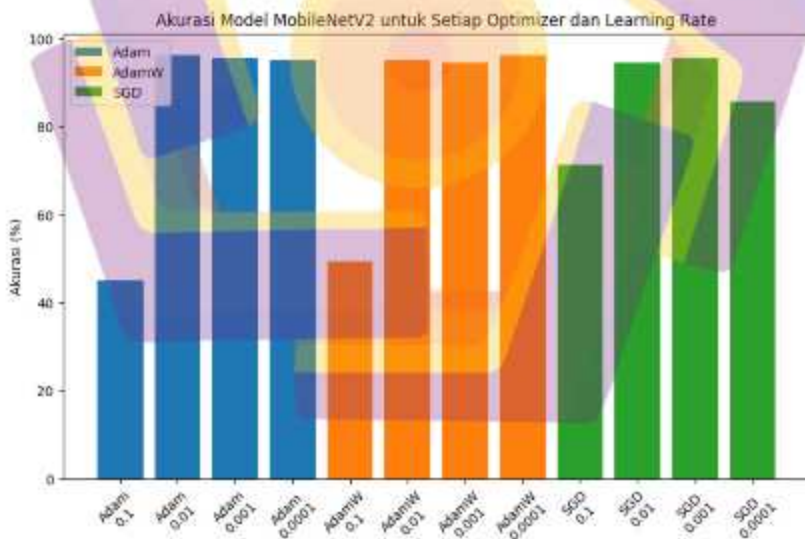
Tabel 4.6 Hasil Pengujian Optimizer Sama dan LR Berbeda

Optimizer	Learning Rate	Akurasi
Adam	0,1	45%
Adam	0,01	96,25%
Adam	0,001	95,63%
Adam	0,0001	95%
AdamW	0,1	49,38%
AdamW	0,01	95%
AdamW	0,001	94,37%

AdamW	0,0001	96,25%
SGD	0,1	71,25%
SGD	0,01	94,37%
SGD	0,001	95,63%
SGD	0,0001	85,62%

Tabel 4.6 menunjukkan hasil pengujian model MobileNetV2 dengan menggunakan optimizer yang sama namun menerapkan variasi learning rate yang berbeda. Tujuan pengujian ini adalah untuk mengetahui pengaruh perubahan nilai learning rate terhadap akurasi model pada masing-masing optimizer yang digunakan.

Hasil pengujian tersebut divisualisasikan pada Gambar 4.20 untuk menunjukkan perbandingan akurasi model MobileNetV2 berdasarkan variasi optimizer dan learning rate.



Gambar 4.20 Akurasi MobileNetV2 Berdasarkan Optimizer dan LR

Pada Gambar 4.20 menunjukkan perbandingan akurasi MobileNetV2 terhadap berbagai kombinasi antara optimizer dan learning rate. Sumbu horizontal menunjukkan kombinasi antara optimizer dan learning rate, sedangkan sumbu vertikal menunjukkan nilai akurasi dalam persen (%). Nilai akurasi meningkat secara signifikan ketika learning rate diatur pada nilai yang tepat, terutama pada learning rate 0.01. Pada optimizer Adam, akurasi meningkat dari 45% pada learning rate 0.1 menjadi 96,25% pada learning rate 0.01, lalu sedikit menurun pada nilai learning rate yang lebih kecil. Pola yang sama juga terlihat pada optimizer AdamW dan SGD, di mana learning rate 0.01 memberikan hasil terbaik di antara semua nilai lainnya.

Jika dilihat berdasarkan jenis optimizer, Adam dan AdamW menghasilkan performa yang hampir serupa, dengan akurasi tertinggi berkisar di atas 95%, sedangkan optimizer SGD memiliki akurasi sedikit lebih rendah, meskipun tetap menunjukkan peningkatan yang jelas pada learning rate tertentu.

Hasil akurasi bisa meningkat karena *learning rate* berada pada nilai optimal yang menyeimbangkan kecepatan dan kestabilan pembelajaran model. Dari hasil tersebut dapat disimpulkan bahwa faktor yang paling berpengaruh terhadap peningkatan akurasi adalah learning rate. Nilai learning rate yang terlalu tinggi membuat model sulit konvergen, sedangkan nilai yang terlalu rendah memperlambat proses pelatihan. Oleh karena itu, pemilihan learning rate yang optimal menjadi kunci utama dalam mendapatkan performa terbaik pada model MobileNetV2.

4.3.7 Perbandingan Loop

Berikut hasil penelitian dengan loop dan tanpa loop yang dapat dilihat pada Tabel 4.3.

Tabel 4.7 Perbandingan Loop dan Tanpa Loop

No	Model Training		Akurasi	Presisi	Recall	F1 Score
1	Dengan Loop	VGG16	98,12%	98,17%	98,12%	98,12%
		MobilenetV2	96,25%	96,28%	96,25%	96,22%
2	Tanpa Loop	VGG16	25%	60%	25%	10%
		MobilenetV2	70%	70%	100%	82%

Pada Tabel 4.3 terdapat model yang menggunakan *loop*, baik model VGG16 maupun MobileNetV2 mampu mencapai kinerja yang tinggi. Hal ini menunjukkan bahwa penggunaan *loop* memberikan hasil pelatihan yang lebih akurat. Sebaliknya, pada model tanpa menggunakan *loop*, performa model jauh lebih rendah dan tidak stabil. Model VGG16 menghasilkan akurasi 25%, dengan presisi 60%, recall 25%, serta F1-score 10%. Hal ini menandakan bahwa proses pelatihan yang dilakukan tanpa *loop* tidak berjalan efektif sehingga model gagal melakukan generalisasi dengan baik. Meskipun MobileNetV2 pada metode tanpa loop menunjukkan akurasi lebih baik yaitu 70%, hasil tersebut masih jauh di bawah performa yang diperoleh dengan loop, serta distribusi metrik yang tidak seimbang (recall 100% namun akurasi dan presisi relatif rendah).

Secara keseluruhan, hasil pada penggunaan *loop* dalam penelitian ini memiliki peran yang sangat penting karena memungkinkan pelatihan model dilakukan secara otomatis untuk setiap kombinasi parameter yang ditentukan. Dengan demikian, proses eksperimen menghasilkan performa yang lebih optimal dibandingkan dengan metode tanpa loop.

4.3.8 Perbandingan Penelitian

Tahapan berikutnya, untuk mengetahui efektivitas model dalam mengenali penyakit daun pisang, penelitian ini melakukan evaluasi terhadap beberapa pendekatan klasifikasi. Temuan awal mengindikasikan perbandingan dengan penelitian sebelumnya dilakukan untuk mengetahui posisi dan kontribusi penelitian ini terhadap studi-studi terdahulu. Melalui perbandingan ini dapat dilihat apakah metode yang digunakan mampu menghasilkan performa yang lebih baik atau efisien. Dengan demikian, bagian ini penting untuk menilai keunggulan dan relevansi model yang dikembangkan. Hasil perbandingan penelitian dapat dilihat pada Tabel 4.4.

Tabel 4.8 Perbandingan Penelitian

No	Author	Dataset	Augmentasi	Model	Hasil			
					Akurasi	Presisi	Recall	F1-Score
1	(Tanwar et al., 2023)	<i>Banana Leaf</i>	Tidak menggunakan augmentasi	<i>Hybrid CNN-SVM</i>	90%	86%	86%	86%
2	(Andreanov Ridhovan et al., 2022)	<i>Banana Leaf</i>	<i>Rescale</i> <i>Rotation range</i> <i>Vertical range</i> <i>Vertical flip</i> <i>Shear range</i> <i>Zoom range</i> <i>Fill mode</i>	<i>DenseNet and Inception Method</i>	84,73%	84,80%	84,73%	84,62%
3	(Tirandasu & Yalla, 2024)	<i>Banana Leaf</i>	Tidak dilakukan augmentasi dalam penelitian ini, fokus utamanya adalah pada ekstraksi dan optimisasi fitur	<i>BAT-KNN Hybrid</i>	95%	94%	94,50%	94.80%
4	(Bluiyan et al., 2023)	<i>Banana Leaf</i>	<i>Gaussian blur</i>	<i>ResNet-50</i>	86,25%	89,26%	86,25%	85,46%
			<i>Flip</i>	<i>Inception-V3</i>	90%	91,96%	90%	89,30%
			<i>Croppal</i>	<i>VGG-16</i>	95,50%	95,45%	95%	94,84%
			<i>Linear contrast</i>	<i>EfficientNetB0</i>	87,50%	91,66%	87,50%	87,67%
			<i>Rotate shear</i>	<i>MobileNetV3</i>	86,25%	89,74%	86,25%	86,22%

			<i>Translate Shear</i>	<i>ResNet-101</i>	90%	92,86%	90%	93,30%
				<i>SqueezeNet</i>	96,25%	96,54%	96,25%	96,17%
5	<i>This study's approach/model</i>	<i>Banana Leaf</i>	<i>Gaussian blur, Cropping, Flipping, Rotasi</i>	<i>VGG16</i>	98,12%	98,17%	98,12%	98,12%
				<i>MobileNetV2</i>	96,25%	96,28%	96,25%	96,22%

Tabel 4.4 menyajikan perbandingan hasil penelitian terkait klasifikasi citra daun pisang menggunakan model *deep learning*. Perbandingan ini mencakup metrik evaluasi berupa akurasi, presisi, recall, dan F1-score yang digunakan untuk menilai performa masing-masing model. Dataset yang digunakan pada seluruh penelitian dalam tabel ini adalah citra daun pisang, yang mengindikasikan bahwa konteks perbandingan tetap relevan secara tematis.

Pada penelitian yang dilakukan oleh (Tanwar et al., 2023; Tirandasu & Yalla, 2024) tidak diterapkan teknik augmentasi, sehingga kinerja model sepenuhnya bergantung pada arsitektur yang digunakan dan strategi optimisasi fitur yang diterapkan. Sebaliknya, penelitian (Andreanov Ridhovan et al., 2022) telah menerapkan beberapa teknik augmentasi dasar, antara lain rescale, rotasi, vertical flip, shear, zoom, dan fill mode. Penerapan strategi tersebut ditujukan untuk memperluas variasi data latih sehingga model memperoleh kemampuan generalisasi yang lebih baik. Namun demikian, capaian akurasi dan F1-score yang dihasilkan masih berada pada kisaran 84%.

Selanjutnya, penelitian (Bhuiyan et al., 2023) menggunakan variasi augmentasi yang lebih kompleks meliputi Gaussian blur, flipping, cropping, linear contrast, rotate shear, translate, dan shear. Penerapan kombinasi augmentasi yang lebih beragam dan mampu meningkatkan kinerja beberapa model *deep learning*

secara signifikan. Model VGG-16 berhasil mencapai akurasi sebesar 95,50%, sementara SqueezeNet memperoleh performa terbaik dengan akurasi 96,25%.

Adapun dalam penelitian saat ini, teknik augmentasi menggunakan yaitu Gaussian blur, cropping, flipping, dan rotasi. Strategi augmentasi ini mampu memperkaya variasi citra daun pisang dan meningkatkan efektivitas proses pembelajaran model. Hasilnya, model VGG16 memperoleh performa tertinggi dengan akurasi 98,12%, sedangkan MobileNetV2 juga menunjukkan kinerja kompetitif dengan akurasi 96,25%.

Penelitian (Tanwar et al., 2023) menggunakan model Hybrid CNN-SVM dan memperoleh akurasi sebesar 90%, dengan nilai presisi, recall, dan F1-score masing-masing sebesar 86%. Sementara itu, pendekatan yang dilakukan oleh (Andreanov Ridhovan et al., 2022) dengan menggunakan kombinasi DenseNet dan *inception method* mencatatkan akurasi sebesar 84,73%, dengan nilai F1-score sebesar 84,62%. Penelitian (Tirandasu & Yalla, 2024) menunjukkan hasil yang lebih baik, dengan model BAT-KNN Hybrid yang mencapai akurasi 95% dan F1-score 94,80%.

Selanjutnya penelitian yang dilakukan oleh (Bhuiyan et al., 2023) berbagai arsitektur model, seperti ResNet-50, Inception-V3, VGG-16, EfficientNetB0, MobileNetV3, ResNet-101, dan SqueezeNet, dengan hasil yang bervariasi. Di antara model-model tersebut, SqueezeNet menunjukkan performa terbaik dengan akurasi 96,25% dan F1-score 96,17%, sementara MobileNetV3 menunjukkan performa yang paling rendah dengan akurasi 86,25%.

Adapun hasil penelitian ini menunjukkan bahwa model VGG16 dan MobileNetV2 yang digunakan mampu mengungguli mayoritas model sebelumnya. Pada VGG16 mencapai hasil akurasi tertinggi sebesar 98,12% dengan F1-score 98,12%, sementara MobileNetV2 menunjukkan performa yang sangat kompetitif dengan akurasi 96,25% dan F1-score 96,22%. Hasil dari kedua model ini menunjukkan bahwa pendekatan yang diusulkan dalam penelitian ini tidak hanya efektif, tetapi juga lebih unggul dalam klasifikasi penyakit daun pisang dibandingkan dengan beberapa model yang telah ada sebelumnya.



BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil penelitian yang sudah dilakukan, dapat disimpulkan bahwa:

1. Model VGG16 menunjukkan performa yang lebih unggul dibandingkan MobileNetV2 dalam klasifikasi penyakit daun pisang. VGG16 dengan konfigurasi terbaik (epoch 15, optimizer adam, learning rate 0,01) mencapai akurasi sebesar 98,12%, presisi 98,17%, recall 98,12%, dan F1-score 98,12%, dengan waktu pelatihan sekitar 79,92 detik. Hasil ini menunjukkan bahwa meskipun MobileNetV2 lebih ringan dan efisien secara arsitektur, VGG16 memberikan hasil klasifikasi yang lebih akurat pada dataset yang digunakan.
2. Penggunaan kedua model CNN ini secara signifikan dapat meningkatkan akurasi klasifikasi penyakit daun pisang. VGG16 memberikan akurasi tertinggi hingga 98,12%, yang menunjukkan keunggulan dalam mengenali pola dan fitur penting pada citra penyakit daun pisang. MobileNetV2 juga memberikan akurasi yang baik, meskipun sedikit lebih rendah, namun dengan keunggulan efisiensi komputasi. Dengan demikian, penerapan kedua model ini dapat meningkatkan akurasi klasifikasi, dimana pilihan model dapat disesuaikan dengan kebutuhan antara akurasi tinggi (VGG16) atau efisiensi komputasi (MobileNetV2).
3. *Optimizer* Adam pada VGG16 dan AdamW pada MobileNetV2 menghasilkan akurasi tinggi dengan *learning rate* sebesar 0,01 untuk VGG16 dan 0,0001

untuk MobileNetV2. *Learning rate* yang terlalu besar menyebabkan model sulit mencapai hasil yang baik, sedangkan *learning rate* yang terlalu kecil membuat proses pelatihan berjalan lambat dan kurang optimal. Dengan demikian, dapat disimpulkan bahwa faktor yang paling berpengaruh terhadap performa dan tingkat akurasi model adalah pemilihan *learning rate* yang tepat. Nilai *learning rate* yang optimal mampu menyeimbangkan antara kecepatan pembelajaran dan kestabilan konvergensi, sehingga model dapat mencapai akurasi tinggi secara konsisten.

5.2. Saran

Berdasarkan hasil dan pengalaman yang diperoleh dalam penelitian ini, terdapat beberapa saran yang dapat menjadi pertimbangan bagi peneliti selanjutnya dalam melakukan penelitian serupa, antara lain:

1. Berdasarkan hasil evaluasi *confusion matrix*, masih terdapat kesalahan klasifikasi terutama pada kelas pestalotiopsis yang terdeteksi sebagai sigatoka, serta pada kelas cordana yang di klasifikasi sebagai pestalotiopsis. Hal ini mengindikasikan bahwa model masih mengalami kesulitan dalam membedakan ciri visual antara kelas-kelas tersebut. Oleh karena itu, pada penelitian selanjutnya disarankan untuk menambah jumlah dataset khususnya pada kelas pestalotiopsis dan cordana agar distribusi data menjadi lebih seimbang serta representasi fitur setiap kelas lebih beragam. Penambahan variasi data diharapkan dapat meningkatkan kemampuan model dalam membedakan pola visual antar kelas, sehingga performa klasifikasi menjadi lebih optimal.

2. Penelitian ini masih dapat diperluas dengan menambahkan kelas penyakit daun pisang lainnya yang belum tercakup dalam dataset, sehingga cakupan klasifikasi menjadi lebih lengkap. Penambahan kelas tersebut diharapkan dapat meningkatkan kepraktisan model dalam mendeteksi berbagai jenis penyakit di kondisi lapangan secara lebih aplikatif dan komprehensif.
3. Penggunaan *optimizer* dalam pelatihan model pada penelitian ini masih terbatas pada 3 jenis *optimizer*. Oleh karena itu, disarankan untuk melakukan eksperimen lebih lanjut dengan menggunakan *optimizer* lain yang belum diuji, seperti Adagrad, Adadelata, Nadam, atau FTRL. Pengujian terhadap berbagai jenis *optimizer* ini penting untuk memperoleh pemahaman yang lebih mendalam terkait pengaruh metode optimasi terhadap performa model, baik dari segi akurasi, konvergensi, maupun efisiensi pelatihan. Dengan demikian, penelitian selanjutnya dapat menemukan konfigurasi *optimizer* yang paling sesuai dengan karakteristik data penyakit daun pisang yang digunakan.

DAFTAR PUSTAKA

- Alex David, S., Begum, A., Kalilulah, S. I., Ruth Naveena, N., Rajathi, K., & Hemalatha, D. (2024). Deep Learning Advances in Brain Tumor Classification: Leveraging VGG16 and MobileNetV2 for Accurate MRI Diagnostics. *4th International Conference on Power, Energy, Control and Transmission Systems: Harnessing Power and Energy for an Affordable Electrification of India, ICPECTS 2024*. <https://doi.org/10.1109/ICPECTS62210.2024.10780014>
- Andreanov Ridhovan, Aries Suharso, & Chaerur Rozikin. (2022a). Disease Detection in Banana Leaf Plants using DenseNet and Inception Method. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 6(5), 710–718. <https://doi.org/10.29207/resti.v6i5.4202>
- Andreanov Ridhovan, Aries Suharso, & Chaerur Rozikin. (2022b). Disease Detection in Banana Leaf Plants using DenseNet and Inception Method. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 6(5), 710–718. <https://doi.org/10.29207/resti.v6i5.4202>
- Bhuiyan, M. A. B., Abdullah, H. M., Arman, S. E., Saminur Rahman, S., & Al Mahmud, K. (2023). BananaSqueezeNet: A very fast, lightweight convolutional neural network for the diagnosis of three prominent banana leaf diseases. *Smart Agricultural Technology*, 4. <https://doi.org/10.1016/j.atech.2023.100214>
- Chaudhari, V., & Patil, M. P. (2023a). Detection and Classification of Banana Leaf Disease Using Novel Segmentation and Ensemble Machine Learning

- Approach. *Applied Computer Systems*, 28(1), 92–99.
<https://doi.org/10.2478/acss-2023-0009>
- Chaudhari, V., & Patil, M. P. (2023b). Detection and Classification of Banana Leaf Disease Using Novel Segmentation and Ensemble Machine Learning Approach. *Applied Computer Systems*, 28(1), 92–99.
<https://doi.org/10.2478/acss-2023-0009>
- Deng, J., Huang, W., Zhou, G., Hu, Y., Li, L., & Wang, Y. (2024). Identification of banana leaf disease based on KVA and GR-ARNet. *Journal of Integrative Agriculture*, 23(10), 3554–3575. <https://doi.org/10.1016/j.jia.2023.11.037>
- Fitriah, D., Suryaningrum, K. M., Sagala, N. T. M., Ayumi, V., & Lim, S. M. (2022). Fine-Tuned MobileNetV2 and VGG16 Algorithm for Fish Image Classification. *Proceedings - 4th International Conference on Informatics, Multimedia, Cyber and Information System, ICIMCIS 2022*, 384–389.
<https://doi.org/10.1109/ICIMCIS56303.2022.10017901>
- Hadi, H. P., Rachmawanto, E. H., & Ali, R. R. (2024). Comparison of DenseNet-121 and MobileNet for Coral Reef Classification. *MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, 23(2), 333–342.
<https://doi.org/10.30812/matrik.v23i2.3683>
- Haris Rangkuti, A., Lun Lau, S., Athala Hasbi, V., Haridhi Indallah, F., Fajri Rosyadah, R., & Aryanto, R. (2023). Development of Banana Image Retrieval Based on its Class Using the CNN Model and Image Distance. *2023 IEEE International Conference on Computing, ICOCO 2023*, 426–431.
<https://doi.org/10.1109/ICOCO59262.2023.10397697>

- Hindarto, D. (2023a). Comparative Analysis VGG16 Vs MobileNet Performance for Fish Identification. *International Journal Software Engineering and Computer Science (IJSECS)*, 3(3), 270–280. <https://doi.org/10.35870/ijsecs.v3i3.1763>
- Hindarto, D. (2023b). Comparative Analysis VGG16 Vs MobileNet Performance for Fish Identification. *International Journal Software Engineering and Computer Science (IJSECS)*, 3(3), 270–280. <https://doi.org/10.35870/ijsecs.v3i3.1763>
- Ibarra, N. C., Rivera, M. P., & Manlises, C. O. (2023). Detection of Panama Disease on Banana Leaves Using the YOLOv4 Algorithm. *2023 15th International Conference on Computer and Automation Engineering, ICCAE 2023*, 209–214. <https://doi.org/10.1109/ICCAE56788.2023.10111416>
- Kumar, M., & Kumar, A. (2024). Deep Learning Meets-Support Vector Machines: An Effective Hybrid Model for Banana Leaf Wilt Disease Severity Assessment. *2024 2nd International Conference on Disruptive Technologies, ICDT 2024*, 386–390. <https://doi.org/10.1109/ICDT61202.2024.10489081>
- Kumar, S., & Kumar, H. (2024). Efficient-VGG16: A Novel Ensemble Method for the Classification of COVID-19 X-ray Images in Contrast to Machine and Transfer Learning. *Procedia Computer Science*, 235, 1289–1299. <https://doi.org/10.1016/j.procs.2024.04.122>
- Liao, X., Wu, Y., Jiang, N., Sun, J., Xu, W., Gao, S., Wang, J., Li, T., Wang, K., & Li, Q. (2023). Automated detection of abnormal respiratory sound from electronic stethoscope and mobile phone using MobileNetV2. *Biocybernetics*

and Biomedical Engineering, 43(4), 763–775.
<https://doi.org/10.1016/j.bbe.2023.11.001>

Md Ripon Sheikh, Md. A. H. M. H. Md. M. I. G. M. S. H. (2024). BananaSet: A dataset of banana varieties in Bangladesh. *Www.Elsevier.Com/Locate/Dib*.
<https://doi.org/10.17632/35gb4v72dr.4>

Mukhtar Amin. (2023). *Produksi Pisang Dunia Berdasarkan Negara*. Atlasbig.Com.

Prasetyo, A., & Utami, E. (2024). Detection and Classification of Banana Leaf Diseases: Systematic Literature Review. *Telematika*, 17(2), 128–141.
<https://doi.org/10.35671/telematika.v17i2.2809>

Raharjo, B. (2022). *Deep Learning dengan Python*.

Rifkie Primartha. (2018). *Belajar Machine Learning Teori dan Praktik*. Informatika.

Rifkie Primartha. (2021). *Algoritma Machine Learning*. Informatika.

Samridhi, S., Kalpana, M., Parimalaragan, R., & Palanichamy, N. V. (2023). Identification of Sigatoka Leaf Spot Disease in Banana Using Convolutional Neural Network (CNN). *Asian Journal of Agricultural Extension, Economics & Sociology*, 41(9), 931–936. <https://doi.org/10.9734/ajaees/2023/v41i92123>

Seetharaman, K., & Mahendran, T. (2022). Leaf Disease Detection in Banana Plant using Gabor Extraction and Region-Based Convolution Neural Network (RCNN). *Journal of The Institution of Engineers (India): Series A*, 103(2), 501–507. <https://doi.org/10.1007/s40030-022-00628-2>

- Senthil Pandi, S., Shanmugapriya, S., Gethara Gowri, C., & Kumar, P. (2024). Accuracy Enhancement in Apple Leaf Diseases Detection and Classification Using VGG16. *Proceedings of 2024 IEEE International Conference of Electron Devices Society Kolkata Chapter, EDKCON 2024*, 494–499. <https://doi.org/10.1109/EDKCON62339.2024.10870809>
- Shetty, S., & Mahesh, T. R. (2024). SKGDC: Effective Segmentation Based Deep Learning Methodology for Banana Leaf, Fruit, and Stem Disease Prediction. *SN Computer Science*, 5(6). <https://doi.org/10.1007/s42979-024-03031-9>
- Shi, S. (2024). A comparison of MobileNetV2, VGG16, and ResNet50 for classifying brain tumors. *2024 IEEE 2nd International Conference on Electrical, Automation and Computer Engineering, ICEACE 2024*, 176–180. <https://doi.org/10.1109/ICEACE63551.2024.10898757>
- Shifat E. Arman, Md. A. B. B. H. M. A. S. I. T. T. C. Md. A. H. (2024). *BananaLSD: A banana leaf images dataset for classification of banana leaf diseases using machine learning*. <https://doi.org/10.1016>
- Suri, Niranjana, & Cabri, Giacomo. (2014). *Adaptive, dynamic, and resilient systems*. CRC Press, Taylor & Francis Group.
- Tanwar, V., Sharma, B., & Aanand, V. (2023). Implementing a Hybrid CNN-SVM Model for Banana Leaf Disease Classification. *2023 IEEE International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering, RMKMATE 2023*. <https://doi.org/10.1109/RMKMATE59243.2023.10369215>

- Tian, J., Jin, Q., Wang, Y., Yang, J., Zhang, S., & Sun, D. (2024). Performance analysis of deep learning-based object detection algorithms on COCO benchmark: a comparative study. *Journal of Engineering and Applied Science*, 71(1). <https://doi.org/10.1186/s44147-024-00411-z>
- Tirandasu, R. K., & Yalla, P. (2024). Predicting Banana Leaf Diseases: Feature Extraction with BL-FEOT and Enhanced Classification using the BAT-KNN Hybrid Algorithm. *SSRG International Journal of Electronics and Communication Engineering*, 11(5), 195–206. <https://doi.org/10.14445/23488549/IJECE-V11I5P119>
- Vakalopoulou, M., Christodoulidis, S., Burgos, N., Colliot, O., & Lepetit, V. (2023). Deep learning: basics and convolutional neural networks (CNN). [https://Hal.Science/Hal-03957224v2](https://hal.science/Hal-03957224v2), 197, 77–115. https://doi.org/10.1007/978-1-0716-3195-9_3i

LAMPIRAN

Lampiran 1 Source Code

```
# Import Library
import tensorflow as tf
from tensorflow.keras.preprocessing.image import
ImageDataGenerator
from tensorflow.keras.applications import VGG16, MobileNetV2
from tensorflow.keras import layers, models, optimizers
from sklearn.metrics import classification_report,
confusion_matrix, accuracy_score, precision_score,
recall_score, f1_score
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import os
import time

# Pastikan TensorFlow menggunakan GPU (Jika ada)
physical_devices = tf.config.list_physical_devices('GPU')
if physical_devices:
    tf.config.experimental.set_memory_growth(physical_devices[
0], True)
    print("GPU detected, using CUDA.")
else:
    print("GPU not detected, running on CPU.")

# Set Path Dataset
train_dir = 'splitbananaleaf/train'
val_dir = 'splitbananaleaf/val'
test_dir = 'splitbananaleaf/test'

batch_size = 32
img_height, img_width = 224, 224
```

```
import tensorflow as tf
print("TensorFlow version:", tf. version )
print("Num GPUs Available: ",
len(tf.config.list_physical_devices('GPU')))
print(tf.config.list_physical_devices('GPU'))
```

```

# Data Preprocessing
train_datagen = ImageDataGenerator(rescale=1./255)
val_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(img height, img width),
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=True
)

val_generator = val_datagen.flow_from_directory(
    val_dir,
    target_size=(img height, img width),
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=False
)

test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(img height, img width),
    batch_size=1,
    class_mode='categorical',
    shuffle=False
)

```

```

# Build Model Function
def build_model(base_model):
    base_model.trainable = False
    model = models.Sequential([
        base_model,
        layers.GlobalAveragePooling2D(),
        layers.Dense(128, activation='relu'),
        layers.Dropout(0.5),
        layers.Dense(4, activation='softmax')
    ])
    return model

```

```

# Function to get optimizer
def get_optimizer(opt_name, lr):

```

```

if opt_name == 'Adam':
    return optimizers.Adam(learning_rate=lr)
elif opt_name == 'AdamW':
    return optimizers.AdamW(learning_rate=lr)
elif opt_name == 'SGD':
    return optimizers.SGD(learning_rate=lr, momentum=0.9)
else:
    raise ValueError("Unknown optimizer")

# Plot Training History
def plot_history(history, title):
    acc = history.history['accuracy']
    val_acc = history.history['val_accuracy']
    loss = history.history['loss']
    val_loss = history.history['val_loss']

    epochs_range = range(len(acc))

    plt.figure(figsize=(14, 5))
    plt.subplot(1, 2, 1)
    plt.plot(epochs_range, acc, label='Training Accuracy')
    plt.plot(epochs_range, val_acc, label='Validation Accuracy')
    plt.legend(loc='lower right')
    plt.title(f'{title} - Accuracy')

    plt.subplot(1, 2, 2)
    plt.plot(epochs_range, loss, label='Training Loss')
    plt.plot(epochs_range, val_loss, label='Validation Loss')
    plt.legend(loc='upper right')
    plt.title(f'{title} - Loss')
    plt.show()

# Plot Confusion Matrix
def plot_confusion_matrix(y_true, y_pred, labels, title):
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=labels, yticklabels=labels)
    plt.ylabel('Actual')
    plt.xlabel('Predicted')
    plt.title(title)
    plt.show()

```

```

# Hyperparameters
optimizers_list = ['Adam', 'AdamW', 'SGD']
learning_rates = [0.1, 0.01, 0.001, 0.0001]

epochs = 30

# Loop untuk semua kombinasi optimizer dan learning rate
for opt_name in optimizers_list:
    for lr in learning_rates:
        print(f"===== Training VGG16 dengan {opt_name} (lr={lr}) =====")

        start_time = time.time()

        vgg16_base = VGG16(weights='imagenet',
include_top=False, input_shape=(img_height, img_width, 3))
        model_vgg16 = build_model(vgg16_base)
        model_vgg16.compile(optimizer=get_optimizer(opt_name,
lr),
                            loss='categorical_crossentropy',
                            metrics=['accuracy'])

        history_vgg16 = model_vgg16.fit(
            train_generator,
            epochs=epochs,
            validation_data=val_generator)

        plot_history(history_vgg16, f"VGG16 - {opt_name} -
(lr)")

        end_time = time.time()
        training_time = end_time - start_time

        # Evaluasi di Test Set
        print("\nEvaluasi di Test Set VGG16:")
        test_generator.reset()
        y_true = test_generator.classes
        y_pred = np.argmax(model_vgg16.predict(test_generator,
steps=test_generator.samples), axis=1)

        # Metrics
        accuracy = accuracy_score(y_true, y_pred)

```

```

        precision = precision_score(y_true, y_pred,
average='weighted')
        recall = recall_score(y_true, y_pred,
average='weighted')
        f1 = f1_score(y_true, y_pred, average='weighted')

        print(f"Accuracy: {accuracy:.4f}")
        print(f"Precision: {precision:.4f}")
        print(f"Recall: {recall:.4f}")
        print(f"F1-Score: {f1:.4f}")
        print(f"Training Time: {training_time:.2f} seconds")

        # Classification Report
        print(classification_report(y_true, y_pred,
target_names=list(train_generator.class_indices.keys())))

        # Confusion Matrix
        plot_confusion_matrix(y_true, y_pred,
list(train_generator.class_indices.keys()), f"Confusion Matrix
VGG16 - {opt_name} - {lr}={lr}")

        print(f"\n==== Training MobileNetV2 denoah {opt_name}
{lr}={lr} ====")

        start_time = time.time()

        mobilenetv2_base = MobileNetV2(weights='imagenet',
include_top=False, input_shape=(img_height, img_width, 3))
        model_mobilenetv2 = build_model(mobilenetv2_base)
        model_mobilenetv2.compile(optimizer=get_optimizer(opt_
name, lr),
                                loss='categorical_crossentropy',
                                metrics=['accuracy'])

        history_mobilenetv2 = model_mobilenetv2.fit(
            train_generator,
            epochs=epochs,
            validation_data=val_generator
        )

        plot_history(history_mobilenetv2, f"MobileNetV2 -
{opt_name} - {lr}")

        end_time = time.time()

```

```

training_time = end_time - start_time

# Evaluasi di Test Set
print("\nEvaluasi di Test Set MobileNetV2:")
test_generator.reset()
y_true = test_generator.classes
y_pred =
np.argmax(model mobilenetv2.predict(test_generator,
steps=test_generator.samples), axis=1)

# Metrics
accuracy = accuracy_score(y_true, y_pred)
precision = precision_score(y_true, y_pred,
average='weighted')
recall = recall_score(y_true, y_pred,
average='weighted')
f1 = f1_score(y_true, y_pred, average='weighted')

print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1-Score: {f1:.4f}")
print(f"Training Time: {training_time:.2f} seconds")

# Classification Report
print(classification_report(y_true, y_pred,
target_names=list(train_generator.class_indices.keys())))

# Confusion Matrix
plot_confusion_matrix(y_true, y_pred,
list(train_generator.class_indices.keys()), f"Confusion Matrix
MobileNetV2 - {opt_name} - Ir={Ir}")

```