

TESIS
PENINGKATAN PERFORMA KLASIFIKASI PENYAKIT DAUN
PADI PADA DATASET TERBATAS MENGGUNAKAN FINE
TUNING ARSITEKTUR INCEPTIONV3



disusun oleh

ROY JORDY

24.51.1564

Konsentrasi : Business Intelligence

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2026

TESIS
PENINGKATAN PERFORMA KLASIFIKASI PENYAKIT DAUN PADI PADA
DATASET TERBATAS MENGGUNAKAN FINE TUNING
ARSITEKTUR INCEPTIONV3

IMPROVING THE PERFORMANCE OF RICE LEAF DISEASE
CLASSIFICATION ON LIMITED DATASETS USING FINE
TUNING OF THE INCEPTIONV3 ARCHITECTURE

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



disusun oleh

ROY JORDY

24.51.1564

Konsentrasi : Business Intelligence

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2026

HALAMAN PERSETUJUAN

PENINGKATAN PERFORMA KLASIFIKASI PENYAKIT DAUN PADI
PADA DATASET TERBATAS MENGGUNAKAN FINE-TUNING
ARSITEKTUR INCEPTIONV3

IMPROVING THE PERFORMANCE OF RICE LEAF DISEASE
CLASSIFICATION ON LIMITED DATASETS USING FINE
TUNING OF THE INCEPTIONV3 ARCHITECTURE

yang disusun dan diajukan oleh

Roy Jordy

24.51.1564

telah disetujui oleh Dosen Pembimbing Tesis
pada tanggal tanggal 9 Januari 2026

Dosen Pembimbing,



Dhan Ariatmanto, S.Kom., M.Kom., Ph.D.

NIK. 190302197

HALAMAN PENGESAHAN

PENINGKATAN PERFORMA KLASIFIKASI PENYAKIT DAUN PADI
PADA DATASET TERBATAS MENGGUNAKAN FINE-TUNING
ARSITEKTUR INCEPTIONV3

IMPROVING THE PERFORMANCE OF RICE LEAF DISEASE
CLASSIFICATION ON LIMITED DATASETS USING FINE
TUNING OF THE INCEPTIONV3 ARCHITECTURE

yang disusun dan diajukan oleh

Roy Jordy

24.51.1564

Telah dipertahankan di depan Dewan Penguji
pada tanggal 9 Januari 2026

Susunan Dewan Penguji

Nama Penguji

Dr. Andi Sunvoto, M.Kom.
NIK. 190302052

Dr. Ferry Wahyu Wibowo, S.Si, M.Cs.
NIK. 190302235

Dhani Ariatmanto, S.Kom., M.Kom., Ph.D.
NIK. 190302197

Tanda Tangan



Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer
Tanggal 9 Januari 2026

DEKAN FAKULTAS ILMU KOMPUTER



Prof. Dr. Kusriani, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Roy Jordy
NIM : 24.51.1564

Menyatakan bahwa Tesis dengan judul berikut:

Peningkatan performa klasifikasi penyakit daun padi pada dataset terbatas menggunakan fine-tuning arsitektur inceptionv3

Dosen Pembimbing : Dhani Ariatmanto, S.Kom., M.Kom., Ph.D.

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 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, 9 Januari 2026

Yang Menyatakan,



Roy Jordy

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan tesis dengan judul **“Peningkatan Performa Klasifikasi Penyakit Daun Padi pada Dataset Terbatas Menggunakan Fine-Tuning Arsitektur InceptionV3”** dengan baik dan tepat waktu.

Tesis ini disusun sebagai salah satu syarat untuk memperoleh gelar Magister pada Program Studi Magister Informatika. Penulis menyadari bahwa penyusunan tesis ini tidak akan terwujud tanpa adanya bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Dr. Andi Sunyoto, M.Kom., selaku Dosen Penguji I, atas saran, kritik, dan masukan yang diberikan dalam rangka penyempurnaan tesis ini.
2. Dr. Ferry Wahyu Wibowo, S.Si., M.Cs., selaku Dosen Penguji II, atas arahan dan masukan yang membangun selama proses evaluasi tesis.
3. Dhani Ariatmanto, S.Kom., M.Kom., Ph.D., selaku Dosen Pembimbing sekaligus Dosen Penguji III, atas bimbingan, arahan, serta motivasi yang diberikan kepada penulis selama proses penyusunan tesis ini.

Penulis juga menyampaikan terima kasih kepada kedua orang tua dan keluarga yang senantiasa memberikan doa, dukungan, serta semangat selama menempuh pendidikan dan penyusunan tesis ini. Ucapan terima kasih turut penulis sampaikan kepada rekan-rekan sekelas serta semua pihak yang telah membantu, baik secara langsung maupun tidak langsung, dalam penyelesaian tesis ini.

Penulis menyadari bahwa tesis ini masih memiliki keterbatasan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diharapkan guna penyempurnaan penelitian ini. Semoga tesis ini dapat memberikan manfaat dan kontribusi bagi pengembangan ilmu pengetahuan, khususnya di bidang informatika.

Yogyakarta, 20 Januari 2026

Penulis



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN KEASLIAN TESIS	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
INTISARI	xv
ABSTRACT	xvi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	6
1.3 Batasan Masalah	7
1.4 Tujuan Penelitian.....	7
1.5 Manfaat Penelitian.....	8
BAB 2 TINJAUAN PUSTAKA	9
2.1 Tinjauan Pustaka.....	9
2.2 Keaslian Penelitian	16
2.3 Landasan Teori	21
2.3.1 Penyakit daun padi.....	21

2.3.2 Deep Learning.....	23
2.3.3 Convolutional Neural Network (CNN).....	24
2.3.4 Transfer Learning.....	25
2.3.5 Arsitektur InceptionV3	27
2.3.6 Fine Tuning.....	29
2.3.7 Evaluasi Model Klasifikasi.....	30
BAB 3 METODE PENELITIAN	32
3.1 Jenis, Sifat, dan Pendekatan Penelitian.....	32
3.2 Metode Pengumpulan Data.....	32
3.3 Metode Analisis Data.....	33
3.4 Alur Penelitian.....	35
BAB 4 HASIL PENELITIAN DAN PEMBAHASAN	40
4.1 Persiapan dataset.....	40
4.1.1 Pengumpulan dataset	40
4.1.2 Prapemrosesan Data.....	43
4.1.3 Split Dataset.....	44
4.2 Implementasi Model Skenario.....	47
4.2.1 Skenario 1 – Semua Layer Non-trainable.....	48
4.2.2 Skenario 2 – Fine Tuning Layer Awal.....	50
4.2.3 Skenario 3 – Fine-Tuning Layer Akhir.....	52
4.2.4 Skenario 4 – Full Fine Tuning (Semua Layer Trainable).....	55
4.3 Evaluasi Performa Model Setiap Skenario	57
4.3.1 Evaluasi Skenario 1 Semua Layer Non-trainable	57

4.3.2 Evaluasi Skenario 2 – Fine Tuning Layer Awal	71
4.3.3 Evaluasi Skenario 3 – Fine-Tuning Layer Akhir	101
4.3.4 Evaluasi Skenario 4 – Full Fine Tuning (Semua Layer Trainable)	130
4.4 Komparasi Performa Model dan Kesimpulan	142
4.4.1 Perbandingan Performa Antar Skenario	142
BAB 5 PENUTUP	148
5.1 Kesimpulan	148
5.2 Saran	149
DAFTAR PUSTAKA	150



DAFTAR TABEL

Tabel 2.1 Matriks literatur review dan posisi penelitian.....	16
Tabel 4.1 Detail dataset penyakit daun padi.....	41
Tabel 4.2 Konfigurasi Skenario Pelatihan Model InceptionV3	48
Tabel 4.3 Akurasi Testing 10 Percobaan Skenario 1 (Optimizer Adam).....	58
Tabel 4.4 Akurasi Testing 10 Percobaan Skenario 1 (Optimizer AdamW).....	62
Tabel 4.5 Akurasi Testing 10 Percobaan Skenario 1 (Optimizer SGD)	67
Tabel 4.6 Akurasi Testing 10 Percobaan Skenario 2 layer 150 (Optimizer Adam)	72
Tabel 4.7 Akurasi Testing 10 Percobaan Skenario 2 layer 250 (Optimizer Adam)	77
Tabel 4.8 Akurasi Testing 10 Percobaan Skenario 2 layer 150 (Optimizer AdamW)	81
Tabel 4.9 Akurasi Testing 10 Percobaan Skenario 2 layer 250 (Optimizer AdamW)	86
Tabel 4.10 Akurasi Testing 10 Percobaan Skenario 2 layer 150 (Optimizer SGD).....	92
Tabel 4.11 Akurasi Testing 10 Percobaan Skenario 2 layer 250 (Optimizer SGD).....	96
Tabel 4.12 Akurasi Testing 10 Percobaan Skenario 3 layer 150 (Optimizer Adam).....	103
Tabel 4.13 Akurasi Testing 10 Percobaan Skenario 3 layer 250 (Optimizer Adam).....	107
Tabel 4.14 Akurasi Testing 10 Percobaan Skenario 3 layer 150 (Optimizer AdamW) ..	112
Tabel 4.15 Akurasi Testing 10 Percobaan Skenario 3 layer 250 (Optimizer AdamW) ..	116
Tabel 4.16 Akurasi Testing 10 Percobaan Skenario 3 layer 150 (Optimizer SGD).....	121
Tabel 4.17 Akurasi Testing 10 Percobaan Skenario 3 layer 250 (Optimizer SGD).....	125
Tabel 4.18 Akurasi Testing 10 Percobaan Skenario 4 (Optimizer Adam).....	130
Tabel 4.19 Akurasi Testing 10 Percobaan Skenario 4 (Optimizer AdamW).....	135
Tabel 4.20 Akurasi Testing 10 Percobaan Skenario 4 (Optimizer SGD)	138
Tabel 4.21 Hasil Perbandingan Skenario Fine Tuning Optimizer	143
Tabel 4.22 Perbandingan dengan Penelitian Sebelumnya pada Dataset yang Sama.....	146

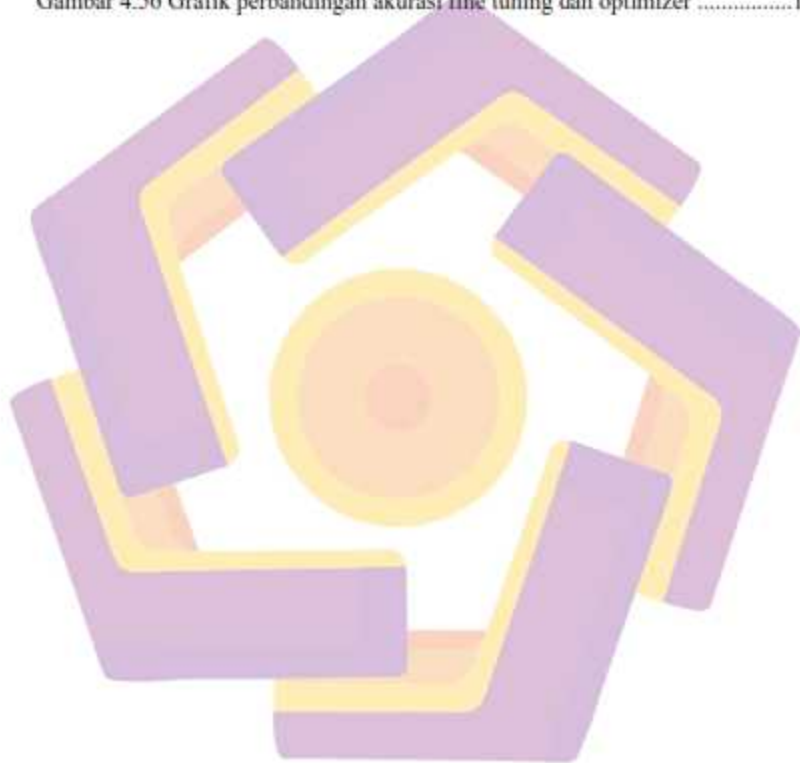
DAFTAR GAMBAR

Gambar 2.1 Arsitektur CNN	25
Gambar 2.2 Arsitektur InceptionV3.....	28
Gambar 2.3 Fine-Tuning of CNN Architecture	30
Gambar 3.1 Alur Penelitian	35
Gambar 4.1 Pengelompokan data penyakit daun padi.....	46
Gambar 4.2 Grafik Training Val Acc dan Loss Skenario 1 (Adam)	59
Gambar 4.3 Confusion matrix Skenario 1 (Adam).....	60
Gambar 4.4 Grafik Training Val Acc Loss Skenario 1 (AdamW)	63
Gambar 4.5 Confusion matrix Skenario 1 (AdamW).....	64
Gambar 4.6 Grafik Training Val Acc dan Loss Skenario 1 (SGD)	68
Gambar 4.7 Confusion matrix Skenario 1 (SGD).....	69
Gambar 4.8 Grafik Layer 150 Training Val Acc dan Loss Skenario 2 (Adam)....	73
Gambar 4.9 Confusion matrix 150 Layer Skenario 2 (Adam).....	74
Gambar 4. 10 Grid-CAM misclassification kelas Brown spot: (a) Leaf smut, (b) Bacterial leaf blight.	75
Gambar 4.11 Citra daun padi dengan true label Leaf Smut.....	76
Gambar 4.12 Grafik Layer 250 Training Val Acc dan Loss Skenario 2 (Adam)..	78
Gambar 4.13 Confusion matrix 250 Layer Skenario 2 (Adam).....	79
Gambar 4.14 Grid-CAM misclassification Bacterial leaf blight	80
Gambar 4.15 Grafik Layer 150 Training Val Acc dan Loss Skenario 2 (AdamW)	82
Gambar 4.16 Confusion matrix 150 Layer Skenario 2 (AdamW).....	84

Gambar 4.17 Grid-CAM misclassification: (a) Bacterial leaf blight, (b) Bacterial leaf blight, (c) Brown spot, (d) Leaf smut.	85
Gambar 4.18 Citra daun padi dengan true label Leaf Smut.	86
Gambar 4.19 Grafik Layer 250 Training Val Acc dan Loss Skenario 2 (AdamW)	87
Gambar 4.20 Confusion matrix 250 Layer Skenario 2 (AdamW).....	89
Gambar 4.21 Grid-CAM misclassification: (a) Brown spot, (b) Leaf smut	90
Gambar 4.22 Citra daun padi dengan true label Leaf Smut.....	91
Gambar 4.23 Grafik Layer 150 Training Val Acc dan Loss Skenario 2 (SGD)....	93
Gambar 4.24 Confusion matrix 150 Layer Skenario 2 (SGD)	94
Gambar 4.25 Grid-CAM misclassification: (a) Bacterial leaf blight, (b) Bacterial leaf blight, (c) Brown Spot, (d) Brown spot, (e) Brown spot, (f) Leaf smut.....	95
Gambar 4.26 Grafik Layer 250 Training Val Acc dan Loss Skenario 2 (SGD)....	97
Gambar 4.27 Confusion matrix 250 Layer Skenario 2 (SGD)	99
Gambar 4.28 Grid-CAM misclassification: (a) Bacterial leaf blight, (b) Brown spot, (c) Brown spot.....	100
Gambar 4.29 Citra daun padi dengan true label Leaf Smut.....	101
Gambar 4.30 Grafik Layer 150 Training Val Acc dan Loss Skenario 3 (Adam)103	
Gambar 4.31 Confusion matrix 150 Layer Skenario 3 (Adam).....	105
Gambar 4.32 Grid-CAM kesalahan klasifikasi: (a) Bacterial leaf blight, (b) Brown spot.....	106
Gambar 4.33 Grafik Layer 250 Training Val Acc dan Loss Skenario 3 (Adam)108	

Gambar 4.34 Confusion matrix 250 Layer Skenario 3 (Adam).....	109
Gambar 4.35 Grid-CAM misclassification Brown spot.....	110
Gambar 4.36 Grafik Layer 150 Training Val Acc, Loss Skenario 3 (AdamW) ..	113
Gambar 4.37 Confusion matrix 150 Layer Skenario 3 (AdamW).....	114
Gambar 4.38 Grid-CAM misclassification Brown spot.....	115
Gambar 4.39 Grafik Layer 250 Training Val Acc dan Loss Skenario 3 (AdamW)	117
Gambar 4.40 Confusion matrix 250 Layer Skenario 3 (AdamW).....	118
Gambar 4.41 Grid-CAM misclassification Brown spot.....	119
Gambar 4.42 Grafik Layer 150 Training Val Acc dan Loss Skenario 3 (SGD) ..	122
Gambar 4.43 Confusion matrix 150 Layer Skenario 3 (SGD)	123
Gambar 4.44 Grid-CAM misclassification: (a) Bacterial leaf blight, (b) Brown spot, (c) Brown spot, (d) Brown spot, (e) Leaf smut.....	124
Gambar 4.45 Grafik Layer 250 Training Val Acc dan Loss Skenario 3 (SGD) ..	126
Gambar 4.46 Confusion matrix 250 Layer Skenario 3 (SGD)	128
Gambar 4.47 Grid-CAM misclassification: (a) Bacterial leaf blight, (b) Brown spot, (c) Brown spot.....	129
Gambar 4.48 Grafik Training Val Acc dan Loss Skenario 4 (Adam)	131
Gambar 4.49 Confusion matrix Skenario 4 (Adam).....	132
Gambar 4.50 Grid-CAM misclassification: (a) Bacterial leaf blight, (b) Brown spot.....	134
Gambar 4.51 Grafik Training Val Acc dan Loss Skenario 4 (AdamW).....	135
Gambar 4.52 Confusion matrix Skenario 4 (AdamW)	136

Gambar 4.53 Grafik Training Val Acc dan Loss Skenario 4 (SGD)	139
Gambar 4.54 Confusion matrix Skenario 4 (SGD).....	140
Gambar 4.55 Grid-CAM misclassification: (a) Bacterial leaf blight, (b) Brown spot, (c) Brown spot, (d) Brown spot.	142
Gambar 4.56 Grafik perbandingan akurasi fine tuning dan optimizer	144



INTISARI

Penyakit daun padi menjadi salah satu penyebab utama penurunan produktivitas pertanian karena mengganggu proses fotosintesis tanaman. Kondisi ini menuntut sistem identifikasi yang akurat dan konsisten agar penyakit dapat dideteksi lebih dini. Penelitian ini bertujuan meningkatkan akurasi klasifikasi penyakit daun padi pada dataset berukuran terbatas melalui penerapan fine tuning pada arsitektur Convolutional Neural Network InceptionV3. Variasi yang dikaji meliputi konfigurasi layer trainable, yaitu non trainable, fine tuning parsial pada layer awal dan layer akhir, serta full fine tuning, dengan penggunaan optimizer Adam, AdamW, dan SGD.

Hasil penelitian menunjukkan bahwa fine tuning berpengaruh signifikan terhadap peningkatan akurasi. Model dengan seluruh layer non trainable menghasilkan akurasi terendah sebesar 70%. Fine tuning parsial pada layer awal dan layer akhir meningkatkan akurasi secara bertahap, dengan hasil terbaik diperoleh pada fine tuning layer akhir menggunakan optimizer Adam di 250 layer trainable yang mencapai akurasi 96,66%. Optimizer AdamW dan SGD menunjukkan performa yang lebih rendah dibandingkan Adam pada konfigurasi terbaiknya.

Berdasarkan hasil tersebut, dapat disimpulkan bahwa fine tuning parsial pada layer akhir dengan 250 layer trainable pada arsitektur InceptionV3 merupakan pendekatan paling efektif untuk meningkatkan akurasi klasifikasi penyakit daun padi pada dataset berukuran terbatas, karena mampu menyesuaikan representasi fitur tingkat tinggi tanpa mengganggu fitur dasar yang telah dipelajari oleh model pretrained. Selain itu, optimizer Adam terbukti menjadi optimizer yang paling tepat karena menghasilkan akurasi tertinggi dan pembaruan bobot yang stabil selama proses pelatihan.

Kata kunci: InceptionV3, fine tuning, klasifikasi, penyakit daun padi, citra digital

ABSTRACT

Rice leaf disease is one of the main factors contributing to the decline in agricultural productivity, as it disrupts the photosynthesis process of the plant. This condition necessitates an accurate and consistent identification system to enable early disease detection. This study aims to improve the accuracy of rice leaf disease classification on a limited-size dataset through the application of fine-tuning on the InceptionV3 Convolutional Neural Network architecture. The variations examined include different trainable layer configurations, namely non-trainable layers, partial fine-tuning on early layers and final layers, as well as full fine-tuning, using the Adam, AdamW, and SGD optimizers.

The results indicate that fine-tuning has a significant impact on improving classification accuracy. The model with all layers set as non-trainable achieved the lowest accuracy of 70%. Partial fine-tuning on both early and final layers resulted in a gradual improvement in accuracy, with the best performance obtained from partial fine-tuning on the final layers using the Adam optimizer with 250 trainable layers, achieving an accuracy of 96.66%. The AdamW and SGD optimizers demonstrated lower performance compared to Adam under the same optimal configuration.

Based on these findings, it can be concluded that partial fine-tuning on the final layers with 250 trainable layers in the InceptionV3 architecture is the most effective approach for improving rice leaf disease classification accuracy on limited-size datasets, as it enables the adaptation of high-level feature representations without disrupting the fundamental features learned by the pretrained model. Furthermore, the Adam optimizer proved to be the most suitable optimizer, as it produced the highest accuracy and maintained stable weight updates throughout the training process.

Keyword: InceptionV3, fine tuning, classification, rice leaf disease, digital images

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Padi (*Oryza sativa*) merupakan salah satu komoditas terpenting dalam pertanian, baik di Indonesia maupun global, karena menjadi bahan pangan pokok bagi lebih dari setengah populasi dunia [1], [2]. Produksi padi sangat rentan menurun akibat serangan berbagai penyakit pada daun yang kerap sulit dikenali sejak dini [3], [4]. Jika infeksi pada daun padi tidak terdeteksi dan ditangani secara cepat, hal ini dapat menyebabkan penurunan hasil panen yang drastis dan mengganggu ketahanan pangan nasional [5]. Bahkan menurut International Rice Research Institute (IRRI), serangan penyakit daun padi dapat memangkas hasil panen hingga 80%, berdampak pada menurunnya pendapatan petani, berkurangnya ketersediaan beras, dan meningkatnya harga pangan [1].

Beberapa penyakit yang umum menyerang daun padi antara lain Bacterial Leaf Blight (BLB), Brown Spot (BS), dan Leaf Smut (LS). Ketiga penyakit ini menjadi ancaman utama dalam sistem budidaya padi di berbagai wilayah penghasil beras. Bacterial Leaf Blight disebabkan oleh bakteri *Xanthomonas oryzae* pv. *oryzae* yang menyerang jaringan daun dan dapat menurunkan hasil panen secara signifikan karena menghambat proses fotosintesis. Brown Spot yang disebabkan oleh jamur *Bipolaris oryzae* dapat mengurangi kualitas gabah dan menyebabkan bercak coklat pada daun, terutama pada kondisi kekurangan nutrisi atau cuaca lembap. Sementara itu, Leaf Smut yang disebabkan oleh jamur *Entyloma oryzae*

menyerang permukaan daun dan merusak jaringan tanaman sehingga mengurangi produktivitas [4], [6]–[8]. Identifikasi penyakit secara visual masih sulit dilakukan karena kemiripan gejala, sehingga dibutuhkan teknologi berbasis citra digital yang dapat mendiagnosis secara cepat dan akurat.

Selama ini, identifikasi penyakit pada daun padi umumnya dilakukan secara manual melalui inspeksi visual oleh petani. Metode tradisional ini memiliki berbagai keterbatasan: prosesnya lambat, memerlukan tenaga ahli, dan hasil pengamatannya bersifat subyektif [9]. Gejala beberapa penyakit daun padi pun sering kali mirip satu sama lain, sehingga menyulitkan petani untuk memastikan jenis penyakit yang menyerang tanamannya. Akibatnya, diagnosis bisa terlambat atau keliru, dan tindakan penanganan tidak optimal. Di sisi lain, pendekatan manual tidak skalabel untuk areal sawah yang luas. Keterbatasan ini mendorong perlunya solusi otomatis yang dapat mendeteksi penyakit secara cepat dan konsisten [1]. Oleh sebab itu, deteksi dan klasifikasi penyakit daun padi yang akurat sejak awal memiliki peran krusial dalam mencegah kerugian hasil panen dan memastikan produktivitas pertanian tetap terjaga.

Beberapa pendekatan pembelajaran mesin seperti Support Vector Machine, K-Nearest Neighbor, Decision Tree, dan Random Forest telah banyak digunakan dalam tugas klasifikasi penyakit tanaman berbasis citra [10]–[13]. Pendekatan ini telah diterapkan pada berbagai jenis tanaman seperti padi, tomat, dan jagung, dan berhasil mengklasifikasikan beberapa jenis penyakit. Diantaranya, Studi [10] yang mengklasifikasikan penyakit daun tomat menunjukkan bahwa metode seperti SVM, KNN dan Random Forest berhasil mencapai akurasi masing-masing sekitar 98,8 %,

98,5 % dan 98,8 % dalam klasifikasi daun tomat sehat vs sakit. Namun, akurasi model cenderung turun ketika citra memiliki perubahan cahaya, latar belakang tidak seragam, atau terdapat noise visual [14]. Selanjutnya, Studi [11] yang mengklasifikasikan penyakit daun padi, penelitian berbasis machine learning tradisional seperti KNN, SVM, Random Forest, dan Decision Tree mencapai akurasi sekitar 94% dengan nilai ROC-AUC 0,92 pada Random Forest. Model-model ini masih mampu mengklasifikasikan jenis penyakit daun padi. Namun, akurasinya cenderung menurun ketika jumlah kelas lebih banyak, tekstur daun lebih beragam, atau citra diambil pada kondisi lapangan yang tidak stabil. Studi oleh [13] mengklasifikasikan daun jagung menggunakan empat model pembelajaran mesin, yaitu Decision Tree, SVM, KNN, dan Artificial Neural Network. Di antara model tersebut, ANN mencapai akurasi tertinggi sebesar 92,7%. SVM mencapai akurasi sekitar 91,3%. KNN dan Decision Tree masing-masing mencapai 90%. Seluruh model berhasil mengklasifikasikan daun sehat, gray leaf spot, dan common rust dengan cukup baik. Namun, performanya masih bergantung pada fitur manual dan dapat menurun ketika citra memiliki karakter visual yang lebih kompleks, sehingga model kurang stabil pada kondisi lapangan yang bervariasi [15]. Keterbatasan tersebut mendorong penggunaan pendekatan deep learning seperti Convolutional Neural Network (CNN), yang dapat mengekstraksi fitur secara otomatis dari citra berukuran besar dan menunjukkan performa yang lebih konsisten dalam menangani variasi visual yang kompleks di kondisi lapangan [16].

Dari beberapa penelitian yang sudah disampaikan di atas tentang berbagai penyakit pada daun tanaman. Penelitian ini berfokus pada penyakit daun padi.

Dalam pengembangan sistem klasifikasi penyakit daun padi berbasis citra, sejumlah penelitian telah membandingkan berbagai arsitektur Convolutional Neural Network yang berfokus pada evaluasi performa akurasi. Diantaranya, Studi [17] menggunakan dataset 9 kelas penyakit daun padi dengan total 5.932 citra. Penelitian ini membandingkan tiga model yaitu CNN, CAAR U-Net, dan MobileNetV2. Dari hasil eksperimen, model CNN memberikan akurasi tertinggi sebesar 98%. Meskipun performanya tinggi, penelitian ini belum menguji model pada kondisi dataset terbatas sehingga belum terlihat bagaimana stabilitas akurasinya saat jumlah citra berkurang. Selanjutnya, Studi oleh [18] menggunakan dataset 5.932 citra daun padi yang terdiri dari empat kelas penyakit dan membandingkan empat model CNN pra-latih, yaitu InceptionResNetV2, MobileNetV2, DenseNet121, dan VGG16. Dari hasil eksperimen, DenseNet121 memperoleh akurasi tertinggi sebesar 99,10%, diikuti MobileNetV2 dengan akurasi 97,10%. Penelitian ini menunjukkan performa tinggi pada kondisi data yang besar, namun model belum diuji pada kondisi data yang terbatas sehingga belum menggambarkan performa ketika jumlah citra jauh lebih sedikit. Studi oleh [19] menggunakan dataset 900 citra dengan membandingkan enam arsitektur CNN seperti DenseNet121, InceptionV3, MobileNetV2, ResNeXt101, ResNet152V, dan SE-ResNeXt101. Dari hasil eksperimen akurasi yang diperoleh tertinggi yaitu InceptionV3 dengan akurasi 97%. Namun, performa penelitian tersebut masih dapat ditingkatkan karena ketergantungan pada augmentasi untuk memperbesar dataset dan belum adanya evaluasi pada dataset kecil. Selanjutnya, Studi oleh [20] yang membandingkan InceptionV3, InceptionResNetV2, EfficientNetB3,

ResNet50, dan VGG19 pada tiga dataset berukuran besar, sedang, dan kecil. Dari hasil penelitian didapatkan bahwa model InceptionV3 menghasilkan akurasi tertinggi sebesar 99,53%, lebih unggul dibandingkan empat model lainnya. Pada dataset terbatas, akurasi InceptionV3 turun menjadi 57,11%, dan pada dataset terbatas model InceptionV3 akurasi tertinggi diantara model yang lainnya dengan akurasi 90,00%. Namun, performa dari penelitian tersebut masih dapat ditingkatkan karena akurasi menurun signifikan pada dataset kecil dan belum dilakukan optimasi lanjutan seperti fine tuning. Hal ini menunjukkan bahwa model masih bisa ditingkatkan lagi, terutama untuk menjaga kestabilan dan akurasi saat data yang tersedia terbatas.

Berbagai penelitian menunjukkan bahwa fine-tuning mampu meningkatkan performa model secara signifikan, terutama ketika dataset berukuran kecil atau memiliki variasi visual yang tinggi. Penelitian lain oleh [21] menunjukkan bahwa fine-tuning pretrained model juga efektif, di mana VGG16 yang di-fine-tune mencapai akurasi tertinggi sebesar 99,3%, lebih tinggi dibandingkan versi yang hanya menggunakan layer-layer non-trainable. Selanjutnya, Studi [22] yang menegaskan bahwa fine-tuning layer tertentu berperan penting dalam memaksimalkan akurasi dan meningkatkan kinerja model, terutama ketika dataset memiliki keragaman yang tinggi. Hasil-hasil ini memperlihatkan bahwa fine-tuning tidak hanya meningkatkan akurasi, tetapi juga membuat model lebih stabil terhadap perubahan fitur visual. Oleh karena itu, pendekatan ini relevan diterapkan pada InceptionV3 untuk meningkatkan kemampuan model dalam mengenali penyakit daun padi pada dataset berskala kecil.

Berdasarkan berbagai penelitian sebelumnya, InceptionV3 mampu memberikan akurasi yang baik dalam klasifikasi penyakit daun padi, tetapi performanya masih berubah-ubah bergantung pada jumlah data, variasi citra, dan pengaturan parameter yang digunakan. Beberapa model juga mengalami penurunan akurasi ketika diuji pada dataset yang kecil atau kondisi visual yang berbeda dari data pelatihan. Kondisi ini menunjukkan bahwa model masih membutuhkan penyesuaian lebih lanjut agar bekerja lebih stabil. Oleh karena itu, penelitian ini berfokus pada penerapan teknik fine-tuning model InceptionV3 untuk klasifikasi penyakit daun padi di dataset berskala kecil. Dengan pendekatan ini diharapkan dapat meningkatkan akurasi, memperbaiki kemampuan generalisasi, dan menghasilkan model yang lebih konsisten pada dataset yang terbatas maupun bervariasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah penelitian ini adalah:

1. Bagaimana pengaruh penerapan fine-tuning parsial dengan menjadikan beberapa layer awal dan beberapa layer akhir sebagai trainable, serta full fine-tuning, terhadap akurasi klasifikasi penyakit daun padi menggunakan model InceptionV3 pada dataset berukuran terbatas?
2. Optimizer mana di antara Adam, AdamW, dan SGD yang paling efektif dalam meningkatkan akurasi klasifikasi penyakit daun padi pada proses fine-tuning model InceptionV3 dengan dataset berukuran terbatas?

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

- 1) Penelitian ini hanya membahas klasifikasi penyakit daun padi, yaitu Bacterial Leaf Blight, Brown Spot, dan Leaf Smut. Dataset di ambil dari UCI Machine Learning Repository [23].
- 2) Penelitian ini berfokus pada analisis pengaruh strategi fine-tuning pada model InceptionV3 terhadap performa klasifikasi penyakit daun padi dengan menggunakan satu dataset yang sama.
- 3) Model yang digunakan arsitektur InceptionV3 dengan pendekatan transfer learning.
- 4) Penelitian ini berfokus pada analisis akurasi model, tanpa membahas implementasi sistem deteksi secara real-time atau berbasis perangkat keras.
- 5) Penelitian ini hanya memfokuskan pada deteksi dan klasifikasi penyakit daun padi, tidak mencakup analisis kondisi tanaman secara keseluruhan atau deteksi penyakit dari bagian lain seperti batang atau akar.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut :

1. Menganalisis pengaruh penerapan fine-tuning parsial pada beberapa layer awal dan beberapa layer akhir, serta full fine-tuning, terhadap akurasi klasifikasi penyakit daun padi pada model InceptionV3 dengan dataset berukuran terbatas.

2. Menganalisis kinerja optimizer Adam, AdamW, dan SGD pada proses fine-tuning model InceptionV3 untuk klasifikasi penyakit daun padi berdasarkan akurasi pada dataset berukuran terbatas.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini mencakup:

Manfaat Teoritis:

1. Memberikan kontribusi terhadap pengembangan ilmu pengetahuan di bidang deep learning, khususnya pada penerapan teknik fine-tuning dalam arsitektur InceptionV3 untuk klasifikasi citra tanaman.
2. Menyajikan hasil analisis tentang peran fine-tuning dan optimizer dalam meningkatkan kemampuan model mengenali penyakit daun padi.

Manfaat Praktis:

1. Menyediakan panduan praktis yang dapat direproduksi untuk implementasi sistem deteksi penyakit padi otomatis dengan performa optimal.
2. Membantu praktisi dan peneliti dalam memilih konfigurasi model yang tepat untuk deployment pada kondisi data terbatas.
3. Mendukung pengembangan sistem precision agriculture yang dapat meningkatkan produktivitas dan ketahanan pangan nasional.
4. Menyediakan dasar bagi pengembangan aplikasi berbasis AI yang dapat digunakan petani atau instansi pertanian untuk identifikasi dini penyakit tanaman secara otomatis.

BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Tinjauan pustaka penelitian ini berfokus pada pengembangan metode klasifikasi penyakit daun padi pada dataset berukuran kecil. Dataset Rice Leaf Diseases dari UCI menjadi acuan utama karena terdiri dari 120 citra tiga kelas sehingga sering digunakan untuk menguji kinerja model pada kondisi data terbatas. Tantangan utama pada dataset kecil adalah kurangnya variasi visual dan tingginya kemungkinan overfitting. Oleh karena itu, banyak penelitian sebelumnya mencoba pendekatan segmentasi, ekstraksi fitur, atau penggunaan model pretrained, namun belum ada yang benar-benar mengkaji fine tuning InceptionV3 secara sistematis.

Penelitian awal dilakukan oleh [24] yang memperkenalkan dataset ini. Mereka menerapkan segmentasi K-means untuk memisahkan area penyakit sebelum mengekstraksi fitur manual. Meskipun segmentasi membantu menonjolkan area penting, pendekatan ini masih bergantung pada fitur hand-crafted yang kurang mampu menangkap pola kompleks. Akurasi yang diperoleh hanya 73,33%. Hasil ini memperlihatkan keterbatasan metode manual untuk citra daun padi yang memiliki variasi warna, tekstur, dan bentuk yang tinggi.

Studi oleh [25] merancang metode ekstraksi fitur untuk mendeteksi penyakit daun padi pada dataset kecil menggunakan pendekatan berbasis warna dan tekstur. Mereka tidak memakai arsitektur pretrained, tetapi menerapkan background removal dengan threshold saturasi, disusul segmentasi kanal hue untuk

menonjolkan area bercak penyakit. Peneliti tersebut menguji berbagai kombinasi fitur warna, tekstur, dan bentuk untuk mendapatkan representasi citra yang paling stabil. Penelitian tersebut akhirnya memilih kombinasi fitur gabungan yang disesuaikan dengan pola warna masing-masing penyakit. Pada penelitian tersebut tidak digunakan augmentasi, sehingga seluruh proses bergantung pada variasi data asli. Model diuji menggunakan beberapa algoritma pembelajaran mesin, dan akurasi terbaik mencapai 86,58%. Penelitian ini menunjukkan bahwa fitur warna dan tekstur dapat meningkatkan performa, namun pendekatan tersebut tetap dibatasi kapasitas generalisasi karena tidak memanfaatkan kelebihan feature extraction mendalam seperti model CNN pretrained.

Peningkatan signifikan mulai terlihat ketika CNN pretrained diterapkan. Studi oleh [26] berbasis CNN pretrained untuk mendeteksi tiga penyakit daun padi pada dataset terbatas. Mereka menggunakan arsitektur VGG16, ResNet50, dan DenseNet121 melalui transfer learning, lalu menambahkan segmentasi lokal berbasis kanal merah sebelum pelatihan. Peneliti tersebut menguji berbagai kombinasi preprocessing, jumlah dense layer, ukuran layer fully connected, serta konfigurasi aktivasi untuk meningkatkan stabilitas model. Penelitian tersebut akhirnya memilih arsitektur DenseNet121 dengan dua tambahan fully connected layer (1000 dan 500 neuron) serta segmentasi lokal sebagai konfigurasi terbaik. Pada penelitian tersebut augmentasi dilakukan untuk memperbesar variasi data, namun tetap dalam batas standar seperti rotasi dan flipping. Model diuji menggunakan pembagian data latih-validasi-uji, dan akurasi tertinggi mencapai 91,67%. Penelitian ini menunjukkan bahwa kombinasi segmentasi berbasis nilai

merah dan transfer learning dapat bekerja baik pada dataset terbatas, tetapi pelatihan hanya dilakukan pada layer atas sehingga belum mengeksplor kemampuan fine tuning yang lebih dalam.

Penelitian lain oleh [27] pendekatan berbasis transfer learning VGG16 untuk mendeteksi penyakit daun padi dengan bantuan segmentasi K-means. Mereka memakai arsitektur pretrained tetapi tetap memanfaatkan teknik segmentasi untuk memperjelas area bercak sebelum pelatihan. Peneliti tersebut menguji beberapa variasi jumlah cluster, optimizer, learning rate, serta nilai dropout untuk mencari pengaturan yang paling stabil. Penelitian tersebut akhirnya memilih kombinasi segmentasi K-means, optimizer RMSProp, dan learning rate 0.001 sebagai konfigurasi paling optimal. Pada penelitian tersebut augmentasi standar diterapkan pada data latih, dan dataset dibagi menjadi 70 persen latih-validasi serta 30 persen uji. Model diuji dalam beberapa skenario dan akurasi tertinggi mencapai 91,66%. Penelitian ini menunjukkan bahwa segmentasi dan hyperparameter tuning dapat meningkatkan performa, tetapi penggunaan VGG16 membatasi kemampuan representasi fitur ketika dataset sangat kecil dan kompleks.

Metode berbasis fitur manual tetap dikembangkan oleh [3] merancang pendekatan ekstraksi fitur tekstur menggunakan GLCM untuk mendeteksi penyakit daun padi pada dataset terbatas. Mereka tidak menggunakan model pretrained, tetapi memanfaatkan Canny Edge Detection untuk menonjolkan pola tepi daun sebelum ekstraksi fitur. Peneliti tersebut menguji berbagai kombinasi arah GLCM, jarak piksel, dan jumlah tetangga K pada KNN untuk mendapatkan konfigurasi yang paling stabil. Penelitian tersebut akhirnya memilih konfigurasi GLCM optimal

dan nilai $K = 7$ dengan aktivasi Canny Edge Detection. Pada penelitian tersebut data tidak diaugmentasi sehingga proses sangat bergantung pada kualitas fitur tekstur asli. Model diuji dengan pembagian data 80:20 dan akurasi terbaik mencapai 91,67%. Pendekatan ini menunjukkan bahwa fitur tekstur masih dapat menghasilkan akurasi kompetitif pada dataset kecil, tetapi tidak memiliki representasi abstrak yang biasanya dihasilkan model CNN pretrained.

Studi oleh [28] LW17 sebagai model CNN ringan yang ditujukan khusus untuk mendeteksi tiga penyakit daun padi pada dataset berukuran kecil. Mereka tidak menggunakan arsitektur pretrained, tetapi membangun model dari awal dengan 17 layer. Peneliti tersebut menguji berbagai variasi konfigurasi pelatihan seperti jumlah layer, jenis pooling, optimizer, learning rate, jumlah epoch, dan rasio train-validation. Penelitian tersebut akhirnya memilih kombinasi yang paling stabil, yaitu max pooling, optimizer Adam, learning rate 0.001, dan rasio 9:1 antara data latih dan validasi. Pada penelitian tersebut melakukan augmentasi sehingga total data pelatihan menjadi tiga kali lipat dibanding penelitian sebelumnya, yang membantu model lebih stabil pada dataset kecil. Model LW17 diuji 10 kali untuk mendapatkan rata-rata akurasi, dan hasil terbaik mencapai 93.75%. Penelitian ini menunjukkan bahwa arsitektur ringan yang dirancang dengan baik dapat menghasilkan akurasi tinggi pada dataset terbatas. Namun LW17 tetap memiliki batasan karena seluruh fiturnya dipelajari dari nol tanpa memanfaatkan kemampuan feature extraction dari model pretrained seperti InceptionV3 yang sudah terlatih pada jutaan citra. LW17 dapat efisien dari sisi komputasi, tetapi tidak memiliki

transfer pengetahuan yang biasanya membantu performa pada kondisi data yang terbatas.

Studi [20] menguji InceptionV3 dan InceptionResNetV2 pada tiga ukuran dataset penyakit daun padi, termasuk dataset kecil berisi 120 citra yang sekelas dengan dataset UCI. Pada dataset besar, akurasi kedua model relatif tinggi, namun saat jumlah data diperkecil performanya berbeda. InceptionV3 tetap stabil dengan akurasi sekitar 86,67 sampai 90%, sedangkan InceptionResNetV2 turun lebih jauh karena arsitekturnya lebih berat. Firnando melatih model dengan transfer learning standar, yaitu membekukan backbone dan hanya melatih layer atas dengan augmentasi dasar. Hasilnya menunjukkan bahwa InceptionV3 lebih cocok menangani data terbatas karena struktur modul Inception mampu menangkap pola bercak dan tekstur daun tanpa membutuhkan data besar. Namun penelitian tersebut belum mencoba variasi fine tuning yang lebih dalam, misalnya membuka sebagian layer akhir atau seluruh backbone. Bagian ini kemudian menjadi fokus penelitian ini untuk melihat sejauh mana penyesuaian layer pada InceptionV3 dapat meningkatkan akurasi pada dataset kecil penyakit daun padi.

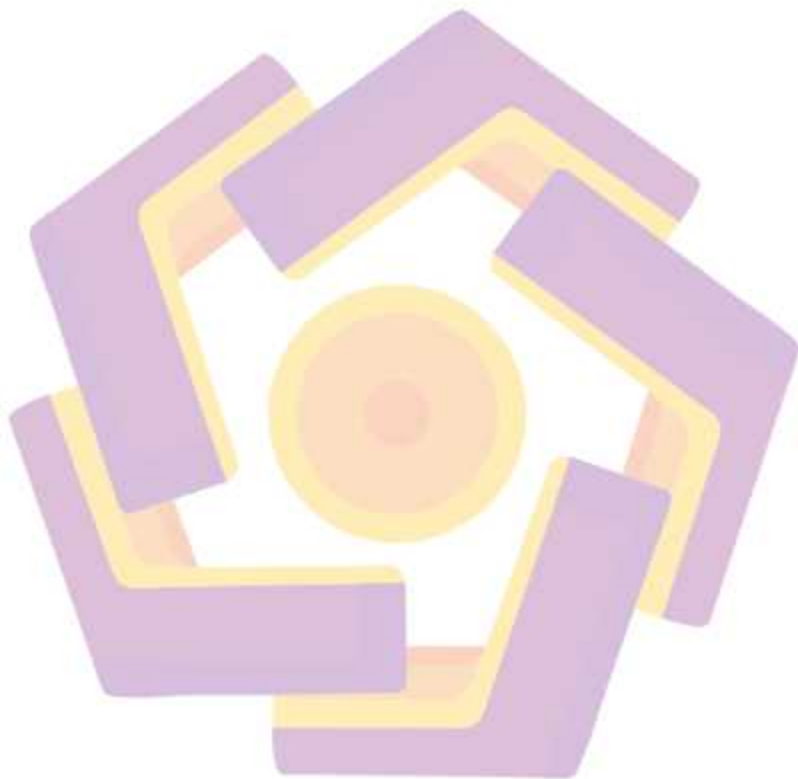
Pada penelitian [29] membahas klasifikasi citra medis menggunakan pendekatan transfer learning dengan arsitektur InceptionV3. Model memanfaatkan bobot awal dari ImageNet dan menerapkan strategi pembekuan layer ekstraksi fitur, sedangkan proses pelatihan difokuskan pada layer klasifikasi. Penelitian ini secara khusus menyoroti peran learning rate dalam menjaga stabilitas pelatihan. Learning rate awal ditetapkan sebesar 0,0001 dan diturunkan secara bertahap ketika validation loss meningkat. Dengan konfigurasi tersebut, model mencapai performa

terbaik dengan akurasi sebesar 99,17%, precision 99,17%, dan recall 99%. Hasil ini menunjukkan bahwa penggunaan learning rate kecil efektif untuk transfer learning pada dataset berukuran terbatas karena mampu mencegah pembaruan bobot yang terlalu agresif.

Penelitian [30] membahas penerapan CNN untuk klasifikasi citra menggunakan arsitektur InceptionV3 pada skenario transfer learning. Model dilatih menggunakan learning rate 0,0001 dan dibandingkan dengan skenario lain yang menggunakan learning rate lebih besar seperti 0,001. Hasil pengujian menunjukkan bahwa learning rate 0,0001 menghasilkan nilai loss yang lebih rendah dan akurasi yang lebih tinggi dibandingkan konfigurasi lainnya. Pada skenario terbaik, model mencapai akurasi sebesar 97%, lebih tinggi dibandingkan penggunaan optimizer dan learning rate yang lebih besar. Penelitian ini menegaskan bahwa learning rate kecil lebih sesuai untuk pelatihan CNN pretrained karena memberikan konvergensi yang lebih stabil.

Hasil dari penelitian sebelumnya menunjukkan bahwa InceptionV3 memiliki potensi yang baik ketika diterapkan pada dataset berukuran kecil. Model ini mampu memberikan performa yang kompetitif dibandingkan arsitektur lain pada kondisi data terbatas, khususnya ketika dipadukan dengan strategi transfer learning dan pengaturan learning rate yang tepat. Meskipun demikian, sebagian besar penelitian masih menerapkan skema pelatihan standar dengan pembekuan backbone dan belum mengkaji pengaruh fine tuning yang lebih dalam secara sistematis. Oleh karena itu, penelitian ini berfokus pada pengujian beberapa variasi fine tuning, meliputi pembekuan seluruh layer, pembukaan sebagian layer akhir,

dan pembukaan seluruh layer InceptionV3, serta membandingkan beberapa optimizer. Pendekatan ini bertujuan untuk mengevaluasi sejauh mana penyesuaian parameter pelatihan dapat meningkatkan performa klasifikasi penyakit daun padi pada dataset yang terbatas.



2.2 Keaslian Penelitian

Tabel 2.1 Matriks literatur review dan posisi penelitian

Peningkatan performa klasifikasi penyakit daun padi pada dataset terbatas menggunakan fine-tuning model cnn inceptionv3

No	Judul Penelitian	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
1	Detection of rice plant disease from RGB and grayscale images using an LW17 deep learning model	Yogesh Kumar Rathore, Rekh Ram Janghel, Chetan Swarup, Saroj Kumar Pandey, Ankit Kumar, Kamred Udham Singh, Teckam Singh (2023), Electronic Research Archive (ERA) – Scopus Q3	CNN ringan (LW17) Dataset UCI Rice Leaf Diseases (120 gambar).	Akurasi 93,75%,	Keunggulan: ringan, cepat, cocok untuk dataset terbatas. Kelemahan: Belum diuji di dengan menggunakan Fine Tuning	Kedua penelitian menggunakan dataset UCI dan berbasis CNN. Perbedaannya, penelitian ini memakai model LW17, sedangkan penelitian saya menggunakan InceptionV3 dengan fine-tuning hyperparameter.
2	An Effective Feature Extraction Method for Rice Leaf Disease Classification	Muhammad Anwarul Azim, Mohammad Khairul Islam, Md. Marufur Rahman, Farah Jahan (2021), TELKOMNIKA	Menggunakan metode ekstraksi fitur warna, bentuk, dan tekstur dari citra daun padi menggunakan segmentasi HSV	Akurasi 86,58%,	Keunggulan: metode sederhana, cukup akurat untuk dataset terbatas. Kelemahan :	Kesamaan dengan penelitian saya ada pada penggunaan dataset UCI Rice Leaf Diseases, sedangkan perbedaannya penelitian saya memakai CNN InceptionV3 dengan fine-tuning

No	Judul Penelitian	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
		(Telecommunication, Computing, Electronics and Control) – Scopus Q2	Saturation dan Hue Threshold pada dataset UCI Rice Leaf Diseases. Klasifikasi dilakukan menggunakan XGBoost dan SVM RBF.		resolusi citra pada dataset tidak seragam.	hyperparameter, bukan metode ekstraksi fitur manual.
3	Rice Leaf Disease Recognition using Local Threshold Based Segmentation and Deep CNN	Anam Islam, Redoun Islam, S. M. Rafizul Haque, S. M. Mohidul Islam, Muhammad Ashik Iqbal Khan (2021), International Journal of Intelligent Systems and Applications (IJISA) – Scopus Q3	Menggunakan CNN (VGG16, ResNet50, DenseNet121) pada tiga dataset termasuk UCI Rice Leaf Diseases.	Segmentasi meningkatkan akurasi dari 83,33% menjadi 91,67%.	Keunggulan : Kombinasi segmentasi dan CNN meningkatkan akurasi deteksi. Kelemahan : Performa menurun untuk dataset berbeda karena variasi sumber.	Kesamaan dengan penelitian saya terdapat pada penggunaan dataset UCI Rice Leaf Diseases dan pendekatan CNN. Perbedaannya, penelitian saya menggunakan InceptionV3 dengan fine-tuning, sedangkan penelitian ini menggunakan DenseNet121 dengan metode segmentasi lokal untuk peningkatan hasil klasifikasi.

No	Judul Penelitian	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
4	The Implementation of Pretrained VGG16 Model for Rice Leaf Disease Classification using Image Segmentation	Jody Ririt Krido Suseno, Yufiz Azhar, Agus Eko Minarno (2023), KINETIK: Game Technology, Information System, Computer Network, Computing, Electronics, and Control – SINTA 2	Model pretrained VGG16, optimizer RMSProp, Adam, dan SGD pada dataset UCI Rice Leaf Diseases.	Akurasi tertinggi 91,66% dengan optimizer RMSProp dan learning rate 0.001 pada segmentasi k-means.	Kelebihan: penggunaan segmentasi meningkatkan akurasi dan efisiensi pemrosesan citra. Kelemahan: hasil masih terbatas pada dataset kecil	Kesamaan dengan penelitian saya terletak pada penggunaan dataset UCI Rice Leaf Diseases dan pendekatan deep learning CNN. Perbedaannya, penelitian ini menggunakan VGG16 pretrained model dengan segmentasi citra, sedangkan penelitian saya menggunakan InceptionV3 dengan fine-tuning hyperparameter untuk meningkatkan performa klasifikasi pada dataset terbatas.
5	Analyzing InceptionV3 and InceptionResNetV2 with Data Augmentation for	Fadel Muhamad Firmando, De Rosal Ignatius Moses Setiadi, Ahmad Rofiqul Muslikh, Syahroni Wahyu Iriananda	Menggunakan InceptionV3 dan InceptionResNetV2 dengan pendekatan transfer learning dan data augmentation.	Akurasi tertinggi diperoleh InceptionV3 yaitu 99,53% (Dataset 1).	Keunggulan: InceptionV3 efektif pada dataset kecil mencapai akurasi tertinggi.	Kesamaan dengan penelitian saya terdapat pada penggunaan dataset UCI Rice Leaf Diseases dan arsitektur InceptionV3. Perbedaannya, penelitian saya berfokus pada

No	Judul Penelitian	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
	Rice Leaf Disease Classification	(2024), Journal of Future Artificial Intelligence and Technologies (FAITH) – Scopus indexed	Eksperimen dilakukan pada tiga dataset publik termasuk UCI Rice Leaf Diseases.	58,94% (Dataset 2), dan 90,00% (Dataset 3).	Kelemahan: Performa tidak stabil pada dataset kecil.	fine-tuning hyperparameter untuk meningkatkan akurasi pada dataset terbatas, sedangkan penelitian ini pengaruh data augmentation terhadap performa model.
6	Klasifikasi Penyakit Daun Padi Menggunakan KNN dengan GLCM dan Canny Edge Detection	Ike Verawati, Ridwan Al Akhyar, Aumurohim (2024), Media Informatika Budidarma – SINTA 2	Klasifikasi penyakit daun padi menggunakan KNN. Ekstraksi fitur menggunakan GLCM. Percobaan dilakukan dengan Canny Edge Detection dan tanpa Canny Edge Detection. Dataset menggunakan UCI Rice Leaf Diseases (120 citra, tiga kelas).	Akurasi tertinggi 91,67	Keunggulan: metode sederhana dan efisien untuk dataset kecil. Kombinasi GLCM dan Canny meningkatkan akurasi. Kelemahan: belum diuji pada kondisi lapangan. Tidak	Penelitian ini menggunakan dataset yang sama. Pendekatan berbeda karena penelitian saya memakai CNN InceptionV3 dengan fine-tuning hyperparameter. Fokus penelitian sebelumnya pada KNN+GLCM+Edge Detection, sedangkan penelitian saya berfokus pada optimasi model deep learning.

No	Judul Penelitian	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
					menggunakan pendekatan deep learning.	
7	Enhancing Skin Cancer Diagnosis Through Fine-Tuning of Pretrained Models: A Two-Phase Transfer Learning Approach	Entesar Hamed I. Eliwa, 2025, International Journal of Breast Cancer – Scopus Q2	VGG16, VGG19, InceptionV3, Xception, DenseNet121, DenseNet201, ResNet152V2, MobileNet, MobileNetV2. Tahap pertama training dengan layer dasar non-trainable. Tahap kedua fine-tuning semua layer.	Akurasi tertinggi setelah fine-tuning ada pada VGG16 sebesar 99.3%. Model lain meningkat signifikan setelah fine-tuning	Kelebihan : Menguji banyak model pretrained. Fine-tuning dibuktikan meningkatkan akurasi. Dataset besar dan valid. Kelemahan : Di uji di dataset besar akurasi bagus, belum di uji untuk dataset terbatas.	Dataset menggunakan penyakit kulit, sedangkan penelitian saya memakai dataset penyakit daun padi. Kesamaannya, kedua penelitian menerapkan teknik fine-tuning untuk meningkatkan akurasi model.

2.3 Landasan Teori

2.3.1 Penyakit daun padi

Penyakit pada daun padi dapat disebabkan oleh berbagai faktor, salah satunya adalah infeksi oleh patogen seperti bakteri, jamur, dan fungi. Beberapa penyakit utama yang menyerang daun padi antara lain Bacterial Leaf Blight (BLB), Leaf Smut, dan Brown Spot. Penyakit-penyakit ini dapat menyebabkan kerugian signifikan terhadap hasil panen dan kualitas padi, sehingga penting untuk memahami penyebab, gejala, dan mekanisme patogen dalam mengatasi masalah tersebut [31].

a) Bacterial Leaf Blight (BLB)

Bacterial Leaf Blight (BLB) pada padi disebabkan oleh bakteri *Xanthomonas oryzae pv. oryzae*. Penyakit ini menyerang daun padi dan dapat menyebabkan hilangnya chlorophyll pada daun, yang mengarah pada penurunan fotosintesis dan hasil panen yang buruk. Seperti yang terlihat pada Gambar 2.1, gejala BLB dimulai dengan munculnya bercak-bercak air pada daun yang kemudian berkembang menjadi noda kuning terang dan mengering. Penularan BLB terutama terjadi melalui air dan dapat diperparah oleh kelembapan tinggi serta cuaca lembap. Pengendalian dapat dilakukan dengan menggunakan varietas padi yang tahan terhadap bakteri ini serta dengan mengatur irigasi yang tepat. Berikut ini penyakit pada daun padi Bacterial leaf blight [32] :



Gambar 2.1 Bacterial leaf blight

b) Leaf Smut

Leaf Smut adalah penyakit yang disebabkan oleh jamur *Ustilagoidea virens*. Penyakit ini ditandai dengan pembentukan bercak-bercak hitam atau struktur mirip spora pada permukaan daun, seperti yang terlihat pada Gambar 2.2, yang mengganggu proses fotosintesis. Jamur ini terutama menyerang daun muda dan membentuk struktur spora di dalam jaringan daun yang terinfeksi. Leaf Smut dapat menurunkan hasil padi secara signifikan, terutama jika infeksi terjadi pada stadium awal tanaman. Pengendalian dilakukan dengan penggunaan benih yang bebas dari patogen dan penerapan fungisida yang efektif.



Gambar 2.2 Leaf Smut

c) Brown Spot

Brown Spot pada padi disebabkan oleh jamur *Bipolaris oryzae*, yang menginfeksi daun dan menyebabkan munculnya bercak coklat dengan batas yang jelas. Penyakit ini terutama terjadi pada tanaman yang sudah

tua atau yang mengalami kondisi stres, seperti kekurangan air atau nutrisi, seperti yang terlihat pada Gambar 2.3. Gejala utama dari Brown Spot adalah adanya bercak coklat besar yang dapat berkembang menjadi lubang pada daun. Jika tidak dikendalikan, penyakit ini dapat menyebabkan penurunan hasil padi yang cukup besar. Pengendalian dapat dilakukan dengan rotasi tanaman, penggunaan varietas tahan, serta pengendalian dengan fungisida yang sesuai.



Gambar 2.3 Brown Spot

2.3.2 Deep Learning

Deep Learning merupakan cabang khusus dari kecerdasan buatan (Artificial Intelligence) yang berfokus pada penggunaan artificial neural network (ANN) berlapis-lapis (multi-layered). Istilah “deep” mengacu pada banyaknya layer tersembunyi (hidden layers) dalam jaringan neural; jika jaringan neural tradisional umumnya hanya memiliki 1–3 layer tersembunyi, maka deep neural network dapat memiliki puluhan bahkan ratusan layer. Dengan arsitektur yang dalam ini, model deep learning mampu mempelajari representasi fitur secara hierarkis dari data masif [33]. Berbeda dengan metode machine learning klasik yang memerlukan desain fitur manual oleh pakar, deep learning secara otomatis mengekstraksi ciri-ciri/pola dari data mentah selama proses pelatihan. Hal ini dimungkinkan karena setiap layer

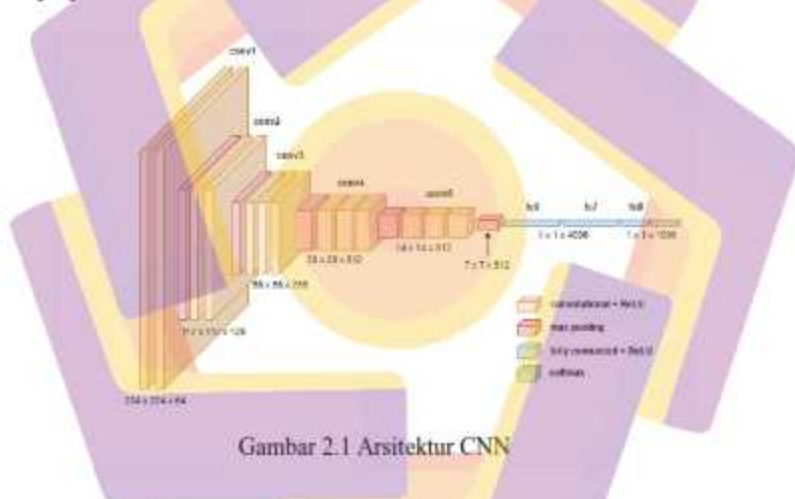
dalam jaringan mempelajari fitur pada level abstraksi yang berbeda: layer awal belajar pola dasar (misalnya tepi, sudut), sementara layer lebih dalam belajar pola kompleks [34].

Kemampuan belajar representasi tanpa fitur buatan ini membuat deep learning unggul dalam berbagai tugas, terbukti memberikan solusi akurat di berbagai bidang seperti visi komputer, pemrosesan bahasa alami, kesehatan, hingga finansial. Secara keseluruhan, deep learning telah merevolusi cara sistem komputasi belajar dari data, model yang sangat dalam dengan jutaan parameter dapat dilatih memanfaatkan dataset besar untuk mencapai kinerja yang melampaui metode konvensional dalam berbagai tugas prediktif [35].

2.3.3 Convolutional Neural Network (CNN)

CNN merupakan kelas jaringan saraf dalam yang dirancang khusus untuk pemrosesan data berstruktur grid seperti citra. Arsitektur CNN terdiri dari beberapa jenis layer utama, yaitu convolutional layer, pooling layer, dan fully-connected layer [36]. Convolutional layer berperan sebagai ekstraktor fitur: layer ini menggunakan filter (kernels) yang di-slide pada citra input untuk menangkap pola-pola lokal seperti tepian, tekstur, atau sudut [37]. Setiap filter menghasilkan feature map yang menonjolkan pola tertentu pada lokasi-lokasi di citra di mana pola tersebut terdeteksi. Melalui stacking beberapa convolutional layer, CNN mampu membangun representasi hierarkis dari citra: layer awal belajar fitur sederhana (misalnya garis tepi dan bentuk dasar), sedangkan layer yang lebih dalam belajar fitur yang semakin kompleks (misalnya motif atau bagian objek), hingga akhirnya mampu mengenali objek utuh. Di antara convolutional layer, disisipkan pooling

layer yang mengurangi dimensi spasial dari feature map (misal melalui operasi max pooling atau average pooling). Pooling ini berfungsi mereduksi jumlah parameter dan komputasi, sekaligus memberikan sifat invariansi translasi model menjadi lebih robust terhadap pergeseran atau distorsi kecil pada citra. Setelah melalui serangkaian layer konvolusi dan pooling, CNN biasanya diakhiri dengan satu atau beberapa fully-connected layer yang berperan sebagai pengklasifikasi, menggabungkan semua fitur yang telah diekstraksi menjadi prediksi akhir kelas [38].



Gambar 2.1 Arsitektur CNN

2.3.4 Transfer Learning

Transfer learning adalah teknik di mana model yang telah dilatih (pre-trained) pada suatu tugas dengan dataset besar digunakan kembali (dipinjam pengetahuannya) untuk memulai pelatihan model pada tugas lain yang terkait, biasanya dengan dataset yang lebih kecil. Intuisi di balik transfer learning adalah bahwa fitur-fitur umum yang dipelajari dari dataset sumber (misalnya ImageNet

yang berisi 1,2 juta gambar dari 1000 kelas) dapat diterapkan pada tugas target yang domainnya serupa [39]. Dalam konteks klasifikasi citra, proses transfer learning biasanya melibatkan penggunaan model convolutional neural networks (CNN) yang telah dilatih sebelumnya, seperti VGG, ResNet, atau Inception, yang kemudian disesuaikan untuk memenuhi kebutuhan dataset spesifik yang tersedia [40]. Pendekatan ini sangat efektif, terutama dalam bidang-bidang seperti kedokteran dan pertanian, di mana pengumpulan dataset citra berlabel yang luas sulit dan mahal [41].

Banyak penelitian menunjukkan keunggulan teknik transfer learning; model yang dilatih di dataset besar dapat mencapai performa tinggi meskipun data latih yang tersedia terbatas karena model tersebut sudah memiliki bobot awal yang menyimpan pengetahuan visual umum [42]. Dengan demikian, transfer learning memfasilitasi pencapaian akurasi kompetitif hanya dengan menggunakan sebagian kecil data dibandingkan jika pelatihan dilakukan dari awal (scratch). Di samping itu, inisialisasi bobot yang efektif dari model pre-trained memberikan pelatihan yang lebih cepat dan stabil serta bertindak sebagai regularisasi yang membantu mencegah overfitting ketika jumlah data latih terbatas [43], [44]. Pendekatan ini telah terbukti sangat efektif dalam bidang pertanian, termasuk dalam klasifikasi penyakit daun padi, di mana pengumpulan dataset citra berlabel dalam jumlah besar merupakan tantangan yang signifikan [45]. Banyak penelitian menunjukkan keunggulan transfer learning dalam mencapai performa yang tinggi meskipun hanya dengan data pelatihan terbatas. Hal ini memungkinkan model untuk mencapai akurasi yang kompetitif hanya dengan menggunakan sebagian kecil dari

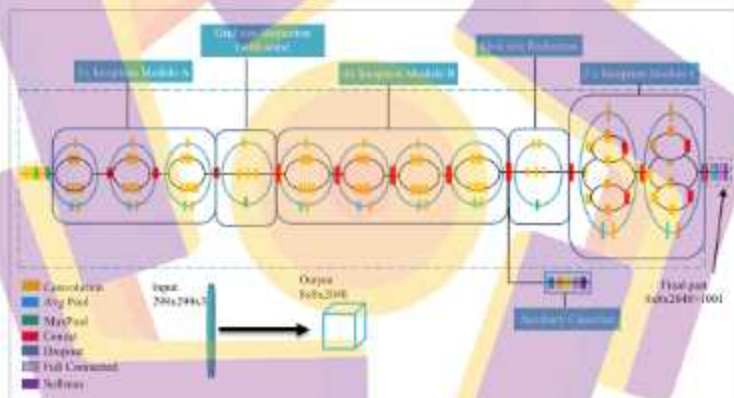
data yang dibutuhkan jika model dilatih dari awal [46]. Selain itu, inisialisasi bobot yang baik dari model pre-trained membuat proses pelatihan menjadi jauh lebih singkat dan stabil, serta bertindak sebagai bentuk regularisasi yang mencegah overfitting pada data kecil.

2.3.5 Arsitektur InceptionV3

Arsitektur Inception V3 adalah salah satu contoh arsitektur Convolutional Neural Networks (CNN) yang dirancang untuk meningkatkan kemampuan klasifikasi gambar. Dikenalkan oleh Google dan merupakan evolusi dari arsitektur sebelumnya seperti GoogLeNet, Inception V3 diperkenalkan dalam konteks kompetisi ImageNet [47]. Arsitektur ini secara signifikan mengolah dan mengekstrak fitur dengan meningkatkan kedalaman jaringan sekaligus mengurangi kompleksitas komputasi. Kurva pembelajaran yang efektif dan penggunaan berbagai ukuran filter secara paralel mendorong performa tinggi dalam pengenalan pola.

Inception V3 mengimplementasikan yang dikenal dengan Inception Modules, yang memungkinkan jaringan untuk menangkap konteks beragam dari fitur visual. Salah satu ciri khas dari Inception Modules adalah kemampuannya untuk menerapkan beberapa kernel dengan ukuran berbeda secara bersamaan, mulai dari 1x1 hingga 5x5, sehingga membantu model dalam menangkap fitur yang lebih kompleks dan mengurangi informasi yang hilang [48]. Dengan demikian, Inception V3 tidak hanya memperhitungkan informasi lokal tetapi juga konteks yang lebih luas dalam satu proses pengolahan.

Inception V3 dikenal karena efisiensinya dalam pengolahan pelatihan, berkat penggunaan convolusi terpisah yang mengurangi jumlah parameter dan kebutuhan komputasi yang berat [49]. Dengan demikian, Inception V3 memberikan banyak keuntungan dalam penerapannya di ranah komputer-vision, termasuk pengenalan objek dan pemrosesan gambar. Hal ini menciptakan peluang bagi penelitian lebih lanjut dalam bidang pertanian untuk meningkatkan ketepatan diagnosis kondisi tanaman berdasarkan citra [50].



Gambar 2.2 Arsitekur InceptionV3

Arsitektur InceptionV3 pada Gambar 2.2 tersebut menunjukkan alur pemrosesan citra mulai dari input 299x299x3 hingga menghasilkan fitur akhir berukuran 8x8x2048. Model ini menggunakan rangkaian modul Inception yang bekerja secara paralel untuk mengekstraksi fitur dari berbagai skala. Pada tahap awal, jaringan melakukan konvolusi dan pooling untuk menurunkan ukuran ruang dan membentuk representasi awal. Setelah itu, model memasuki tiga kelompok

utama, yaitu Inception Module A, B, dan C. Setiap modul berisi beberapa jalur konvolusi dengan ukuran kernel berbeda. Jalur-jalur ini digabungkan kembali sehingga model dapat menangkap detail lokal maupun pola yang lebih luas.

Di antara kelompok modul, terdapat bagian Grid Size Reduction. Bagian ini mengecilkan ukuran fitur dengan tetap mempertahankan jumlah channel sehingga proses dapat berjalan lebih efisien. InceptionV3 juga menyertakan auxiliary classifier sebagai jalur tambahan untuk membantu stabilitas pelatihan pada jaringan yang dalam. Pada bagian akhir, fitur $8 \times 8 \times 2048$ diratakan lalu diteruskan ke lapisan fully connected dan softmax untuk menghasilkan prediksi kelas. Struktur ini membantu model belajar pola visual secara lebih efektif.

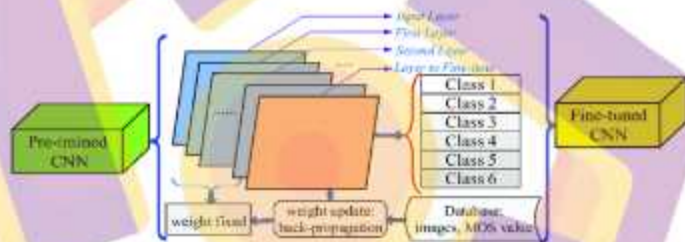
2.3.6 Fine Tuning

Fine-tuning merupakan teknik untuk menyesuaikan sebagian lapisan dari model CNN yang sudah dilatih pada dataset besar seperti ImageNet. Bobot pada model pre-trained menyimpan fitur dasar yang stabil. Fitur ini meliputi pola tepi, tekstur, dan bentuk umum yang bermanfaat untuk banyak tugas klasifikasi citra. Pada proses fine tuning, sebagian lapisan awal tetap non-trainable agar pengetahuan umum tersebut tidak berubah. Lapisan atas dibuka kembali agar dapat belajar pola visual khusus dari dataset baru yang digunakan, misalnya citra penyakit daun padi.

Pelatihan ulang dilakukan menggunakan backpropagation, tetapi hanya untuk lapisan yang dibuka. Pendekatan ini membuat pelatihan lebih efisien karena model tidak memulai dari nol. Fine tuning juga dapat meningkatkan akurasi pada dataset berukuran kecil karena model hanya menyesuaikan bagian yang relevan dengan tugas baru. Beberapa penelitian menunjukkan bahwa pembekuan lapisan tertentu,

terutama batch normalization, dapat menjaga stabilitas distribusi fitur selama proses pelatihan ulang.

Konsep ini ditunjukkan pada gambar berikut. Model pre-trained ditempatkan sebagai fondasi awal. Lapisan awal dipertahankan dengan bobot tetap. Lapisan atas dipilih sebagai bagian yang dilatih ulang. Proses ini menghasilkan model fine tuned yang lebih adaptif terhadap kelas target. Ilustrasi tersebut menggambarkan alur umum fine tuning pada CNN dan relevan diterapkan pada arsitektur InceptionV3 yang digunakan dalam penelitian ini. [51].



Gambar 2.3 Fine-Tuning of CNN Architecture

Pada Gambar 2.3, lapisan awal yang berisi fitur dasar tetap non-trainable. Lapisan atas dibuka kembali agar model dapat belajar pola visual penyakit daun padi. Model kemudian dilatih ulang dengan update bobot hanya pada bagian yang tidak non-trainable sehingga proses pelatihan lebih efisien dan akurasi dapat meningkat pada dataset berskala kecil.

2.3.7 Evaluasi Model Klasifikasi

Tahapan Tahapan evaluasi akurasi merupakan langkah penting dalam penelitian ini karena digunakan untuk menilai efektivitas model dalam melakukan klasifikasi terhadap citra daun padi. Evaluasi dilakukan dengan membandingkan

jumlah prediksi benar yang dihasilkan model dengan jumlah keseluruhan sampel pada data pengujian. Nilai akurasi menjadi indikator utama untuk menentukan sejauh mana proses pelatihan dan konfigurasi model mampu menghasilkan prediksi yang tepat [54].

Dalam penelitian ini, analisis akurasi didasarkan pada informasi yang ditampilkan melalui confusion matrix. Matriks ini menunjukkan distribusi prediksi model untuk setiap kelas dan memudahkan identifikasi jumlah sampel yang terklasifikasi benar serta sampel yang mengalami kesalahan prediksi. Komponen utama yang digunakan dalam perhitungan akurasi adalah True Positive (TP) untuk seluruh kelas serta jumlah keseluruhan sampel uji. Pada klasifikasi multi-kelas, nilai akurasi dihitung menggunakan pendekatan sederhana yang sesuai dengan struktur data, yaitu perbandingan antara prediksi benar dan total sampel [55].

Persamaan akurasi ditunjukkan pada Persamaan (1) dan untuk f1-score pada Persamaan (2) berikut:

$$\text{Akurasi} = \frac{\text{Prediksi Benar}}{\text{Total Sampel}} \quad (1)$$

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

Nilai akurasi menunjukkan besarnya proporsi prediksi yang benar terhadap keseluruhan data yang diuji. Sementara itu, F1-Score digunakan untuk mengukur keseimbangan antara nilai precision dan recall, sehingga lebih representatif dalam mengevaluasi kinerja model pada data yang tidak seimbang.

BAB 3

METODE PENELITIAN

3.1 Jenis, Sifat, dan Pendekatan Penelitian

Penelitian ini termasuk dalam jenis penelitian eksperimental kuantitatif, yang bertujuan untuk mengevaluasi performa model deep learning InceptionV3 dalam klasifikasi penyakit daun padi pada dataset terbatas melalui proses tuning hyperparameter. Penelitian ini bersifat deduktif, dimulai dari teori terkait efektivitas arsitektur CNN khususnya InceptionV3 dalam pengolahan citra digital, serta pentingnya tuning parameter untuk meningkatkan akurasi pada dataset kecil.

Dalam penelitian ini, model InceptionV3 digunakan sebagai kerangka utama klasifikasi citra, dan dilakukan eksperimen dengan beberapa kombinasi hyperparameter seperti learning rate, batch size, jumlah epoch, dan optimizer. Proses tuning dilakukan untuk mengetahui konfigurasi terbaik yang menghasilkan performa optimal.

Sebagai penelitian kuantitatif, pendekatan ini menekankan pada analisis data numerik dari hasil klasifikasi, yang diukur dengan metrik evaluasi seperti akurasi, precision, recall, dan F1-score. Eksperimen dilakukan pada tiga konfigurasi utama: model default (tanpa tuning), tuning parsial, dan tuning menyeluruh.

3.2 Metode Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari UCI Machine Learning Repository, yaitu dataset citra daun padi yang terinfeksi oleh beberapa jenis penyakit, di antaranya Bacterial Leaf Blight, Leaf

Smut, dan Brown Spot. Dataset ini telah banyak digunakan dalam penelitian-penelitian terdahulu di bidang klasifikasi penyakit pada daun padi, sehingga layak dijadikan basis pengujian model klasifikasi berbasis machine learning.

Dataset tersebut terdiri dari gambar-gambar daun padi dalam format RGB yang masing-masing telah memiliki label kelas (jenis penyakit). Data diunduh dalam format kompresi dan selanjutnya diekstraksi ke direktori lokal untuk proses analisis lanjutan. Setiap citra diperiksa kualitasnya, termasuk ukuran, resolusi, dan kejernihan. Gambar-gambar dengan kualitas sangat rendah atau kabur dihapus dari dataset. Dengan pendekatan ini, proses pengumpulan data dalam penelitian bersifat digital dan eksperimen, serta menggunakan dataset yang telah divalidasi oleh komunitas ilmiah, sehingga layak dijadikan sumber data yang dapat diandalkan untuk keperluan klasifikasi berbasis deep learning.

3.3 Metode Analisis Data

Metode analisis data dalam penelitian ini dilakukan untuk mengevaluasi performa model pada setiap skenario pelatihan yang telah dirancang. Analisis berfokus pada bagaimana model mempelajari pola dari data pelatihan, mempertahankan konsistensi saat divalidasi, dan memberikan hasil yang akurat pada data pengujian. Evaluasi dilakukan dengan menggunakan nilai akurasi yang dihasilkan pada tiga tahap, yaitu akurasi pelatihan, akurasi validasi, dan akurasi pengujian. Nilai akurasi pelatihan digunakan untuk melihat sejauh mana model mampu mempelajari fitur dari data yang diberikan. Akurasi validasi dianalisis untuk menilai stabilitas proses pelatihan dan mendeteksi kemungkinan overfitting ketika akurasi pelatihan terlalu tinggi dibandingkan akurasi validasi. Akurasi

pengujian menjadi indikator utama untuk melihat performa sebenarnya ketika model diuji pada data yang belum pernah dilihat sebelumnya. Selain itu, F1-Score digunakan untuk melengkapi evaluasi akurasi dengan mempertimbangkan keseimbangan antara precision dan recall.

Selain menggunakan akurasi, analisis juga dilakukan dengan mengamati grafik training dan validation loss pada setiap skenario. Grafik ini membantu memahami dinamika pelatihan model, seperti apakah proses belajar berjalan stabil atau mengalami fluktuasi. Loss yang menurun secara konsisten menunjukkan proses pembelajaran yang baik, sedangkan perbedaan yang terlalu besar antara training loss dan validation loss dapat menunjukkan indikasi ketidakseimbangan dalam proses pelatihan.

Untuk mendukung analisis lebih lanjut, confusion matrix digunakan untuk melihat distribusi prediksi pada setiap kelas penyakit daun padi. Dengan matriks ini, dapat diamati kelas mana yang paling sering diprediksi dengan benar dan kelas mana yang masih menghasilkan kesalahan prediksi.

Hasil evaluasi dari setiap skenario kemudian dibandingkan untuk melihat perbedaan performa secara menyeluruh. Perbandingan dilakukan dengan memperhatikan akurasi pengujian, nilai F1-Score, pola loss selama pelatihan, serta bentuk distribusi prediksi pada confusion matrix. Seluruh proses analisis ini digunakan sebagai dasar untuk menjawab rumusan masalah dan menarik kesimpulan penelitian.

Alur penelitian ini menjelaskan tahapan-tahapan yang dilakukan dalam proses klasifikasi penyakit daun padi menggunakan model InceptionV3 berbasis transfer learning. Setiap tahap memiliki peran penting dalam menghasilkan model dengan performa terbaik. Secara umum, alur penelitian ditunjukkan pada Gambar 3.1 alur penelitian.

A. Permasalahan dan Studi Literatur

Tahap awal penelitian dimulai dengan mengidentifikasi masalah pada deteksi penyakit daun padi yang sering menyebabkan penurunan hasil panen. Identifikasi secara manual masih lambat dan tidak selalu akurat. Gejala antar penyakit pun sering mirip, sehingga diperlukan sistem klasifikasi otomatis yang lebih cepat dan konsisten untuk membantu deteksi dini di lapangan.

Studi literatur dilakukan dengan meninjau penelitian yang menggunakan machine learning dan deep learning untuk klasifikasi penyakit tanaman. Pendekatan tradisional seperti SVM, KNN, Random Forest, dan Decision Tree mampu memberikan akurasi tinggi, tetapi performanya menurun ketika citra memiliki variasi cahaya, latar belakang kompleks, atau tekstur daun yang beragam. CNN kemudian digunakan karena mampu mengekstraksi fitur otomatis dan bekerja lebih stabil pada citra bertekstur kompleks.

Beberapa penelitian sebelumnya juga menganalisis dan membandingkan berbagai arsitektur CNN dalam klasifikasi penyakit tanaman. InceptionV3 muncul sebagai salah satu model dengan performa yang baik, namun akurasi masih berubah ketika dataset kecil atau kondisi visual berbeda dari data pelatihan. Literatur terbaru menunjukkan bahwa fine-tuning dapat

meningkatkan akurasi dan membuat model lebih adaptif terhadap variasi citra. Berdasarkan temuan tersebut, penelitian ini memilih InceptionV3 dan menerapkan fine-tuning untuk meningkatkan stabilitas dan akurasi model pada dataset daun padi berskala kecil.

B. Persiapan Dataset

Pada tahap persiapan dataset, data citra daun padi disusun berdasarkan kelas penyakit dan dibagi dalam dua tahap. Sebanyak 75% data digunakan untuk pelatihan dan validasi, sedangkan 25% data digunakan untuk pengujian. Data pelatihan dan validasi selanjutnya dibagi kembali dengan proporsi 75% sebagai data pelatihan dan 25% sebagai data validasi, sehingga diperoleh proporsi akhir sebesar 56,25% data pelatihan, 18,75% data validasi, dan 25% data pengujian. Sebelum proses pelatihan dilakukan, seluruh citra melalui tahap prapemrosesan. Pada tahap ini, citra diubah ukurannya menjadi 299×299 piksel agar sesuai dengan ukuran input standar arsitektur InceptionV3, kemudian dikonversi ke dalam bentuk tensor dan dinormalisasi menggunakan nilai mean dan standar deviasi ImageNet. Tahapan ini bertujuan untuk memastikan data masukan sesuai dengan karakteristik model pretrained yang digunakan.

C. Implementasi Model Skenario

Pada tahap implementasi model skenario, model InceptionV3 digunakan sebagai backbone utama dan diterapkan ke dalam beberapa konfigurasi pelatihan. Fine tuning dilakukan pada dua bagian, yaitu layer awal dan layer akhir. Pada skenario fine tuning layer awal, sebanyak 150 dan

250 layer pertama dijadikan trainable, sementara layer lainnya tetap non trainable. Pada skenario fine tuning layer akhir, sebanyak 150 dan 250 layer terakhir dijadikan trainable, sedangkan layer awal dibekukan. Pendekatan ini bertujuan membandingkan pengaruh penyesuaian fitur dasar dan fitur tingkat tinggi terhadap akurasi klasifikasi.

Setiap skenario dilatih menggunakan tiga optimizer, yaitu Adam, AdamW, dan SGD, dengan learning rate sebesar 0,0001. Proses pelatihan menggunakan batch size 8 dan jumlah maksimum epoch sebanyak 100. Untuk mencegah overfitting dan menghentikan pelatihan saat performa tidak lagi meningkat, digunakan mekanisme early stopping berdasarkan kinerja validasi. Seluruh skenario kemudian dievaluasi menggunakan data testing untuk memperoleh nilai akurasi sebagai dasar perbandingan performa antar konfigurasi.

D. Evaluasi Performa Model Setiap Skenario

Pada tahap evaluasi performa model setiap skenario, masing-masing model hasil pelatihan diuji menggunakan data testing yang sama. Evaluasi ini bertujuan untuk menilai kemampuan generalisasi model dalam mengklasifikasikan tiga kelas penyakit daun padi pada data yang belum pernah dilihat sebelumnya. Pengujian dilakukan secara terpisah untuk setiap skenario guna memperoleh gambaran performa masing-masing konfigurasi model.

Kinerja model diukur menggunakan nilai akurasi dan F1-score yang diperoleh dari hasil prediksi pada data uji. Kedua metrik tersebut digunakan untuk memberikan gambaran yang lebih komprehensif terhadap kemampuan model, khususnya dalam menangani distribusi kelas yang berbeda. Hasil evaluasi pada tahap ini selanjutnya menjadi dasar untuk dilakukan perbandingan performa antar skenario pada subbab berikutnya.

E. Komparasi Performa Model dan Kesimpulan

Pada tahap ini, dilakukan komparasi performa antar skenario berdasarkan hasil pengujian menggunakan data testing. Perbandingan dilakukan dengan memperhatikan nilai akurasi dan F1-score untuk menentukan skenario dengan performa terbaik. Hasil komparasi tersebut digunakan untuk mengidentifikasi pengaruh perbedaan strategi fine-tuning dan konfigurasi pelatihan terhadap kinerja model InceptionV3.

Selanjutnya, performa terbaik yang diperoleh dibandingkan dengan hasil penelitian sebelumnya yang menggunakan dataset yang sama untuk melihat sejauh mana peningkatan performa yang dicapai. Berdasarkan hasil perbandingan tersebut, ditarik kesimpulan mengenai kemampuan model dalam mengenali penyakit daun padi pada dataset berskala kecil, sekaligus menunjukkan kontribusi penelitian ini dalam meningkatkan akurasi dan kestabilan performa model dibandingkan penelitian terdahulu.

BAB 4

HASIL PENELITIAN DAN PEMBAHASAN




4.1 Persiapan dataset

Tahap persiapan dataset dilakukan untuk memastikan data siap digunakan dalam proses pelatihan dan evaluasi model. Pada tahap ini dilakukan pengumpulan dataset, pembagian data menjadi subset pelatihan, validasi, dan pengujian, serta penerapan augmentasi untuk menambah variasi citra. Seluruh langkah ini bertujuan agar model dapat belajar secara optimal dari data yang tersedia dan memiliki kemampuan generalisasi yang baik pada citra yang belum pernah dilihat sebelumnya.

4.1.1 Pengumpulan dataset

Pada penelitian ini, data yang digunakan berasal dari Rice Leaf Diseases Dataset yang diperoleh melalui situs UCI Machine Learning Repository [23]. Dataset ini berisi tiga kelas penyakit daun padi, yaitu Bacterial Leaf Blight, Brown Spot, dan Leaf Smut, dengan masing-masing kelas terdiri atas 40 citra sehingga total dataset berjumlah 120 citra. Sebelum proses pembagian data dilakukan, seluruh citra di-load terlebih dahulu dari Google Drive ke lingkungan kerja Google Colab untuk memastikan bahwa setiap file terbaca dengan benar dan untuk memverifikasi jumlah citra pada masing-masing kelas. Informasi ini diperlukan agar pembagian dataset tetap seimbang dan sesuai kebutuhan penelitian. Rincian jumlah data dan distribusinya dapat dilihat pada Tabel 4.1.

Tabel 4.1 Detail dataset penyakit daun padi

Jenis Penyakit	Contoh citra daun terkena penyakit	Jumlah data
Bacterial Leaf Blight		40
Brown Spot		40
Leaf Smut		40
Total		120

Setelah memastikan jumlah citra pada setiap kelas sesuai dengan distribusi pada Tabel 4.1, langkah berikutnya adalah melakukan proses load data ke lingkungan kerja Google Colab. Proses ini dilakukan agar seluruh citra dapat diakses oleh sistem sebelum memasuki tahap pembagian data dan pelatihan model. Pada tahap ini, direktori dataset dibaca secara otomatis untuk memuat seluruh file citra dari masing-masing kelas penyakit. Proses load data ini juga berfungsi sebagai pengecekan awal untuk memastikan bahwa tidak terdapat file yang rusak atau tidak terbaca oleh sistem. Potongan kode berikut menunjukkan proses yang digunakan untuk mengimpor dataset dari Google Drive ke lingkungan Colab.

```

import os
import numpy as np

original_dataset_dir =
'/content/drive/MyDrive/ARTECH/Eksperimen/rice+leaf+diseases'

classes = sorted(os.listdir(original_dataset_dir))
print("Kelas yang terdeteksi:", classes)
for c in classes:
    class_dir = os.path.join(original_dataset_dir, c)
    num_images = len([f for f in os.listdir(class_dir) if
f.lower().endswith(('.jpg', '.jpeg', '.png'))])
    print(f"{c}: {num_images} gambar")

```

Pada script kode di atas digunakan untuk membaca dataset asli dari direktori penyimpanan di Google Drive. Variabel `original_dataset_dir` berfungsi untuk menyimpan path utama dataset yang berisi tiga folder kelas penyakit daun padi, yaitu Bacterial Leaf Blight, Brown Spot, dan Leaf Smut.

Fungsi `os.listdir()` digunakan untuk mendeteksi nama-nama kelas yang terdapat di dalam direktori tersebut, sedangkan perulangan `for` digunakan untuk menghitung jumlah citra pada setiap kelas dengan membaca file berformat `.jpg`, `.jpeg`, dan `.png`. Hasil eksekusi kode ini menampilkan daftar kelas yang terdeteksi beserta jumlah citra yang tersedia pada masing-masing kelas. Berdasarkan hasil tersebut, dataset terdiri dari tiga kelas dengan total 120 citra, masing-masing kelas berisi 40 citra daun padi.

4.1.2 Prapemrosesan Data

Sebelum digunakan dalam proses pelatihan, seluruh citra melalui tahap prapemrosesan untuk menyesuaikan karakteristik input dengan arsitektur InceptionV3. Tahap prapemrosesan yang dilakukan meliputi perubahan ukuran citra menjadi 299×299 piksel, konversi citra ke dalam bentuk tensor, serta normalisasi menggunakan nilai mean dan standar deviasi ImageNet. Tahapan ini bertujuan agar distribusi data masukan sesuai dengan model pretrained yang digunakan.

Tahap prapemrosesan ini diimplementasikan pada bagian kode yang mendefinisikan transformasi data sebelum data dimuat ke dalam model.

```

from torchvision import transforms

data_transforms = transforms.Compose([
    transforms.Resize((299, 299)),
    transforms.ToTensor(),
    transforms.Normalize(
        mean=[0.485, 0.456, 0.406],
        std=[0.229, 0.224, 0.225]
    )
])

```

Pada potongan kode di atas, fungsi `transforms.Resize((299, 299))` digunakan untuk mengubah ukuran seluruh citra masukan menjadi 299×299 piksel agar sesuai dengan ukuran input standar arsitektur InceptionV3. Selanjutnya, `transforms.ToTensor()` berfungsi untuk mengonversi citra ke dalam bentuk tensor sehingga dapat diproses oleh model berbasis PyTorch. Tahap terakhir adalah normalisasi menggunakan `transforms.Normalize()`, dengan nilai mean dan standard deviation yang berasal dari dataset ImageNet. Normalisasi ini dilakukan untuk

menyelaraskan distribusi data masukan dengan karakteristik model InceptionV3 yang telah dilatih sebelumnya.

Transformasi prapemrosesan tersebut diterapkan secara konsisten pada data pelatihan, validasi, dan pengujian melalui mekanisme pemuatan data (data loader), sehingga seluruh data yang digunakan dalam penelitian berada pada format dan skala yang sama.

4.1.3 Split Dataset

Tahapan ini bertujuan untuk membagi dataset ke dalam tiga subset utama, yaitu data pelatihan (training), data validasi (validation), dan data pengujian (testing). Proses pembagian data dilakukan menggunakan Google Colab dengan bantuan pustaka scikit-learn, khususnya fungsi `train_test_split`, agar pembagian data dapat dilakukan secara acak namun proporsional untuk setiap kelas.

Dataset yang telah dibaca sebelumnya dibagi menjadi dua tahap. Tahap pertama memisahkan 25% dari total data di setiap kelas sebagai data pengujian (testing). Tahap kedua membagi sisa 75% data menjadi 75% untuk pelatihan (training) dan 25% untuk validasi (validation). Dengan pembagian tersebut, diperoleh komposisi akhir yaitu 22 citra untuk pelatihan, 8 citra untuk validasi, dan 10 citra untuk pengujian pada setiap kelas penyakit daun padi.

Berikut merupakan potongan kode yang digunakan untuk melakukan proses pengelompokan data:

```

import os
import shutil
import numpy as np
from sklearn.model_selection import train_test_split

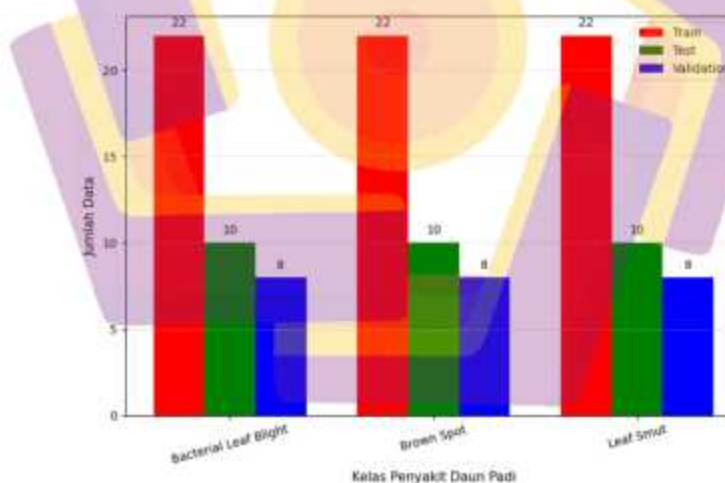
original_dataset_dir =
    '/content/drive/MyDrive/AMIKOM/Eksperimen/rice+leaf+diseases'
base_dir = '/content/drive/MyDrive/AMIKOM/Eksperimen/Spilitdata'

os.makedirs(base_dir, exist_ok=True)
train_dir = os.path.join(base_dir, 'train')
val_dir = os.path.join(base_dir, 'val')
test_dir = os.path.join(base_dir, 'test')
for dir_path in [train_dir, val_dir, test_dir]:
    os.makedirs(dir_path, exist_ok=True)
classes = sorted(os.listdir(original_dataset_dir))
print("Kelas yang terdeteksi:", classes)
for c in classes:
    class_dir = os.path.join(original_dataset_dir, c)
    images = np.array([f for f in os.listdir(class_dir) if
        f.lower().endswith(('.jpg', '.jpeg', '.png'))])
    trainval_files, test_files = train_test_split(images, test_size=0.25,
        random_state=41)
    train_files, val_files = train_test_split(trainval_files, test_size=0.25,
        random_state=42)
    for subset, file_list in zip(['train', 'val', 'test'], [train_files,
        val_files, test_files]):
        subset_dir = os.path.join(base_dir, subset, c)
        os.makedirs(subset_dir, exist_ok=True)
        for f in file_list:
            shutil.copy(os.path.join(class_dir, f), os.path.join(subset_dir, f))
    print(f"({c}): Train={len(train_files)}, Val={len(val_files)},
        Test={len(test_files)}")

```

Kode di atas digunakan untuk membagi dataset penyakit daun padi menjadi tiga subset utama, yaitu data pelatihan, validasi, dan pengujian. Pada bagian awal, pustaka `os`, `shutil`, `numpy`, dan `train_test_split` digunakan untuk mengatur direktori, menyalin file citra, serta membagi data secara proporsional. Variabel `original_dataset_dir` berfungsi menyimpan lokasi dataset utama, sedangkan `base_dir` digunakan sebagai direktori hasil pembagian data.

Selanjutnya, sistem mendeteksi tiga kelas penyakit daun padi (Bacterial Leaf Blight, Brown Spot, dan Leaf Smut) menggunakan fungsi `os.listdir()`. Proses pembagian data dilakukan dengan dua tahap menggunakan `train_test_split`: tahap pertama memisahkan 25% data untuk pengujian, dan tahap kedua membagi sisa 75% data menjadi 75% untuk pelatihan dan 25% untuk validasi. Hasil pembagian menghasilkan 22 citra untuk pelatihan, 8 citra untuk validasi, dan 10 citra untuk pengujian pada setiap kelas. Setelah itu, sistem membuat folder baru untuk setiap subset dan menyalin citra ke dalam direktori yang sesuai. Hasil akhir menunjukkan bahwa seluruh data berhasil dibagi secara seimbang ke dalam tiga subset, dengan total 120 citra yang terdistribusi merata pada ketiga kelas penyakit daun padi.



Gambar 4.1 Pengelompokan data penyakit daun padi

Berdasarkan Gambar 4.1 Pengelompokan data penyakit daun padi, dapat dilihat bahwa pembagian data pada setiap kelas penyakit daun padi memiliki proporsi yang

seimbang. Setiap kelas terdiri atas 22 citra data pelatihan, 8 citra data validasi, dan 10 citra data pengujian. Keseimbangan jumlah data pada setiap subset ini penting untuk memastikan proses pelatihan model berjalan adil tanpa bias terhadap salah satu kelas.

4.2 Implementasi Model Skenario

Pada Pada tahap ini, model InceptionV3 diterapkan ke dalam beberapa skenario pelatihan yang dirancang untuk mengevaluasi pengaruh strategi fine-tuning terhadap kinerja klasifikasi penyakit daun padi. Seluruh skenario menggunakan arsitektur dasar yang sama, yaitu InceptionV3 pretrained. Perbedaan antar skenario hanya terletak pada pengaturan layer yang dibuat trainable.

Konfigurasi parameter pelatihan ditetapkan secara seragam pada seluruh skenario untuk menjaga konsistensi evaluasi. Parameter yang digunakan meliputi learning rate sebesar 0,0001, jumlah epoch sebanyak 100, dan batch size sebesar 8. Selain itu, penelitian ini menggunakan tiga jenis optimizer, yaitu Adam, AdamW, dan SGD. Rincian konfigurasi model yang digunakan pada setiap skenario disajikan pada Tabel 4.2.

Implementasi model dilakukan melalui empat skenario pelatihan yang dijalankan secara terpisah sesuai dengan konfigurasi layer trainable masing-masing. Setelah proses training selesai, model dari setiap skenario dievaluasi menggunakan data testing untuk mengukur kinerja klasifikasi.

Tabel 4.2 Konfigurasi Skenario Pelatihan Model InceptionV3

Skenario	Layer trainable	Learning Rate	Epoch	Batch Size
SK 1	0	0,0001	100	8
SK 2	150, 250 Layer Awal	0,0001	100	8
SK 3	150, 250 Layer akhir	0,0001	100	8
SK 4	Full trainable	0,0001	100	8

4.2.1 Skenario 1 – Semua Layer Non-trainable

Skenario pertama menerapkan model InceptionV3 dengan seluruh layer dalam kondisi non-trainable. Pada skenario ini, bobot pretrained yang dimiliki oleh model tidak mengalami pembaruan selama proses training. Pelatihan hanya dilakukan pada bagian classifier layer yang ditambahkan pada tahap akhir model.

Tujuan dari skenario ini adalah untuk memperoleh kinerja dasar sebagai pembandingan terhadap skenario fine-tuning. Pendekatan ini merepresentasikan penerapan transfer learning tanpa penyesuaian bobot pada layer convolutional, sehingga kemampuan model sepenuhnya bergantung pada fitur umum yang telah dipelajari dari dataset besar sebelumnya.

```

model = models.inception_v3(pretrained=True)

model.fc = nn.Linear(model.fc.in_features, num_classes)
model.AuxLogits.fc = nn.Linear(model.AuxLogits.fc.in_features, num_classes)

for p in model.parameters():
    p.requires_grad = False

for p in model.fc.parameters():
    p.requires_grad = True
for p in model.AuxLogits.fc.parameters():
    p.requires_grad = True

```

Pada skenario ini, model InceptionV3 diinisialisasi menggunakan bobot pretrained dan digunakan sebagai backbone ekstraksi fitur. Seluruh parameter pada backbone model dibuat dalam kondisi non-trainable sehingga bobot hasil pretraining tidak mengalami pembaruan selama proses pelatihan.

Classifier layer pada bagian akhir model, yaitu fully connected layer utama dan auxiliary classifier, disesuaikan dengan jumlah kelas penyakit daun padi dan dibuat trainable. Dengan demikian, proses pembelajaran hanya terjadi pada layer klasifikasi tanpa memengaruhi bobot convolutional layer pada backbone InceptionV3. Pendekatan ini bertujuan untuk mengevaluasi kemampuan transfer learning murni sebagai baseline, di mana model memanfaatkan fitur umum hasil pretraining tanpa melakukan fine-tuning pada layer ekstraksi fitur. Hasil dari skenario ini digunakan sebagai pembanding terhadap skenario fine-tuning parsial dan full fine-tuning pada tahap selanjutnya. penelitian.

4.2.2 Skenario 2 – Fine Tuning Layer Awal

Pada skenario kedua, fine tuning diterapkan pada layer awal model InceptionV3. Sejumlah layer pada bagian awal model diatur sebagai trainable, sedangkan layer lainnya tetap non trainable. Jumlah layer awal yang dilatih divariasikan menjadi lima konfigurasi, yaitu 150 layer, dan 250 layer. Variasi ini digunakan untuk mengamati pengaruh peningkatan kedalaman fine tuning terhadap proses pembelajaran model.

Pelatihan pada layer awal bertujuan untuk menyesuaikan fitur dasar seperti tepi dan tekstur agar lebih sesuai dengan karakteristik citra daun padi. Dengan menambah jumlah layer yang dilatih secara bertahap, penelitian ini mengevaluasi sejauh mana pelatihan ulang fitur umum dapat meningkatkan kinerja klasifikasi pada dataset yang berjumlah terbatas. Hasil dari setiap konfigurasi layer awal trainable dibandingkan untuk melihat perubahan performa model seiring dengan bertambahnya jumlah layer yang dilatih.

Mekanisme pengaturan layer awal trainable diimplementasikan pada fungsi `create_model`, sebagaimana ditunjukkan pada potongan kode berikut.

```
elif mode == "first_k":  
  
    k = scenario.config['k_layers']  
    for i, p in enumerate(all_params):  
        if i < k:  
            p.requires_grad = True  
    for p in model.fc.parameters():  
        p.requires_grad = True  
    for p in model.AuxLogits.fc.parameters():  
        p.requires_grad = True
```

Ketika mode pelatihan bernilai `first_k`, model melakukan fine-tuning parsial dengan mengaktifkan sejumlah parameter awal sebagai trainable. Seluruh parameter model telah dibekukan pada tahap sebelumnya, sehingga hanya parameter yang memenuhi kondisi tertentu yang diaktifkan kembali.

Parameter `k_layers` menentukan jumlah parameter awal yang dibuat trainable. Proses iterasi dilakukan terhadap seluruh parameter model yang telah disusun secara berurutan. Kondisi $i < k$ memastikan bahwa hanya parameter pada bagian awal model sesuai nilai `k_layers` yang diaktifkan, sedangkan parameter lainnya tetap non-trainable.

Selain itu, classifier layer utama dan auxiliary classifier selalu dibuat trainable pada mode ini. Hal ini bertujuan agar hasil ekstraksi fitur dari layer awal yang dilatih dapat dipetakan secara optimal ke kelas keluaran sesuai dengan jumlah kelas pada dataset penelitian.

```
# As first k layers
{'name': 'A_150', 'folder': 'A_150', 'fine_tuning_mode': 'first_k', 'k_layers':
150, 'lr': 1e-4, 'optimizer': 'sgd', 'epochs': 100, 'batch_size': 81,
{'name': 'A_200', 'folder': 'A_200', 'fine_tuning_mode': 'first_k', 'k_layers':
200, 'lr': 1e-4, 'optimizer': 'sgd', 'epochs': 100, 'batch_size': 81,
```

Bagian kode tersebut mendefinisikan lima konfigurasi skenario fine-tuning pada layer awal model InceptionV3. Seluruh skenario menggunakan mode pelatihan `first_k`, yang menunjukkan bahwa fine-tuning dilakukan dengan mengaktifkan sejumlah parameter awal pada model, sementara parameter lainnya tetap berada dalam kondisi non-trainable.

Parameter `k_layers` berfungsi sebagai penentu jumlah layer awal yang dibuat trainable pada setiap skenario. Pada skenario `A_150`, sebanyak 150 parameter awal diaktifkan untuk dilatih. Jumlah ini ditingkatkan secara bertahap pada skenario hingga `A_250`. Setiap konfigurasi dijalankan sebagai eksperimen terpisah dengan tujuan untuk mengamati pengaruh peningkatan jumlah layer awal yang dilatih terhadap kinerja model.

Seluruh konfigurasi skenario menggunakan parameter pelatihan yang sama, yaitu learning rate sebesar 0,0001, untuk optimizer akan dilakukan percobaan yaitu Adam, AdamW dan SGD, jumlah epoch sebanyak 100, dan batch size sebesar 8. Dengan menyamakan parameter pelatihan, perbedaan performa yang dihasilkan antar skenario dapat dikaitkan secara langsung dengan variasi jumlah layer awal yang dibuat trainable.

4.2.3 Skenario 3 – Fine-Tuning Layer Akhir

Pada skenario ketiga, fine tuning diterapkan pada layer akhir model InceptionV3. Pada skenario ini, layer awal hingga bagian tengah arsitektur dipertahankan sebagai non trainable, sedangkan sejumlah layer pada bagian akhir diatur sebagai trainable. Pendekatan ini bertujuan untuk menyesuaikan fitur tingkat tinggi yang lebih spesifik terhadap karakteristik penyakit daun padi tanpa mengubah fitur dasar yang telah dipelajari sebelumnya.

Jumlah layer akhir yang dilatih divariasikan menjadi 150 layer, dan 250 layer. Variasi ini digunakan untuk mengevaluasi pengaruh peningkatan kedalaman fine tuning pada bagian akhir model terhadap kinerja klasifikasi. Dengan melatih

layer yang lebih dekat dengan lapisan klasifikasi, model diharapkan mampu mempelajari pola visual penyakit daun padi secara lebih efektif, terutama pada kondisi dataset yang berjumlah terbatas.

Pengaturan layer akhir yang dilatih diimplementasikan pada fungsi `create_model` dengan menggunakan mode pelatihan `last_k`. Mekanisme ini diaktifkan hanya ketika parameter `fine_tuning_mode` bernilai `last_k`, sebagaimana ditunjukkan pada potongan kode berikut.

```

if mode == "last_k":
    k = scenario_config['k_layers']
    for i, p in enumerate(all_params):
        if i >= total_layers - k:
            p.requires_grad = True
    for p in model.fc.parameters():
        p.requires_grad = True
    for p in model.AuxLogits.fc.parameters():
        p.requires_grad = True

```

Pada mode `last_k`, model melakukan fine-tuning parsial dengan mengaktifkan sejumlah parameter pada bagian akhir arsitektur InceptionV3. Seluruh parameter model telah dibuat non-trainable pada tahap sebelumnya, sehingga hanya parameter yang memenuhi kondisi pada bagian ini yang akan diperbarui selama proses pelatihan.

Parameter `k_layers` menentukan jumlah layer akhir yang dibuat trainable. Kondisi $i \geq \text{total_layers} - k$ memastikan bahwa hanya parameter pada bagian akhir model yang diaktifkan sebagai trainable, sedangkan parameter pada bagian awal dan tengah tetap menggunakan bobot hasil pretraining.

Pendekatan ini memungkinkan penyesuaian fitur tingkat tinggi yang berperan dalam proses pengambilan keputusan klasifikasi, tanpa mengubah fitur dasar yang telah dipelajari pada tahap awal jaringan.

Variasi jumlah layer akhir yang dilatih ditentukan pada tahap pendefinisian skenario eksperimen sebagai berikut.

```
# B: last K layers
({'name': 'B_150', 'folder': 'B_150', 'fine_tuning_mode': 'last_k', 'k_layers':
150, 'lr': 1e-4, 'optimizer': 'sgd', 'epochs': 100, 'batch_size': 8},
{'name': 'B_250', 'folder': 'B_250', 'fine_tuning_mode': 'last_k', 'k_layers':
250, 'lr': 1e-4, 'optimizer': 'sgd', 'epochs': 100, 'batch_size': 8},
```

Setiap konfigurasi skenario merepresentasikan jumlah layer akhir yang dilatih secara berbeda. Seluruh konfigurasi menggunakan parameter pelatihan yang sama, yaitu learning rate sebesar 0,0001, jumlah epoch sebanyak 100, dan batch size sebesar 8 parameter ini digunakan tetap tetapi untuk optimizer yang digunakan nantinya akan diuji yaitu optimizer Adam, AdamW, SGD. Dengan menyamakan parameter pelatihan, perbedaan performa yang dihasilkan antar skenario dapat dikaitkan secara langsung dengan variasi jumlah layer akhir yang dibuat trainable.

Melalui skenario ini, penelitian membandingkan efektivitas fine-tuning pada layer akhir dengan fine-tuning pada layer awal, serta mengevaluasi strategi fine-tuning parsial yang paling sesuai untuk meningkatkan kinerja klasifikasi pada dataset penyakit daun padi yang berjumlah terbatas.

4.2.4 Skenario 4 – Full Fine Tuning (Semua Layer Trainable)

Pada skenario keempat, diterapkan strategi full fine-tuning pada model InceptionV3. Pada pendekatan ini, seluruh layer pada model diatur sebagai trainable sehingga seluruh bobot model mengalami pembaruan selama proses pelatihan. Berbeda dengan skenario fine-tuning parsial, strategi ini memungkinkan model menyesuaikan seluruh representasi fitur, baik fitur tingkat rendah maupun fitur tingkat tinggi, secara langsung terhadap dataset penyakit daun padi.

Penerapan full fine-tuning bertujuan untuk mengevaluasi kemampuan maksimum model dalam mempelajari karakteristik data penelitian. Dengan membuka seluruh layer untuk dilatih, model tidak hanya memanfaatkan fitur umum hasil pretraining, tetapi juga mengadaptasi seluruh parameter agar lebih sesuai dengan pola visual spesifik pada citra daun padi. Pendekatan ini digunakan sebagai pembandingan terhadap skenario non-trainable dan fine-tuning parsial.

Pengaturan full fine-tuning diimplementasikan pada fungsi `create_model` dengan menggunakan mode pelatihan full. Mode ini diaktifkan ketika parameter `fine_tuning_mode` bernilai full, sebagaimana ditunjukkan pada potongan kode berikut.

```
elif mode == "full":  
    for p in model.parameters():  
        p.requires_grad = True
```

Ketika mode pelatihan bernilai full, seluruh parameter pada model InceptionV3 dibuat dalam kondisi trainable. Tidak ada parameter yang dibekukan,

sehingga seluruh bobot model, mulai dari layer awal hingga layer akhir, dapat diperbarui selama proses pelatihan.

Pendekatan ini memungkinkan model untuk menyesuaikan fitur dasar, fitur menengah, dan fitur tingkat tinggi secara bersamaan. Dengan demikian, model memiliki fleksibilitas penuh untuk mempelajari representasi fitur yang paling sesuai dengan karakteristik dataset penyakit daun padi.

Konfigurasi skenario full fine-tuning ditentukan pada tahap pendefinisian skenario eksperimen sebagai berikut.

```
{  
  'name': 'FULL',  
  'folder': 'FULL',  
  'fine_tuning_mode': 'full',  
  'lr': 1e-4,  
  'optimizer': 'adam',  
  'epochs': 100,  
  'batch_size': 8  
}
```

Pada skenario ini, seluruh layer model dilatih menggunakan parameter pelatihan yang sama dengan skenario lainnya, yaitu learning rate sebesar 0,0001, jumlah epoch sebanyak 100, dan batch size sebesar 8, parameter. Penyesuaian parameter pelatihan ini dilakukan untuk memastikan bahwa perbedaan performa yang dihasilkan dapat dikaitkan secara langsung dengan strategi fine-tuning yang digunakan, bukan akibat perbedaan konfigurasi pelatihan. Untuk optimizer yang digunakan atau di uji kan yaitu Adam, AdamW, dan SGD.

4.3 Evaluasi Performa Model Setiap Skenario

Bagian ini membahas hasil evaluasi dan komparasi model yang diperoleh dari seluruh skenario pelatihan yang telah dirancang. Evaluasi dilakukan untuk menilai kinerja model InceptionV3 dalam mengklasifikasikan penyakit daun padi dengan mempertimbangkan keterbatasan jumlah data yang digunakan. Fokus utama evaluasi diarahkan pada nilai akurasi sebagai indikator kinerja klasifikasi. Pada setiap skenario, model dilatih dan dievaluasi menggunakan tiga jenis optimizer, yaitu Adam, AdamW, dan SGD. Pengujian beberapa optimizer dilakukan untuk menganalisis pengaruh metode optimasi terhadap kinerja model pada kondisi dataset terbatas. Seluruh hasil pengujian optimizer ditampilkan dan dianalisis secara menyeluruh sebagai bagian dari evaluasi.

Selain itu, proses pembelajaran model dianalisis melalui grafik training accuracy, validation accuracy, training loss, dan validation loss. Grafik-grafik tersebut digunakan untuk menilai stabilitas pelatihan dan perilaku konvergensi model selama proses training. Sebagai analisis lanjutan, confusion matrix digunakan untuk menggambarkan distribusi prediksi kelas dan pola kesalahan klasifikasi yang dihasilkan oleh model pada setiap skenario dan optimizer.

4.3.1 Evaluasi Skenario 1 Semua Layer Non-trainable

Pada Skenario 1, evaluasi dilakukan secara terpisah untuk setiap optimizer yang digunakan, yaitu Adam, AdamW, dan SGD. Analisis ini bertujuan untuk mengamati perbedaan pola pembelajaran model pada kondisi seluruh layer backbone non-trainable. Evaluasi difokuskan pada nilai akurasi serta perilaku

training dan validation accuracy dan loss yang ditunjukkan melalui grafik hasil pelatihan, sebelum dilakukan perbandingan antar optimizer.berikutnya.

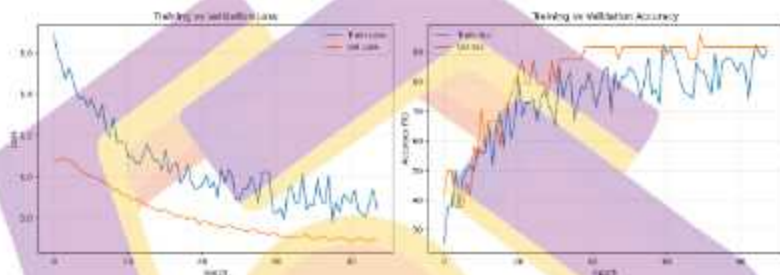
4.3.1.1 Evaluasi Skenario 1 dengan Optimizer Adam

Pada bagian ini ditampilkan hasil pelatihan model pada Skenario 1 menggunakan optimizer Adam, yang dilakukan sebanyak 10 percobaan. Tabel 4.3 memperlihatkan akurasi data testing dari masing-masing percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4. 3 Akurasi Testing 10 Percobaan Skenario 1 (Optimizer Adam)

Percobaan	Akurasi	F1-Score
Percobaan 1	73,33	71.14
Percobaan 2	76,66	76.84
Percobaan 3	76,66	76.80
Percobaan 4	83,33	82.56
Percobaan 5	73,33	71.33
Percobaan 6	80	79.90
Percobaan 7	80	78.48
Percobaan 8	70	69.17
Percobaan 9	60	59.37
Percobaan 10	73,33	71.14

Dari 10 percobaan yang dilakukan pada Skenario 1, diperoleh variasi nilai akurasi pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan dengan nilai akurasi serta F1-Score tertinggi sebagai representasi performa maksimum model. Visualisasi ini mencakup grafik training accuracy, validation accuracy, training loss, dan validation loss, ditunjukkan pada Gambar 4.2.



Gambar 4.2 Grafik Training Val Acc dan Loss Skenario 1 (Adam)

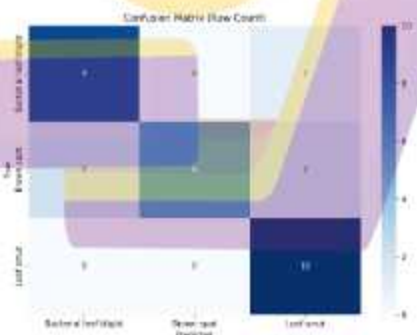
Berdasarkan grafik training dan validation loss pada Gambar 4.2 Grafik Training Val Acc dan Loss Skenario 1 (Adam), terlihat bahwa training loss menurun cukup signifikan pada fase awal pelatihan, kemudian cenderung melambat dan berfluktuasi pada epoch selanjutnya. Pola ini menunjukkan bahwa model mampu menyesuaikan bobot classifier layer, namun memiliki keterbatasan dalam menurunkan kesalahan lebih lanjut karena backbone InceptionV3 berada dalam kondisi non-trainable. Validation loss juga menunjukkan tren penurunan yang stabil tanpa peningkatan di akhir epoch, yang mengindikasikan tidak terjadinya overfitting pada skenario ini.

Pada grafik training dan validation accuracy, terlihat bahwa training accuracy meningkat secara bertahap selama proses pelatihan, sementara validation

accuracy mengalami peningkatan cepat di awal epoch dan kemudian cenderung stabil. Stabilitas validation accuracy menunjukkan bahwa model memiliki kemampuan generalisasi yang cukup baik, namun tidak mengalami peningkatan yang signifikan pada epoch akhir. Hal ini mengindikasikan bahwa performa model telah mencapai batas optimalnya pada kondisi tanpa fine-tuning backbone.

Secara keseluruhan, hasil ini menunjukkan bahwa model pada Skenario 1 mampu belajar dan melakukan generalisasi, namun dengan tingkat akurasi yang masih terbatas. Keterbatasan tersebut disebabkan oleh tidak dilakukannya pelatihan ulang pada layer backbone, sehingga fitur yang digunakan belum sepenuhnya sesuai dengan karakteristik penyakit daun padi.

Sebagai evaluasi lanjutan, confusion matrix digunakan untuk menggambarkan distribusi prediksi kelas model pada data testing, seperti ditunjukkan pada Gambar 4.3.



Gambar 4.3 Confusion matrix Skenario 1 (Adam)

Berdasarkan confusion matrix pada Gambar 4.3, model pada Skenario 1 dengan optimizer Adam mampu mengklasifikasikan 25 dari 30 data uji dengan

benar, sehingga diperoleh nilai akurasi sebesar 83,33%. Kelas Leaf smut menunjukkan performa terbaik dengan seluruh sampel berhasil diklasifikasikan secara benar. Pada kelas Bacterial leaf blight, sebagian besar sampel juga berhasil diklasifikasikan dengan benar, meskipun masih terdapat satu sampel yang salah diprediksi sebagai Leaf smut.

Kesalahan klasifikasi paling banyak terjadi pada kelas Brown spot, di mana beberapa sampel salah diklasifikasikan sebagai Bacterial leaf blight dan Leaf smut. Pola ini menunjukkan bahwa karakteristik visual Brown spot memiliki kemiripan dengan dua kelas lainnya, sehingga lebih sulit dibedakan ketika backbone model tidak dilatih ulang. Secara keseluruhan, confusion matrix ini menunjukkan bahwa model telah mampu melakukan klasifikasi dengan cukup baik, namun masih memiliki keterbatasan dalam membedakan kelas dengan pola visual yang saling tumpang tindih.

Perhitungan akurasi pada klasifikasi multi-kelas dilakukan berdasarkan jumlah prediksi benar dibandingkan jumlah keseluruhan sampel uji. Rumus akurasi yang digunakan adalah:

$$Akurasi = \frac{\text{Prediksi Benar}}{\text{Total Sampel}}$$

Dengan prediksi benar sebanyak 25 sampel dan total sampel uji sebanyak 30, diperoleh nilai akurasi:

$$Akurasi = \frac{25}{30} = 0,83$$

Berdasarkan hasil perhitungan tersebut, model pada Skenario 1 dengan optimizer Adam menghasilkan akurasi sebesar 83,33% pada data testing. Hasil ini menunjukkan bahwa model telah mampu melakukan klasifikasi dengan cukup baik, namun masih terdapat kesalahan prediksi, khususnya pada kelas Brown spot, yang mengindikasikan keterbatasan model ketika backbone InceptionV3 tidak dilatih ulang.

4.3.1.2 Evaluasi Skenario 1 dengan Optimizer AdamW

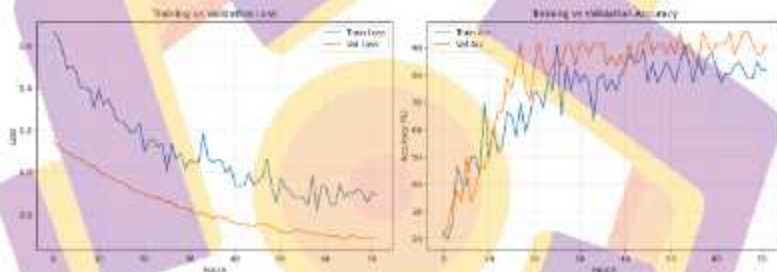
Pada bagian ini ditampilkan hasil pelatihan model pada Skenario 1 menggunakan optimizer AdamW, yang dilakukan sebanyak 10 percobaan. Tabel 4.4 memperlihatkan akurasi data testing dari masing-masing percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.4 Akurasi Testing 10 Percobaan Skenario 1 (Optimizer AdamW)

Percobaan	Akurasi	F1-Score
Percobaan 1	73,33	71.14
Percobaan 2	60	57.78
Percobaan 3	70	69.25
Percobaan 4	80	79.37
Percobaan 5	86,66	86.44
Percobaan 6	70	66.46
Percobaan 7	70	68.33
Percobaan 8	66,66	64.03
Percobaan 9	70	68.15

Percobaan	Akurasi	F1-Score
Percobaan 10	63,33	62.50

Dari 10 percobaan yang dilakukan pada Skenario 1, diperoleh variasi nilai akurasi pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan dengan nilai akurasi dan F1-Score tertinggi sebagai representasi performa maksimum model. Visualisasi ini mencakup grafik training accuracy, validation accuracy, training loss, dan validation loss, sebagaimana ditunjukkan pada Gambar 4.4.

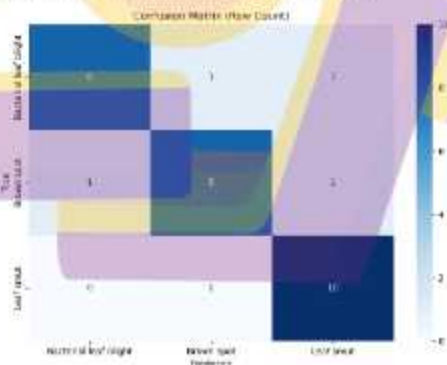


Gambar 4.4 Grafik Training Val Acc Loss Skenario 1 (AdamW)

Berdasarkan grafik training dan validation loss pada Gambar 4.4 Grafik Training Val Acc Loss Skenario 1 (AdamW), terlihat bahwa training loss mengalami penurunan yang cukup konsisten pada fase awal pelatihan, kemudian cenderung berfluktuasi pada epoch selanjutnya. Validation loss juga menunjukkan tren penurunan yang stabil dan berada di bawah training loss sepanjang proses pelatihan. Pola ini menunjukkan bahwa optimizer AdamW mampu menjaga proses optimasi berjalan stabil tanpa indikasi overfitting yang signifikan pada Skenario 1.

Pada grafik training dan validation accuracy, terlihat bahwa training accuracy meningkat secara bertahap seiring bertambahnya epoch, sementara validation accuracy menunjukkan peningkatan yang relatif cepat dan kemudian stabil pada nilai yang lebih tinggi dibandingkan training accuracy. Stabilitas validation accuracy pada kisaran nilai yang tinggi menunjukkan bahwa model memiliki kemampuan generalisasi yang cukup baik meskipun backbone InceptionV3 tidak dilatih ulang.

Secara keseluruhan, penggunaan optimizer AdamW pada Skenario 1 menghasilkan pola pembelajaran yang lebih stabil dibandingkan optimizer sebelumnya, khususnya pada validation accuracy dan validation loss. Untuk melihat distribusi prediksi kelas dan pola kesalahan klasifikasi yang dihasilkan oleh model, evaluasi lanjutan dilakukan menggunakan confusion matrix ditunjukkan pada Gambar 4.5 Confusion matrix Skenario 1 (AdamW).



Gambar 4.5 Confusion matrix Skenario 1 (AdamW)

Berdasarkan confusion matrix pada Gambar 4.5 Confusion matrix Skenario 1 (AdamW), terlihat distribusi hasil prediksi model pada Skenario 1 menggunakan optimizer AdamW terhadap tiga kelas penyakit daun padi. Nilai pada diagonal utama menunjukkan jumlah sampel yang berhasil diklasifikasikan dengan benar, sedangkan nilai di luar diagonal menunjukkan kesalahan klasifikasi antar kelas.

Pada kelas Bacterial leaf blight, sebanyak 8 dari 10 sampel berhasil diklasifikasikan dengan benar, sementara masing-masing 1 sampel salah diprediksi sebagai Brown spot dan Leaf smut. Pada kelas Brown spot, sebanyak 8 sampel berhasil diklasifikasikan dengan benar, dengan masing-masing 1 sampel salah diklasifikasikan sebagai Bacterial leaf blight dan Leaf smut. Seluruh sampel pada kelas Leaf smut berhasil diklasifikasikan dengan benar tanpa kesalahan prediksi.

Jumlah prediksi benar diperoleh dari penjumlahan nilai pada diagonal utama confusion matrix, yaitu:

Perhitungan akurasi pada klasifikasi multi-kelas dilakukan berdasarkan jumlah prediksi benar dibandingkan jumlah keseluruhan sampel uji. Rumus akurasi yang digunakan adalah:

$$Akurasi = \frac{Prediksi Benar}{Total Sampel}$$

Dengan prediksi benar sebanyak 26 sampel dan total sampel uji sebanyak 30, diperoleh nilai akurasi:

$$Akurasi = \frac{26}{30} = 0,8667$$

Berdasarkan hasil perhitungan tersebut, model pada Skenario 1 dengan optimizer AdamW menghasilkan akurasi sebesar 86,67% pada data testing. Hasil ini menunjukkan bahwa optimizer AdamW mampu memberikan performa yang lebih baik dibandingkan optimizer Adam pada kondisi seluruh layer backbone InceptionV3 berada dalam keadaan non-trainable.

Kesalahan klasifikasi masih terjadi pada kelas Bacterial leaf blight dan Brown spot, yang mengindikasikan adanya kemiripan karakteristik visual antar kedua kelas tersebut. Sementara itu, kelas Leaf smut dapat dikenali dengan sangat baik oleh model. Secara keseluruhan, hasil ini menunjukkan bahwa optimizer AdamW memberikan kemampuan generalisasi yang cukup baik pada skenario baseline dan dapat dijadikan pembanding yang kuat untuk evaluasi pada skenario fine-tuning selanjutnya

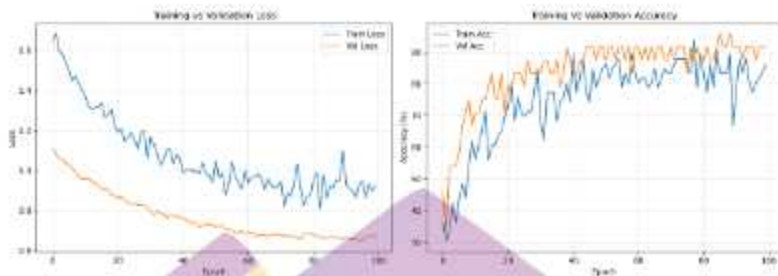
4.3.1.3 Evaluasi Skenario 1 dengan Optimizer SGD

Pada bagian ini ditampilkan hasil pelatihan model pada Skenario 1 menggunakan optimizer SGD, yang dilakukan sebanyak 10 percobaan. Tabel 4.5 memperlihatkan akurasi data testing dari masing-masing percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.5 Akurasi Testing 10 Percobaan Skenario 1 (Optimizer SGD)

Percobaan	Akurasi	F1-Score
Percobaan 1	70	66.69
Percobaan 2	73,33	72.28
Percobaan 3	86,66	86.81
Percobaan 4	66,66	66.78
Percobaan 5	73,33	72.36
Percobaan 6	76,66	74.17
Percobaan 7	63,33	59.17
Percobaan 8	83,33	83.07
Percobaan 9	73,33	69.37
Percobaan 10	70	68.29

Dari 10 percobaan yang dilakukan pada Skenario 1, diperoleh variasi nilai akurasi pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan dengan nilai akurasi dan F1-Score tertinggi sebagai representasi performa maksimum model. Visualisasi ini mencakup grafik training accuracy, validation accuracy, training loss, dan validation loss, yang ditunjukkan pada Gambar 4.6 Grafik Training Val Acc dan Loss Skenario 1 (SGD).

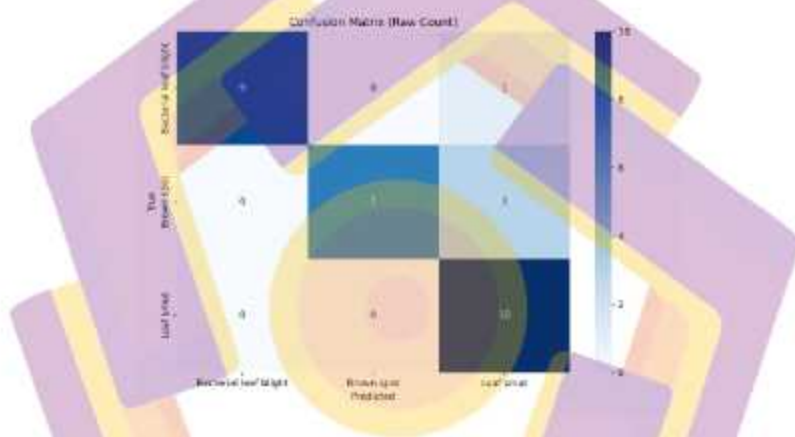


Gambar 4.6 Grafik Training Val Acc dan Loss Skenario 1 (SGD)

Berdasarkan grafik training dan validation loss pada Gambar 4.6 Grafik Training Val Acc dan Loss Skenario 1 (SGD), terlihat bahwa training loss menurun secara bertahap sejak awal pelatihan, meskipun masih menunjukkan fluktuasi pada beberapa epoch. Validation loss juga mengalami penurunan yang relatif konsisten dan berada di bawah training loss sepanjang proses pelatihan. Pola ini menunjukkan bahwa proses pelatihan dengan optimizer SGD berjalan stabil tanpa indikasi overfitting yang signifikan.

Pada grafik training dan validation accuracy, terlihat bahwa training accuracy meningkat secara bertahap seiring bertambahnya epoch, sementara validation accuracy meningkat lebih cepat pada fase awal dan kemudian cenderung stabil pada nilai yang lebih tinggi. Meskipun training accuracy menunjukkan fluktuasi, validation accuracy relatif stabil pada epoch selanjutnya, yang mengindikasikan bahwa model mampu melakukan generalisasi dengan cukup baik pada data validasi.

Secara keseluruhan, penggunaan optimizer SGD pada Skenario 1 menghasilkan proses pelatihan yang stabil dengan peningkatan akurasi yang konsisten, meskipun masih terdapat fluktuasi pada training accuracy. Untuk melihat distribusi prediksi kelas dan kesalahan klasifikasi yang dihasilkan oleh model pada data testing, evaluasi lanjutan dilakukan menggunakan confusion matrix sebagaimana ditunjukkan pada Gambar 4.7 Confusion matrix Skenario 1 (SGD).



Gambar 4.7 Confusion matrix Skenario 1 (SGD)

Berdasarkan confusion matrix pada Gambar 4.7 Confusion matrix Skenario 1 (SGD), terlihat distribusi hasil prediksi model pada Skenario 1 menggunakan optimizer SGD terhadap tiga kelas penyakit daun padi. Nilai pada diagonal utama menunjukkan jumlah prediksi yang benar, sedangkan nilai di luar diagonal menunjukkan kesalahan klasifikasi antar kelas.

Pada kelas Bacterial leaf blight, sebanyak 9 dari 10 sampel berhasil diklasifikasikan dengan benar, sementara 1 sampel salah diklasifikasikan sebagai Leaf smut. Pada kelas Brown spot, sebanyak 7 sampel berhasil diklasifikasikan

dengan benar, sedangkan 3 sampel salah diprediksi sebagai Leaf smut. Seluruh sampel pada kelas Leaf smut berhasil diklasifikasikan dengan benar tanpa adanya kesalahan prediksi.

Jumlah prediksi benar diperoleh dari penjumlahan nilai pada diagonal utama confusion matrix, yaitu :

$$Akurasi = \frac{\text{Prediksi Benar}}{\text{Total Sampel}}$$

Dengan prediksi benar sebanyak 26 sampel dan total sampel uji sebanyak 30, diperoleh nilai akurasi:

$$Akurasi = \frac{26}{30} = 0,8667$$

Berdasarkan hasil perhitungan tersebut, model pada Skenario 1 dengan optimizer SGD menghasilkan akurasi sebesar 86,67% pada data testing dengan kondisi seluruh layer backbone InceptionV3 bersifat non-trainable.

Kesalahan klasifikasi paling banyak terjadi pada kelas Brown spot yang cenderung salah diprediksi sebagai Leaf smut. Hal ini menunjukkan bahwa karakteristik visual kedua kelas tersebut memiliki kemiripan yang sulit dibedakan ketika fitur yang digunakan berasal dari backbone yang tidak dilatih ulang. Sementara itu, kelas Leaf smut dapat dikenali dengan sangat baik oleh model, yang mengindikasikan bahwa ciri visual kelas tersebut cukup konsisten dan mudah ditangkap oleh classifier layer.

Secara keseluruhan, hasil ini menunjukkan bahwa penggunaan optimizer SGD pada kondisi non-trainable backbone mampu menghasilkan akurasi yang cukup baik, namun masih memiliki keterbatasan dalam membedakan kelas dengan karakteristik visual yang saling tumpang tindih. Temuan ini menjadi dasar untuk mengevaluasi peningkatan performa pada skenario dengan beberapa layer trainable dan full fine-tuning pada tahap selanjutnya.

4.3.2 Evaluasi Skenario 2 – Fine Tuning Layer Awal

Pada bagian ini dilakukan evaluasi terhadap Skenario 2, yaitu penerapan fine-tuning pada beberapa layer awal model InceptionV3, sementara layer lainnya tetap bersifat non-trainable. Evaluasi difokuskan untuk melihat pengaruh peningkatan jumlah layer awal yang dilatih terhadap performa model pada dataset penyakit daun padi.

Evaluasi pada skenario ini dibagi berdasarkan optimizer yang digunakan, yaitu Adam, AdamW, dan SGD. Untuk setiap optimizer, dilakukan pengujian dengan lima variasi jumlah layer awal yang bersifat trainable, yaitu 150 layer, dan 250 layer. Setiap variasi dievaluasi menggunakan grafik training dan validation accuracy serta training dan validation loss untuk mengamati proses pelatihan model. Selain itu, performa akhir model juga dievaluasi pada data testing untuk melihat hasil klasifikasi yang dihasilkan pada setiap konfigurasi layer.

4.3.2.1 Evaluasi Skenario 2 dengan Optimizer Adam

Pada bagian ini dibahas evaluasi Skenario 2 menggunakan optimizer Adam, dengan penerapan fine-tuning pada beberapa layer awal model InceptionV3.

Evaluasi dilakukan melalui variasi jumlah layer awal yang bersifat trainable, yaitu 150 layer, dan 250 layer, sementara layer lainnya tetap non-trainable. Setiap variasi dievaluasi menggunakan grafik training dan validation accuracy serta training dan validation loss, untuk melihat pengaruh jumlah layer awal yang dilatih terhadap proses pelatihan dan performa model.

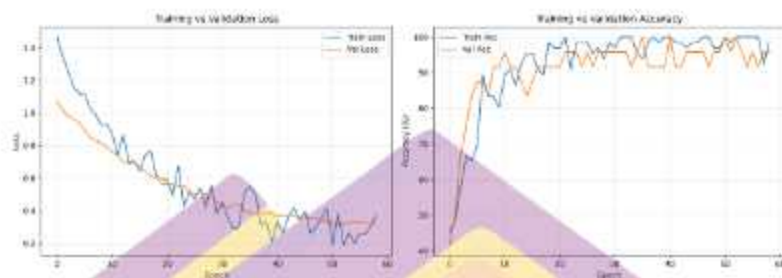
A. 150 Layer Trainable

Pada bagian ini ditampilkan hasil pelatihan model pada Skenario 2 pada layer awal trainable menggunakan optimizer Adam, yang dilakukan sebanyak 10 percobaan. Tabel 4.6 memperlihatkan akurasi data testing dari masing-masing percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.6 Akurasi Testing 10 Percobaan Skenario 2 layer 150 (Optimizer Adam)

Percobaan	Akurasi	F1-Score
Percobaan 1	73,33	72,80
Percobaan 2	76,66	74,10
Percobaan 3	93,33	93,12
Percobaan 4	80	78,39
Percobaan 5	80	80,38
Percobaan 6	83,33	82,78
Percobaan 7	83,33	83,30
Percobaan 8	80	79,89
Percobaan 9	93,33	93,12
Percobaan 10	86,66	86,81

Dari 10 percobaan yang dilakukan pada Skenario 2, diperoleh variasi nilai akurasi pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan



Gambar 4.8 Grafik Layer 150 Training Val Acc dan Loss Skenario 2 (Adam)

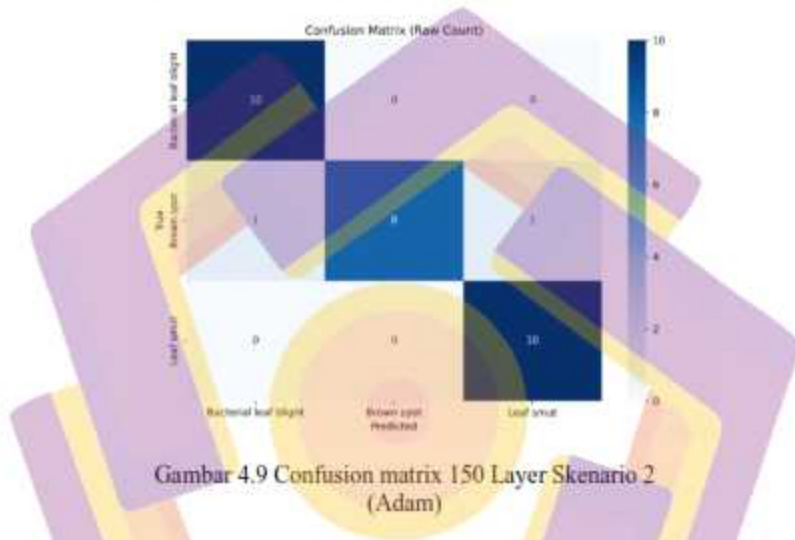
Berdasarkan grafik pada Gambar 4.8, training dan validation loss, training loss menurun tajam pada epoch awal hingga sekitar epoch 15–20, kemudian menurun lebih lambat dengan fluktuasi kecil. Validation loss juga menunjukkan penurunan yang konsisten dan cenderung stabil setelah epoch sekitar 25, tanpa adanya penurunan yang signifikan pada epoch berikutnya.

Pada grafik training dan validation accuracy, training accuracy mencapai nilai mendekati 100% sekitar epoch 20 dan cenderung stabil hingga akhir pelatihan. Validation accuracy juga mencapai nilai tinggi pada rentang epoch yang sama dan tidak menunjukkan peningkatan yang berarti setelahnya. Kondisi ini menunjukkan bahwa model telah mencapai performa optimal lebih awal.

Meskipun jumlah epoch maksimum ditetapkan sebesar 100, proses pelatihan dihentikan lebih awal pada sekitar epoch 70 akibat penerapan early stopping. Early stopping menghentikan pelatihan ketika tidak terjadi peningkatan performa validation loss dan validation accuracy dalam beberapa epoch berturut-

turut, sehingga pelatihan tidak dilanjutkan hingga epoch maksimum yang ditetapkan.

Untuk melihat dampak konfigurasi ini terhadap hasil klasifikasi pada data testing, evaluasi selanjutnya dilakukan menggunakan confusion matrix pada Gambar 4.9.



Confusion matrix menunjukkan hasil klasifikasi pada data uji untuk tiga kelas penyakit daun padi. Kelas Bacterial leaf blight seluruhnya terklasifikasi dengan benar sebanyak 10 data tanpa kesalahan. Kelas Brown spot memiliki 8 data yang terklasifikasi benar, sedangkan 2 data mengalami kesalahan prediksi, masing-masing satu diprediksi sebagai Bacterial leaf blight dan satu sebagai Leaf smut. Kelas Leaf smut juga terklasifikasi sempurna dengan 10 data benar dan tanpa kesalahan. Pola ini menunjukkan bahwa kesalahan prediksi hanya terjadi pada kelas Brown spot, sedangkan dua kelas lainnya dapat dikenali dengan sangat baik oleh model.

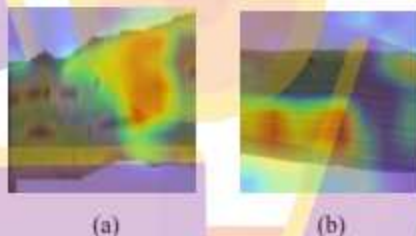
Perhitungan akurasi dilakukan menggunakan rumus akurasi multi-kelas:

$$Akurasi = \frac{Prediksi\ Benar}{Total\ Sampel}$$

Dengan nilai prediksi benar (TP) = 28 dan total sampel uji = 30, diperoleh:

$$Akurasi = \frac{28}{30} = 0,9333$$

Berdasarkan hasil evaluasi, pada data uji terdapat sampel yang mengalami kesalahan klasifikasi dan seluruhnya berasal dari kelas Brown spot. Seluruh sampel misclassification tersebut divisualisasikan menggunakan Grid-CAM untuk dianalisis lebih lanjut. Visualisasi ini bertujuan untuk mengamati area atensi model ketika menghasilkan prediksi yang tidak sesuai dengan label sebenarnya.



Gambar 4. 10 Grid-CAM misclassification kelas Brown spot: (a) Leaf smut, (b) Bacterial leaf blight.

Gambar menampilkan visualisasi Grid-CAM pada dua sampel data uji yang mengalami kesalahan klasifikasi, di mana keduanya berasal dari kelas Brown spot namun diprediksi sebagai kelas lain. Pada Gambar (a), Area yang menjadi perhatian model didominasi oleh bercak berwarna coklat kehitaman dengan pola memanjang, yang secara visual menyerupai karakteristik penyakit Leaf Smut, sehingga citra

Brown Spot tersebut diprediksi sebagai Leaf Smut. Pola ini sejalan dengan ciri visual Leaf Smut yang umumnya ditandai oleh bercak gelap memanjang dan menyebar mengikuti arah tulang daun, sehingga pada kondisi tertentu sulit dibedakan secara visual. Untuk memperjelas kemiripan tersebut, ditampilkan citra daun dengan true label Leaf Smut sebagai pembanding pada Gambar 4.11.



Gambar 4.11 Citra daun padi dengan true label Leaf Smut.

Sementara itu, pada Gambar (b) perhatian model lebih terarah pada area memanjang pada permukaan daun yang relatif homogen dan berwarna hijau, yang bukan merupakan area bercak penyakit utama. Fokus model pada area yang kurang representatif terhadap ciri visual Brown Spot ini menyebabkan model gagal menangkap karakteristik penyakit yang sebenarnya, sehingga citra tersebut keliru diprediksi sebagai Bacterial Leaf Blight.

B. 250 Layer Trainable

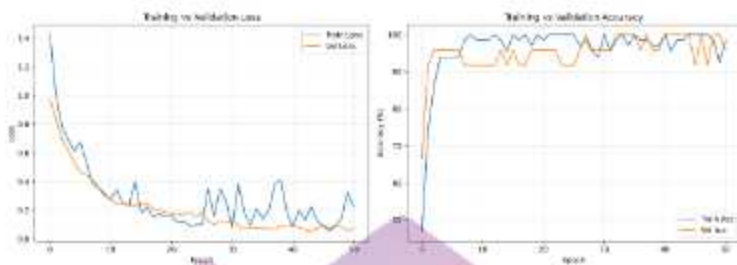
Pada tahap ini dilakukan evaluasi Skenario 2 dengan menerapkan fine-tuning pada 250 layer awal model InceptionV3 menggunakan optimizer Adam, yang dilakukan sebanyak 10 percobaan. sementara layer lainnya tetap bersifat non-trainable. Tabel 4.7 memperlihatkan akurasi data testing dari masing-masing

percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.7 Akurasi Testing 10 Percobaan Skenario 2 layer 250 (Optimizer Adam)

Percobaan	Akurasi	F1-Score
Percobaan 1	86,66	86,44
Percobaan 2	93,33	93,12
Percobaan 3	70	68,36
Percobaan 4	80	80,38
Percobaan 5	86,66	86,33
Percobaan 6	83,33	82,55
Percobaan 7	90	89,93
Percobaan 8	93,33	93,46
Percobaan 9	73,33	71,22
Percobaan 10	96,66	96,66

Dari 10 percobaan yang dilakukan pada Skenario 2, diperoleh variasi nilai akurasi pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan dengan nilai akurasi dan F1-Score tertinggi sebagai representasi performa maksimum model. Ditampilkan grafik training dan validation accuracy serta training dan validation loss pada Gambar 4.12.



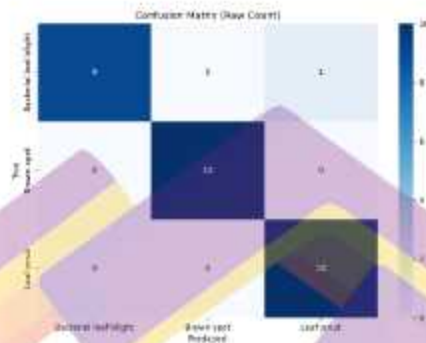
Gambar 4.12 Grafik Layer 250 Training Val Acc dan Loss Skenario 2 (Adam)

Berdasarkan grafik pada Gambar 4.12, proses pelatihan menunjukkan konvergensi yang lebih cepat dibandingkan konfigurasi layer sebelumnya. Nilai loss training dan loss validasi menurun tajam pada awal pelatihan dan mulai stabil setelah sekitar epoch 20. Selisih antara loss training dan validasi relatif kecil, yang menunjukkan proses pembelajaran berjalan stabil tanpa indikasi overfitting yang kuat.

Akurasi training meningkat sangat cepat dan mencapai nilai mendekati maksimum sejak epoch awal. Akurasi validasi juga mengikuti pola serupa dan stabil di kisaran tinggi hingga akhir pelatihan. Meskipun batas maksimum pelatihan ditetapkan sebesar 100 epoch, mekanisme early stopping menghentikan proses pelatihan lebih awal, yaitu sekitar epoch 50, karena tidak terdapat peningkatan signifikan pada performa validasi setelah titik tersebut. Hal ini menunjukkan bahwa model telah mencapai kondisi optimal sebelum seluruh epoch terpenuhi.

Pola perubahan loss dan akurasi selama proses pelatihan dan validasi dapat dilihat pada grafik hasil training dan validasi pada gambar di atas. Untuk menilai

performa klasifikasi secara lebih rinci pada data uji, hasil evaluasi confusion matrix ditunjukkan pada Gambar 4.13.



Gambar 4.13 Confusion matrix 250 Layer Skenario 2 (Adam)

Berdasarkan confusion matrix pada Gambar 4.13, kelas Bacterial leaf blight terklasifikasi benar sebanyak 9 dari 10 citra, dengan 1 citra salah diprediksi sebagai kelas Leaf smut. Kelas Brown spot seluruhnya terklasifikasi dengan benar sebanyak 10 dari 10 citra tanpa kesalahan. Demikian pula, seluruh citra pada kelas Leaf smut berhasil diklasifikasikan dengan benar sebanyak 10 dari 10 citra. Hasil ini menunjukkan bahwa kesalahan klasifikasi hanya terjadi pada kelas Bacterial leaf blight, sementara kelas Brown spot dan Leaf smut dapat dikenali dengan sangat baik oleh model.

Perhitungan akurasi dilakukan menggunakan rumus akurasi multi-kelas:

$$Akurasi = \frac{Prediksi Benar}{Total Sampel}$$

Dengan nilai prediksi benar (TP) = 29 dan total sampel uji = 30, diperoleh:

$$Akurasi = \frac{29}{30} = 0,96$$

Berdasarkan hasil evaluasi, pada data uji kelas Bacterial leaf blight terdapat sampel yang mengalami kesalahan klasifikasi. Visualisasi Grid-CAM digunakan untuk menganalisis area atensi model pada prediksi yang tidak sesuai dengan label sebenarnya.



Gambar-4.14 Grid-CAM misclassification Bacterial leaf blight

Pada Gambar 4.14 menampilkan visualisasi Grid-CAM pada data uji dengan label sebenarnya Bacterial Leaf Blight yang mengalami kesalahan klasifikasi dan diprediksi sebagai kelas Leaf Smut. Hasil visualisasi menunjukkan bahwa perhatian model menyebar pada area yang relatif luas di sepanjang permukaan daun dan tidak terfokus secara spesifik pada bercak khas penyakit. Fokus yang kurang terarah ini membuat model lebih menekankan pola umum berbentuk memanjang pada daun, yang secara visual menyerupai karakteristik Leaf Smut, sehingga ciri khas Bacterial Leaf Blight tidak tertangkap dengan baik dan menghasilkan prediksi yang tidak sesuai dengan label sebenarnya.

4.3.2.2 Evaluasi Skenario 2 dengan Optimizer AdamW

Pada bagian ini dilakukan evaluasi Skenario 2 dengan menerapkan fine tuning pada beberapa layer awal model InceptionV3 menggunakan optimizer AdamW, sementara layer lainnya tetap bersifat non-trainable. Penggunaan AdamW bertujuan untuk menguji pengaruh mekanisme weight decay yang lebih terkontrol terhadap proses pembelajaran model, khususnya pada kondisi dataset yang berukuran terbatas.

A. 150 Layer Trainable

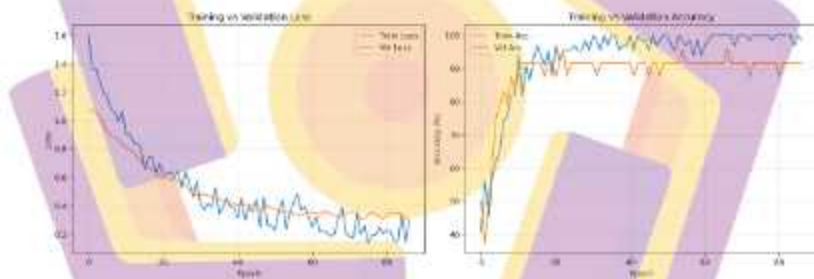
Pada tahap ini dilakukan evaluasi Skenario 2 dengan menerapkan fine-tuning pada 150 layer awal model InceptionV3 menggunakan optimizer AdamW, yang dilakukan sebanyak 10 percobaan, sementara layer lainnya tetap bersifat non-trainable. Tabel 4.8 memperlihatkan akurasi data testing dari masing-masing percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.8 Akurasi Testing 10 Percobaan Skenario 2 layer 150 (Optimizer AdamW)

Percobaan	Akurasi	F1-Score
Percobaan 1	76,66	76.67
Percobaan 2	86,66	86.33
Percobaan 3	80	78.57
Percobaan 4	83,33	83.33
Percobaan 5	83,33	83.64
Percobaan 6	83,33	83.31
Percobaan 7	86,66	86.77

Percobaan	Akurasi	F1-Score
Percobaan 8	83,33	82.40
Percobaan 9	80	79.89
Percobaan 10	76,66	74.12

Dari 10 percobaan yang dilakukan pada Skenario 2, diperoleh variasi nilai akurasi pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan dengan nilai akurasi dan F1-Score tertinggi sebagai representasi performa maksimum model. Ditampilkan grafik training dan validation accuracy serta training dan validation loss pada Gambar 4.15.



Gambar 4.15 Grafik Layer 150 Training Val Acc dan Loss Skenario 2 (AdamW)

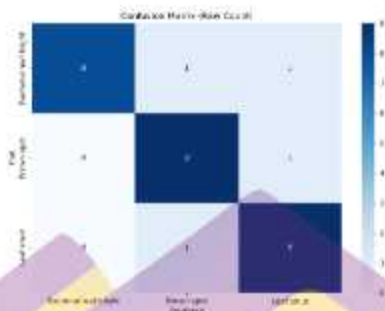
Grafik menunjukkan bahwa loss training dan loss validasi mengalami penurunan yang stabil sejak awal pelatihan. Loss training turun dari kisaran 1,6 dan mulai berada di bawah 0,5 setelah sekitar epoch 30. Loss validasi mengikuti pola serupa dan mulai stabil di kisaran 0,35–0,40 sejak sekitar epoch 40. Pola ini

menunjukkan proses pembelajaran berjalan konsisten dan tidak terjadi lonjakan loss yang signifikan.

Pada grafik akurasi, akurasi training meningkat cepat pada fase awal dan mencapai nilai di atas 90% sebelum epoch 15. Setelah itu akurasi training terus meningkat hingga mendekati 100% dan relatif stabil. Akurasi validasi juga meningkat dengan cepat dan bertahan di kisaran 90–93% sejak sekitar epoch 20. Selisih antara akurasi training dan validasi tetap kecil, yang menunjukkan generalisasi model masih terjaga meskipun jumlah layer trainable ditingkatkan.

Meskipun batas maksimum pelatihan ditetapkan hingga 100 epoch, proses training dihentikan lebih awal oleh mekanisme early stopping pada sekitar epoch 85. Early stopping terjadi karena performa validasi tidak lagi menunjukkan peningkatan yang berarti dalam beberapa epoch berturut-turut. Hal ini mengindikasikan bahwa model telah mencapai kondisi optimal dan pelatihan lanjutan berpotensi meningkatkan risiko overfitting.

Untuk melihat distribusi kesalahan klasifikasi pada masing-masing kelas penyakit daun padi secara lebih detail, hasil evaluasi model selanjutnya ditunjukkan pada confusion matrix pada Gambar 4.16.



Gambar 4.16 Confusion matrix 150 Layer Skenario 2 (AdamW)

Confusion matrix menunjukkan bahwa sebagian besar data uji berhasil diklasifikasikan dengan benar. Pada kelas Bacterial leaf blight, terdapat 8 data yang terklasifikasi tepat, sementara 2 data salah klasifikasi masing-masing ke kelas Brown spot dan Leaf smut. Pada kelas Brown spot, sebanyak 9 data berhasil diprediksi dengan benar dan 1 data salah ke kelas Leaf smut. Pada kelas Leaf smut, sebanyak 9 data terklasifikasi dengan benar dan 1 data salah ke kelas Brown spot. Kesalahan klasifikasi terlihat relatif kecil dan tersebar, tanpa dominasi kesalahan pada satu kelas tertentu.

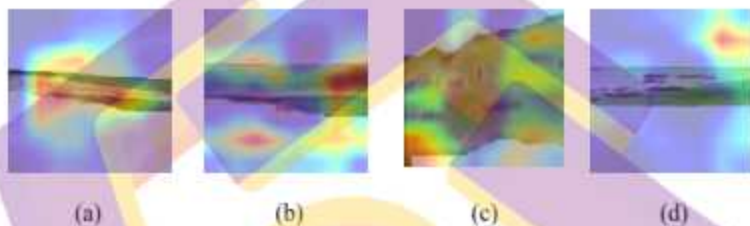
Perhitungan akurasi dilakukan menggunakan rumus akurasi multi-kelas:

$$Akurasi = \frac{Prediksi\ Benar}{Total\ Sampel}$$

Dengan nilai prediksi benar (TP) = 27 dan total sampel uji = 30, diperoleh:

$$Akurasi = \frac{27}{30} = 0,86$$

Berdasarkan hasil evaluasi, pada data uji terdapat sampel yang mengalami kesalahan klasifikasi dan seluruhnya berasal dari kelas Brown spot. Seluruh sampel misclassification tersebut divisualisasikan menggunakan Grid-CAM untuk dianalisis lebih lanjut. Visualisasi ini bertujuan untuk mengamati area atensi model ketika menghasilkan prediksi yang tidak sesuai dengan label sebenarnya.



Gambar 4.17 Grid-CAM misclassification: (a) Bacterial leaf blight, (b) Bacterial leaf blight, (c) Brown spot, (d) Leaf smut.

Gambar 4.17 menampilkan visualisasi Grid-CAM pada beberapa sampel data uji yang mengalami kesalahan klasifikasi. Pada Gambar (a), citra dengan label sebenarnya Bacterial Leaf Blight salah diprediksi sebagai Brown Spot, di mana area atensi model cenderung menyebar pada tekstur permukaan daun secara umum dan tidak terlokalisasi pada ciri khas penyakit. Gambar (b) menunjukkan sampel Bacterial leaf blight yang diprediksi sebagai Leaf smut, di mana aktivasi model lebih menonjol pada pola memanjang pada daun. Pada Gambar (c), Area perhatian model didominasi oleh bercak coklat kehitaman berpola memanjang yang menyerupai karakteristik Leaf Smut, sehingga citra Brown Spot diprediksi sebagai Leaf Smut. Untuk memperjelas kemiripan tersebut, ditampilkan citra daun dengan true label Leaf Smut sebagai pembandingan pada Gambar 4.18.



Gambar 4.18 Citra daun padi dengan true label Leaf Smut.

Sementara itu, pada Gambar (d), area atensi model justru lebih dominan pada bagian latar belakang, bukan pada permukaan daun, yang mengindikasikan bahwa informasi non-relevan turut memengaruhi proses prediksi dan menyebabkan kesalahan klasifikasi.

B. 250 Layer Trainable

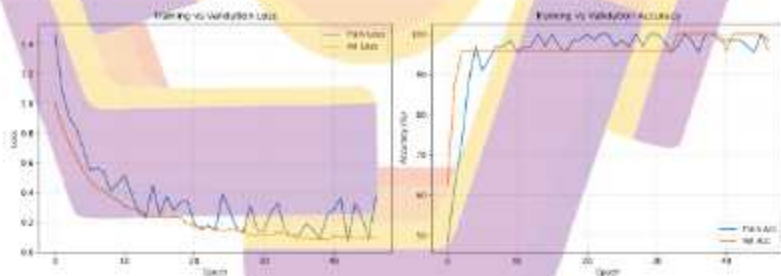
Pada tahap ini dilakukan evaluasi Skenario 2 dengan menerapkan fine-tuning pada 250 layer awal model InceptionV3 menggunakan optimizer AdamW, yang dilakukan sebanyak 10 percobaan, sementara layer lainnya tetap bersifat non-trainable. Tabel 4.9 memperlihatkan akurasi data testing dari masing-masing percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.9 Akurasi Testing 10 Percobaan Skenario 2 layer 250 (Optimizer AdamW)

Percobaan	Akurasi	F1-Score
Percobaan 1	83,33	81.75
Percobaan 2	86,66	86.53
Percobaan 3	80	78.57
Percobaan 4	76,66	76.29

Percobaan	Akurasi	F1-Score
Percobaan 5	90	89.93
Percobaan 6	83,33	82.78
Percobaan 7	93,33	93.32
Percobaan 8	90	89.50
Percobaan 9	93,33	93.33
Percobaan 10	90	89.93

Dari 10 percobaan yang dilakukan pada Skenario 2, diperoleh variasi nilai akurasi dan F1-Score pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan dengan nilai akurasi dan F1-Score tertinggi sebagai representasi performa maksimum model. Ditampilkan grafik training dan validation accuracy serta training dan validation loss pada Gambar 4.19.



Gambar 4.19 Grafik Layer 250 Training Val Acc dan Loss Skenario 2 (AdamW)

Grafik pada Gambar 4.19 menunjukkan bahwa nilai loss training dan validasi menurun tajam pada fase awal pelatihan. Loss training turun dari sekitar 1,4 dan mencapai nilai di bawah 0,3 sebelum epoch 15. Loss validasi mengikuti

pola serupa dan mulai stabil di kisaran 0,1–0,15 setelah epoch 20. Pola ini menunjukkan proses pembelajaran berlangsung efektif dan cepat mencapai kondisi konvergen.

Pada grafik akurasi, akurasi validasi meningkat sangat cepat dan telah mencapai sekitar 95% pada epoch awal, kemudian stabil hingga mendekati 100% pada epoch-epoch berikutnya. Akurasi training juga meningkat konsisten dan berada pada kisaran 95–100% hampir sepanjang pelatihan. Jarak antara akurasi training dan validasi relatif kecil, menandakan generalisasi model yang baik meskipun jumlah layer trainable cukup besar.

Meskipun jumlah maksimum epoch ditetapkan sebanyak 100, mekanisme early stopping menghentikan proses pelatihan lebih awal, yaitu sekitar epoch 44–46. Early stopping terjadi karena performa validasi sudah stabil dan tidak menunjukkan peningkatan yang signifikan dalam beberapa epoch berturut-turut. Hal ini mengindikasikan bahwa pelatihan lanjutan tidak diperlukan dan justru berpotensi meningkatkan risiko overfitting.

Untuk melihat pola kesalahan klasifikasi pada masing-masing kelas penyakit daun padi secara lebih rinci, hasil evaluasi model selanjutnya ditunjukkan pada confusion matrix pada Gambar 4.20.



Gambar 4.20 Confusion matrix 250 Layer Skenario 2 (AdamW)

Confusion matrix menunjukkan bahwa model mampu mengklasifikasikan sebagian besar citra daun padi dengan benar pada setiap kelas. Pada kelas Bacterial leaf blight, seluruh 10 data uji berhasil diprediksi dengan benar tanpa kesalahan. Pada kelas Brown spot, terdapat 9 data yang terklasifikasi dengan benar dan 1 data salah diprediksi sebagai Leaf smut. Pada kelas Leaf smut, sebanyak 9 data terklasifikasi dengan benar dan 1 data salah diprediksi sebagai Brown spot. Kesalahan klasifikasi yang terjadi bersifat kecil dan hanya terjadi pada dua kelas yang memiliki karakteristik visual yang relatif mirip.

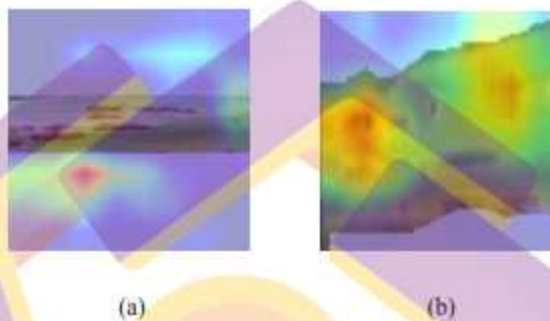
Perhitungan akurasi dilakukan menggunakan rumus akurasi multi-kelas:

$$Akurasi = \frac{Prediksi\ Benar}{Total\ Sampel}$$

Dengan nilai prediksi benar (TP) = 28 dan total sampel uji = 30, diperoleh:

$$Akurasi = \frac{28}{30} = 0,9333$$

Berdasarkan hasil evaluasi, pada data uji terdapat sampel yang mengalami kesalahan klasifikasi pada kelas Brown spot dan Leaf smut. Visualisasi Grid-CAM digunakan untuk menganalisis area atensi model pada prediksi yang tidak sesuai dengan label sebenarnya.



Gambar 4.21 Grid-CAM misclassification: (a) Brown spot, (b) Leaf smut

Gambar 4.21 menampilkan visualisasi Grid-CAM pada dua sampel data uji yang mengalami kesalahan klasifikasi. Pada Gambar (a), citra dengan label sebenarnya Brown spot diprediksi sebagai Leaf smut, dengan area atensi model yang cenderung menyebar dan tidak terlokalisasi pada bercak penyakit utama.

Sementara itu, pada Gambar (b), Area perhatian model didominasi oleh bercak coklat kehitaman berpola memanjang yang menyerupai karakteristik Leaf Smut, sehingga citra Brown Spot diprediksi sebagai Leaf Smut. Untuk memperjelas kemiripan tersebut, ditampilkan citra daun dengan true label Leaf Smut sebagai pembanding pada Gambar 4.22.



Gambar 4.22 Citra daun padi dengan true label Leaf Smut.

4.3.2.3 Evaluasi Skenario 2 dengan Optimizer SGD

Pada tahap ini dilakukan evaluasi Skenario 2 dengan menerapkan fine-tuning pada model InceptionV3 menggunakan optimizer SGD, dengan variasi jumlah layer awal yang dijadikan trainable. Konfigurasi yang diuji meliputi fine-tuning pada 150 layer awal, hingga 250 layer awal, sementara layer lainnya tetap bersifat non-trainable. Variasi ini bertujuan untuk mengamati pengaruh kedalaman fine-tuning terhadap kemampuan model dalam mempelajari pola penyakit daun padi.

Proses pelatihan dijalankan dengan batas maksimum 100 epoch dan dilengkapi mekanisme early stopping. Early stopping diterapkan karena pada dataset berukuran kecil, pelatihan yang terlalu panjang sering kali tidak lagi meningkatkan performa validasi dan berpotensi menyebabkan overfitting. Dengan mekanisme ini, proses pelatihan dihentikan ketika tidak terjadi peningkatan kinerja validasi dalam beberapa epoch berturut-turut, peningkatan yang signifikan. Evaluasi pada skenario ini difokuskan pada pengamatan perubahan nilai loss dan akurasi pada data training dan validasi, serta dampaknya terhadap performa klasifikasi pada data uji.

A. 150 Layer Trainable

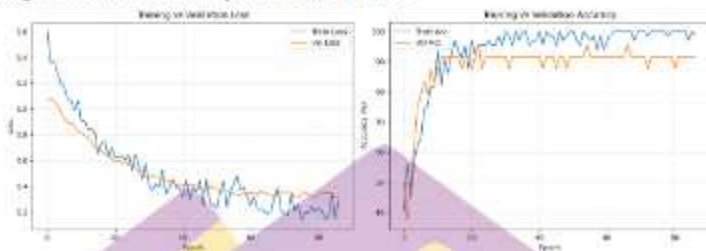
Pada tahap ini dilakukan evaluasi Skenario 2 dengan menerapkan fine-tuning pada 150 layer awal model InceptionV3 menggunakan optimizer SGD, yang dilakukan sebanyak 10 percobaan. sementara layer lainnya tetap bersifat non-trainable. Tabel 4.10 memperlihatkan akurasi data testing dari masing-masing percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.10 Akurasi Testing-10 Percobaan Skenario 2 layer-150 (Optimizer SGD)

Percobaan	Akurasi	F1-Score
Percobaan 1	70	68.29
Percobaan 2	66,66	65.97
Percobaan 3	80	79.80
Percobaan 4	73,33	72.89
Percobaan 5	66,66	66.89
Percobaan 6	76,66	76.29
Percobaan 7	80	80.08
Percobaan 8	70	68.11
Percobaan 9	66,66	63.56
Percobaan 10	70	66.59

Dari 10 percobaan yang dilakukan pada Skenario 2, diperoleh variasi nilai akurasi pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan dengan nilai akurasi dan F1-Score tertinggi sebagai representasi performa

maksimum model. Ditampilkan grafik training dan validation accuracy serta training dan validation loss pada Gambar 4.23.



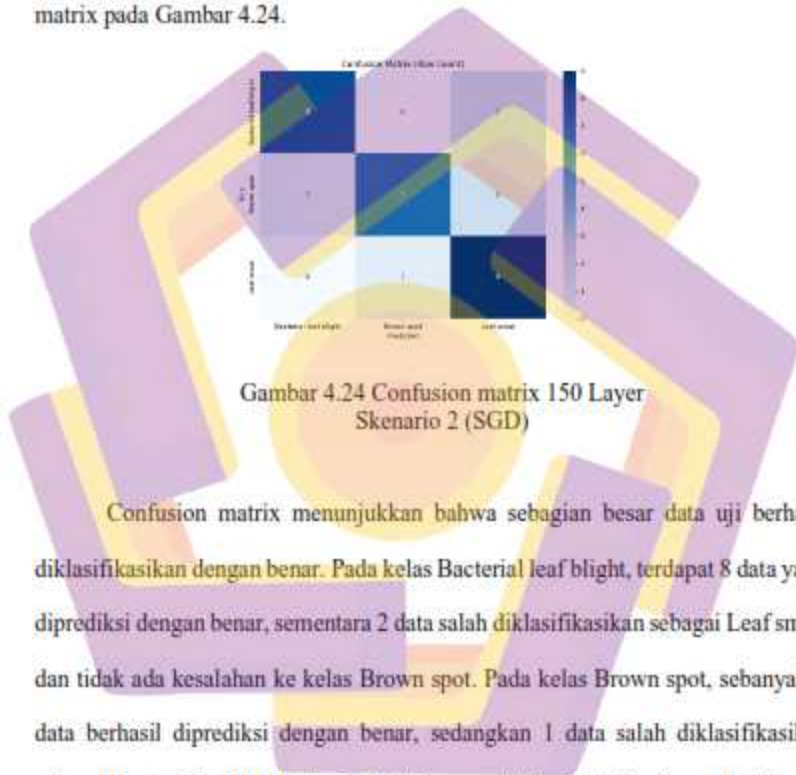
Gambar 4.23 Grafik Layer 150 Training Val Acc dan Loss Skenario 2 (SGD)

Grafik menunjukkan bahwa loss training dan loss validasi mengalami penurunan yang stabil sejak awal pelatihan. Loss training turun dari kisaran 1,6 dan mulai berada di bawah 0,5 setelah sekitar epoch 30. Loss validasi mengikuti pola serupa dan mulai stabil di kisaran 0,35–0,40 sejak sekitar epoch 40. Pola ini menunjukkan proses pembelajaran berjalan konsisten dan tidak terjadi lonjakan loss yang signifikan.

Pada grafik akurasi, akurasi training meningkat cepat pada fase awal dan mencapai nilai di atas 90% sebelum epoch 15. Setelah itu akurasi training terus meningkat hingga mendekati 100% dan relatif stabil. Akurasi validasi juga meningkat dengan cepat dan bertahan di kisaran 90–93% sejak sekitar epoch 20. Selisih antara akurasi training dan validasi tetap kecil, yang menunjukkan generalisasi model masih terjaga meskipun jumlah layer trainable ditingkatkan.

Meskipun batas maksimum pelatihan ditetapkan hingga 100 epoch, proses training dihentikan lebih awal oleh mekanisme early stopping pada sekitar epoch

85. Early stopping terjadi karena performa validasi tidak lagi menunjukkan peningkatan yang berarti dalam beberapa epoch berturut-turut. Untuk melihat distribusi kesalahan klasifikasi pada masing-masing kelas penyakit daun padi secara lebih detail, hasil evaluasi model selanjutnya ditunjukkan pada confusion matrix pada Gambar 4.24.



Gambar 4.24 Confusion matrix 150 Layer Skenario 2 (SGD)

Confusion matrix menunjukkan bahwa sebagian besar data uji berhasil diklasifikasikan dengan benar. Pada kelas Bacterial leaf blight, terdapat 8 data yang diprediksi dengan benar, sementara 2 data salah diklasifikasikan sebagai Leaf smut, dan tidak ada kesalahan ke kelas Brown spot. Pada kelas Brown spot, sebanyak 7 data berhasil diprediksi dengan benar, sedangkan 1 data salah diklasifikasikan sebagai Bacterial leaf blight dan 2 data lainnya salah ke kelas Leaf smut. Pada kelas Leaf smut, terdapat 9 data yang terklasifikasi dengan benar, dan hanya 1 data yang salah diklasifikasikan sebagai Brown spot, tanpa kesalahan ke kelas Bacterial leaf blight.

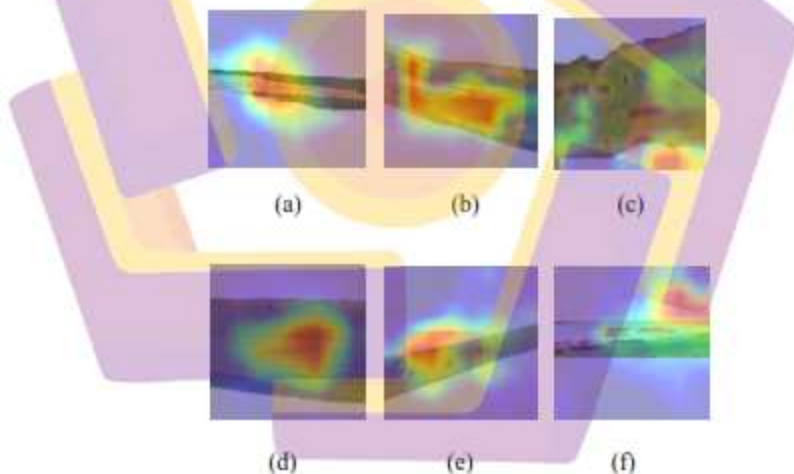
Perhitungan akurasi dilakukan menggunakan rumus akurasi multi-kelas:

$$Akurasi = \frac{Prediksi\ Benar}{Total\ Sampel}$$

Dengan nilai prediksi benar (TP) = 24 dan total sampel uji = 30, diperoleh:

$$Akurasi = \frac{24}{30} = 0,80$$

Berdasarkan hasil evaluasi, pada data uji terdapat sampel yang mengalami kesalahan klasifikasi pada kelas Bacterial leaf blight, Brown spot, dan Leaf smut. Visualisasi Grid-CAM digunakan untuk menganalisis area atensi model pada prediksi yang tidak sesuai dengan label sebenarnya.



Gambar 4.25 Grid-CAM misclassification: (a) Bacterial leaf blight, (b) Bacterial leaf blight, (c) Brown Spot, (d) Brown spot, (e) Brown spot, (f) Leaf smut.

Gambar 4.25 menampilkan visualisasi Grid-CAM pada beberapa sampel data uji yang mengalami kesalahan klasifikasi. Pada Gambar (a) dan (b), citra

dengan label sebenarnya Bacterial leaf blight diprediksi sebagai Leaf smut, di mana area atensi model lebih terfokus pada pola memanjang pada permukaan daun yang menyerupai karakteristik kelas tersebut. Pada Gambar (c) dan (d), citra kelas Brown spot diprediksi sebagai Leaf smut, dengan aktivasi model yang cenderung menyebar dan tidak sepenuhnya terlokalisasi pada bercak penyakit utama. Selanjutnya, pada Gambar (e), citra Brown spot diprediksi sebagai Bacterial leaf blight, yang mengindikasikan adanya kemiripan pola visual pada area daun tertentu. Sementara itu, pada Gambar (f), citra kelas Leaf smut diprediksi sebagai Brown spot, di mana perhatian model lebih dominan pada area latar belakang, sehingga fitur daun yang relevan kurang menjadi fokus model.

B. 250 Layer Trainable

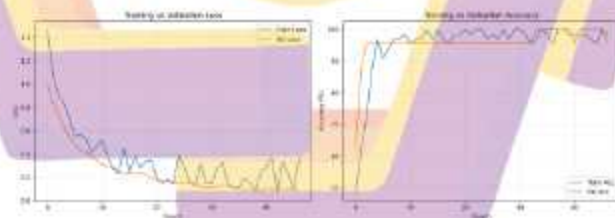
Pada tahap ini dilakukan evaluasi Skenario 2 dengan menerapkan fine-tuning pada 250 layer awal model InceptionV3 menggunakan optimizer SGD, yang dilakukan sebanyak 10 percobaan. sementara layer lainnya tetap bersifat non-trainable. Tabel 4.11 memperlihatkan akurasi data testing dari masing-masing percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.11 Akurasi Testing 10 Percobaan Skenario 2 layer 250 (Optimizer SGD)

Percobaan	Akurasi	F1-Score
Percobaan 1	70	68.29
Percobaan 2	70	64.37
Percobaan 3	80	79.37

Percobaan	Akurasi	F1-Score
Percobaan 4	90	89.93
Percobaan 5	83,33	83.47
Percobaan 6	80	79.23
Percobaan 7	66,66	65.00
Percobaan 8	80	79.37
Percobaan 9	66,66	75.83
Percobaan 10	70	68.15

Dari 10 percobaan yang dilakukan pada Skenario 2, diperoleh variasi nilai akurasi pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan dengan nilai akurasi tertinggi sebagai representasi performa maksimum model. Ditampilkan grafik training dan validation accuracy serta training dan validation loss pada pada Gambar 4.26.



Gambar 4.26 Grafik Layer 250 Training Val Ace dan Loss Skenario 2 (SGD)

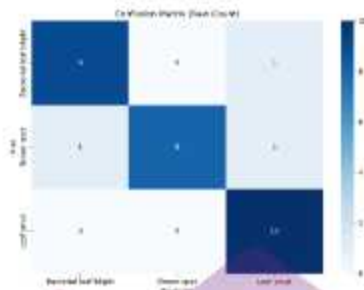
Grafik menunjukkan bahwa nilai loss training dan validasi menurun tajam pada fase awal pelatihan. Loss training turun dari sekitar 1,4 dan mencapai nilai di bawah 0,3 sebelum epoch 15. Loss validasi mengikuti pola serupa dan mulai stabil

di kisaran 0,1–0,15 setelah epoch 20. Pola ini menunjukkan proses pembelajaran berlangsung efektif dan cepat mencapai kondisi konvergen.

Pada grafik akurasi, akurasi validasi meningkat sangat cepat dan telah mencapai sekitar 95% pada epoch awal, kemudian stabil hingga mendekati 100% pada epoch-epoch berikutnya. Akurasi training juga meningkat konsisten dan berada pada kisaran 95–100% hampir sepanjang pelatihan. Jarak antara akurasi training dan validasi relatif kecil, menandakan generalisasi model yang baik meskipun jumlah layer trainable cukup besar.

Meskipun jumlah maksimum epoch ditetapkan sebanyak 100, mekanisme early stopping menghentikan proses pelatihan lebih awal, yaitu sekitar epoch 44–46. Early stopping terjadi karena performa validasi sudah stabil dan tidak menunjukkan peningkatan yang signifikan dalam beberapa epoch berturut-turut. Hal ini mengindikasikan bahwa pelatihan lanjutan tidak diperlukan dan justru berpotensi meningkatkan risiko overfitting.

Untuk melihat pola kesalahan klasifikasi pada masing-masing kelas penyakit daun padi secara lebih rinci, hasil evaluasi model selanjutnya ditunjukkan pada confusion matrix pada Gambar 4.27.



Gambar 4.27 Confusion matrix 250 Layer Skenario 2 (SGD)

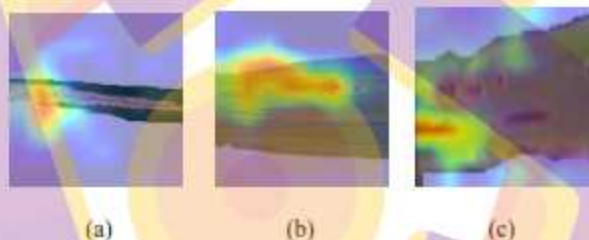
Confusion matrix menunjukkan bahwa model mampu mengklasifikasikan citra daun padi dengan baik pada ketiga kelas penyakit, meskipun masih terdapat beberapa kesalahan kecil. Pada kelas Bacterial leaf blight, dari 10 data uji terdapat 9 data yang diprediksi dengan benar dan 1 data salah diprediksi sebagai Leaf smut. Pada kelas Brown spot, sebanyak 8 data berhasil diklasifikasikan dengan benar, sementara 2 data mengalami kesalahan, yaitu masing-masing 1 data diprediksi sebagai Bacterial leaf blight dan 1 data sebagai Leaf smut. Sementara itu, pada kelas Leaf smut, seluruh 10 data uji berhasil diprediksi dengan benar tanpa kesalahan. Secara keseluruhan, kesalahan klasifikasi yang terjadi relatif kecil dan tersebar pada kelas yang memiliki kemiripan karakteristik visual, sehingga menunjukkan bahwa model telah mempelajari fitur utama masing-masing penyakit dengan cukup baik. Perhitungan akurasi dilakukan menggunakan rumus akurasi multi-kelas:

$$Akurasi = \frac{Prediksi\ Benar}{Total\ Sampel}$$

Dengan nilai prediksi benar (TP) = 27 dan total sampel uji = 30, diperoleh:

$$Akurasi = \frac{27}{30} = 0,90$$

Hasil ini menunjukkan bahwa konfigurasi fine-tuning dengan optimizer SGD mampu memberikan performa klasifikasi yang baik dengan tingkat akurasi 90%. Berdasarkan hasil evaluasi, pada data uji terdapat sampel yang mengalami kesalahan klasifikasi pada kelas Bacterial leaf blight dan Brown spot. Visualisasi Grid-CAM digunakan untuk menganalisis area atensi model pada prediksi yang tidak sesuai dengan label sebenarnya.



Gambar 4.28 Grid-CAM misclassification: (a) Bacterial leaf blight, (b) Brown spot, (c) Brown spot.

Gambar 4.28 Grid-CAM pada sampel data uji yang mengalami kesalahan klasifikasi. Pada Gambar (a), citra dengan label sebenarnya Bacterial leaf blight diprediksi sebagai Leaf smut karena area atensi model lebih terfokus pada bagian daun yang memanjang dan homogen, bukan pada area lesi yang menjadi ciri khas penyakit tersebut. Pada Gambar (b), citra Brown spot diprediksi sebagai Bacterial leaf blight, di mana aktivasi Grid-CAM terkonsentrasi pada pola linear pada permukaan daun, sehingga bercak lokal yang seharusnya menjadi karakteristik utama Brown spot. Sementara itu, pada Gambar (c), Area perhatian model didominasi oleh bercak coklat kehitaman dengan pola memanjang yang memiliki

kemiripan visual dengan karakteristik Leaf Smut, sehingga citra Brown Spot diprediksi sebagai Leaf Smut. Untuk memperjelas kemiripan tersebut, ditampilkan citra daun dengan true label Leaf Smut sebagai pembanding pada Gambar 4.29.



Gambar 4.29 Citra daun padi dengan true label Leaf Smut.

4.3.3 Evaluasi Skenario 3 – Fine-Tuning Layer Akhir

Pada skenario ini dilakukan evaluasi dengan menerapkan fine-tuning pada layer akhir model InceptionV3. Berbeda dengan skenario sebelumnya yang memfokuskan pelatihan pada layer awal, pada skenario ini hanya sebagian layer akhir yang dijadikan trainable, sementara layer lainnya tetap bersifat non-trainable. Pendekatan ini bertujuan untuk menyesuaikan representasi fitur tingkat tinggi agar lebih relevan dengan karakteristik citra penyakit daun padi.

Konfigurasi yang diuji pada skenario ini dibatasi pada dua variasi, yaitu fine-tuning pada 150 layer akhir dan 250 layer akhir. Pemilihan dua konfigurasi ini dimaksudkan untuk melihat pengaruh peningkatan jumlah layer akhir yang dilatih terhadap performa model, tanpa menambah kompleksitas pelatihan secara berlebihan.

Proses pelatihan dijalankan dengan batas maksimum 100 epoch dan dilengkapi mekanisme early stopping. Early stopping digunakan untuk

menghentikan pelatihan ketika performa validasi tidak lagi menunjukkan peningkatan yang signifikan, sehingga risiko overfitting pada dataset berukuran kecil dapat diminimalkan. Hasil evaluasi untuk masing-masing konfigurasi layer akhir tersebut selanjutnya dianalisis melalui grafik loss dan akurasi pada bagian berikutnya. epoch.

4.3.2.1 Evaluasi Skenario 3 dengan Optimizer Adam

Pada bagian ini dibahas evaluasi Skenario 3 menggunakan optimizer Adam, dengan penerapan fine-tuning pada beberapa layer awal model InceptionV3. Evaluasi dilakukan melalui variasi jumlah layer awal yang bersifat trainable, yaitu 150 layer, dan 250 layer, sementara layer lainnya tetap non-trainable. Setiap variasi dievaluasi menggunakan grafik training dan validation accuracy serta training dan validation loss, untuk melihat pengaruh jumlah layer awal yang dilatih terhadap proses pelatihan dan performa model.

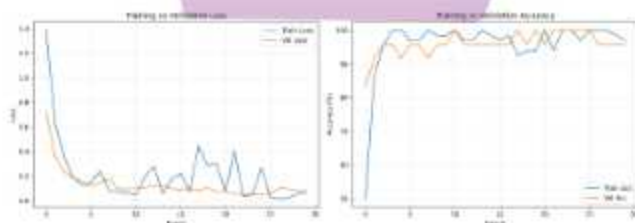
A. 150 Layer Trainable

Pada tahap ini dilakukan evaluasi Skenario 3 dengan menerapkan fine-tuning pada 150 layer akhir model InceptionV3 menggunakan optimizer Adam, yang dilakukan sebanyak 10 percobaan. sementara layer lainnya tetap bersifat non-trainable. Tabel 4.12 memperlihatkan akurasi data testing dari masing-masing percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.12 Akurasi Testing 10 Percobaan Skenario 3 layer 150 (Optimizer Adam)

Percobaan	Akurasi	F1-Score
Percobaan 1	86,66	86,33
Percobaan 2	93,33	93,27
Percobaan 3	83,33	82,56
Percobaan 4	90	89,50
Percobaan 5	93,33	93,46
Percobaan 6	80	78,89
Percobaan 7	90	89,82
Percobaan 8	83,33	82,58
Percobaan 9	90	89,50
Percobaan 10	90	90,15

Dari 10 percobaan yang dilakukan pada Skenario 3, diperoleh variasi nilai akurasi pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan dengan nilai akurasi tertinggi sebagai representasi performa maksimum model. Ditampilkan grafik training dan validation accuracy serta training dan validation loss pada pada Gambar 4.30.

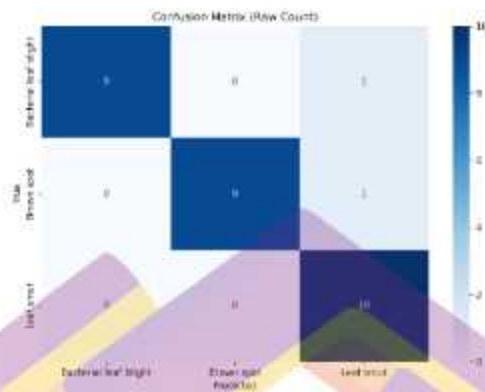


Gambar 4.30 Grafik Layer 150 Training Val Acc dan Loss Skenario 3 (Adam)

Grafik loss menunjukkan penurunan yang sangat cepat pada fase awal pelatihan. Loss training turun tajam dari kisaran 1,3 pada epoch awal dan sudah berada di bawah 0,2 sebelum epoch 5. Loss validasi mengikuti pola yang sama dan mulai stabil pada kisaran 0,1 sejak sekitar epoch 6–8. Kondisi ini menunjukkan bahwa fine-tuning pada layer akhir mampu mempercepat proses konvergensi karena parameter yang dilatih lebih sedikit dan lebih spesifik terhadap tugas klasifikasi.

Pada grafik akurasi terlihat bahwa akurasi training meningkat sangat cepat dan mencapai nilai mendekati 100% sejak sekitar epoch 3–4. Akurasi validasi juga meningkat secara signifikan dan berada pada kisaran 95–100% sejak epoch 4. Selisih antara akurasi training dan validasi relatif kecil dan stabil, yang menandakan model memiliki kemampuan generalisasi yang baik meskipun hanya sebagian layer akhir yang dilatih.

Meskipun jumlah maksimum epoch ditetapkan sebesar 100, mekanisme early stopping menghentikan proses pelatihan lebih awal, yaitu pada sekitar epoch 28–30. Early stopping terjadi karena nilai loss dan akurasi validasi tidak lagi menunjukkan peningkatan yang signifikan dalam beberapa epoch berturut-turut. Hal ini mengindikasikan bahwa model telah mencapai kondisi optimal, dan pelatihan lanjutan berpotensi menimbulkan overfitting tanpa peningkatan performa yang berarti. Untuk melihat dampak konfigurasi ini terhadap hasil klasifikasi pada data testing, evaluasi selanjutnya dilakukan menggunakan confusion matrix pada Gambar 4.31.



Gambar 4.31 Confusion matrix 150 Layer Skenario 3 (Adam)

Confusion matrix menunjukkan hasil klasifikasi pada data uji untuk tiga kelas penyakit daun padi. Kelas Bacterial leaf blight seluruhnya terklasifikasi dengan benar sebanyak 10 data tanpa kesalahan. Kelas Brown spot memiliki 8 data yang terklasifikasi benar, sedangkan 2 data mengalami kesalahan prediksi, masing-masing satu diprediksi sebagai Bacterial leaf blight dan satu sebagai Leaf smut. Kelas Leaf smut juga terklasifikasi sempurna dengan 10 data benar dan tanpa kesalahan. Pola ini menunjukkan bahwa kesalahan prediksi hanya terjadi pada kelas Brown spot, sedangkan dua kelas lainnya dapat dikenali dengan sangat baik oleh model.

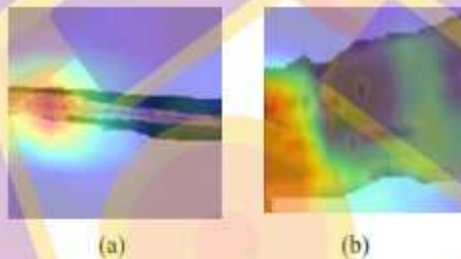
Perhitungan akurasi dilakukan menggunakan rumus akurasi multi-kelas:

$$\text{Akurasi} = \frac{\text{Prediksi Benar}}{\text{Total Sampel}}$$

Dengan nilai prediksi benar (TP) = 28 dan total sampel uji = 30, diperoleh:

$$Akurasi = \frac{28}{30} = 0,9333$$

Berdasarkan hasil evaluasi, pada data uji terdapat sampel yang mengalami kesalahan klasifikasi pada kelas Bacterial leaf blight dan Brown spot. Visualisasi Grid-CAM digunakan untuk menganalisis area atensi model pada prediksi yang tidak sesuai dengan label sebenarnya.



Gambar 4.32 Grid-CAM kesalahan klasifikasi: (a) Bacterial leaf blight, (b) Brown spot

Gambar menampilkan visualisasi Grid-CAM pada sampel data uji yang mengalami kesalahan klasifikasi. Pada Gambar (a), citra dengan label sebenarnya Bacterial leaf blight diprediksi sebagai Leaf smut karena area atensi model lebih terfokus pada bagian daun yang memanjang dan homogen, bukan pada area lesi yang menjadi ciri khas penyakit tersebut. Pada Gambar (b), citra Brown spot diprediksi sebagai Bacterial leaf blight, di mana aktivasi Grid-CAM terkonsentrasi pada pola linear pada permukaan daun, sehingga bercak lokal yang seharusnya menjadi karakteristik utama Brown spot kurang mendapat perhatian.

B. 250 Layer Trainable

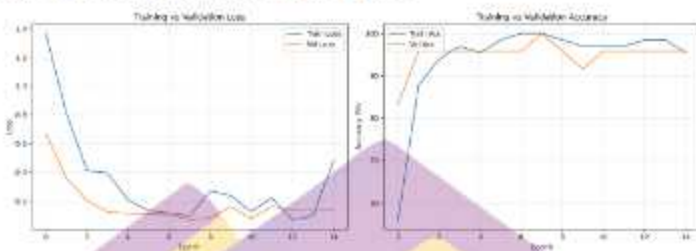
Pada tahap ini dilakukan evaluasi Skenario 3 dengan menerapkan fine-tuning pada 250 layer akhir model InceptionV3 menggunakan optimizer Adam, yang dilakukan sebanyak 10 percobaan. sementara layer lainnya tetap bersifat non-trainable. Tabel 4.13 memperlihatkan akurasi data testing dari masing-masing percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.13 Akurasi Testing 10 Percobaan Skenario 3 layer 250 (Optimizer Adam)

Percobaan	Akurasi	F1-Score
Percobaan 1	96,66	96,66
Percobaan 2	96,66	96,66
Percobaan 3	96,66	96,66
Percobaan 4	96,66	96,66
Percobaan 5	96,66	96,66
Percobaan 6	96,66	96,66
Percobaan 7	96,66	96,66
Percobaan 8	96,66	96,66
Percobaan 9	96,66	96,66
Percobaan 10	96,66	96,66

Dari 10 percobaan yang dilakukan pada Skenario 3, seluruh percobaan menghasilkan nilai akurasi yang sama pada data testing yaitu 96,66. Tidak ditemukan variasi akurasi antar percobaan, sehingga menunjukkan performa model

yang konsisten dan stabil. Ditampilkan grafik training dan validation accuracy serta training dan validation loss pada Gambar 4.33.



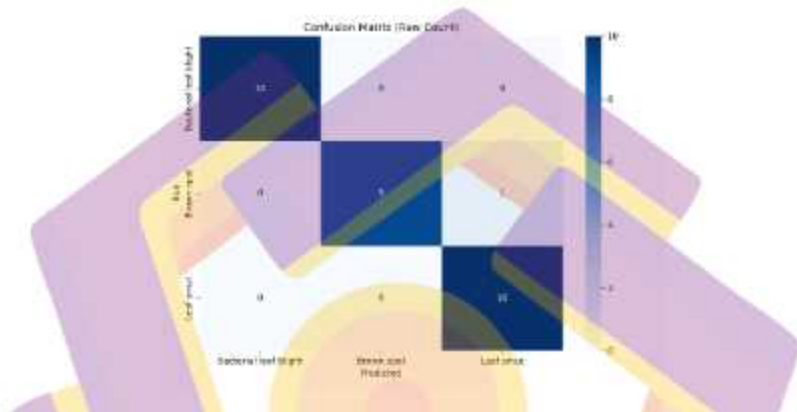
Gambar 4.33 Grafik Layer 250 Training Val Acc dan Loss Skenario 3 (Adam)

Berdasarkan grafik pada Gambar 4.33, proses pelatihan menunjukkan konvergensi yang cepat. Nilai loss training dan loss validasi menurun tajam pada epoch awal dan mulai stabil sekitar epoch 5 hingga epoch 7. Selisih antara loss training dan loss validasi relatif kecil, yang menunjukkan proses pembelajaran berjalan stabil dan tidak menunjukkan indikasi overfitting yang signifikan.

Akurasi training meningkat dengan cepat sejak epoch awal dan mencapai nilai tinggi pada epoch ke 3 hingga ke 5. Akurasi validasi mengikuti pola yang serupa dan stabil di kisaran tinggi hingga akhir pelatihan, meskipun terdapat fluktuasi kecil pada beberapa epoch. Kondisi ini menunjukkan bahwa model mampu melakukan generalisasi dengan baik pada data validasi.

Proses pelatihan dihentikan pada epoch ke 15 oleh mekanisme early stopping. Penghentian ini dilakukan karena setelah beberapa epoch terakhir tidak terlihat peningkatan yang signifikan pada nilai loss validasi maupun akurasi validasi. Nilai loss cenderung stabil dan akurasi validasi berada pada kisaran tinggi

yang relatif konstan. Kondisi ini menunjukkan bahwa model telah mencapai performa optimal, sehingga melanjutkan pelatihan berpotensi tidak memberikan peningkatan kinerja dan justru meningkatkan risiko overfitting, hasil evaluasi confusion matrix ditunjukkan pada Gambar 4.34.



Gambar 4.34 Confusion matrix 250 Layer Skenario 3 (Adam)

Berdasarkan confusion matrix yang ditampilkan, model menunjukkan kinerja klasifikasi yang sangat baik pada data uji. Untuk kelas Bacterial leaf blight, seluruh data uji sebanyak 10 citra berhasil diklasifikasikan dengan benar tanpa kesalahan. Hal ini menunjukkan bahwa model mampu mengenali ciri khas kelas tersebut secara konsisten. Pada kelas Brown spot, dari 10 data uji, sebanyak 9 citra berhasil diklasifikasikan dengan benar, sedangkan 1 citra salah diklasifikasikan sebagai Leaf smut. Kesalahan ini menunjukkan adanya kemiripan pola visual antara kedua kelas tersebut pada beberapa kasus tertentu.

Untuk kelas Leaf smut, seluruh data uji sebanyak 10 citra berhasil diklasifikasikan dengan benar tanpa terjadi kesalahan prediksi. Hal ini menandakan

bahwa model mampu membedakan kelas Leaf smut dengan sangat baik dibandingkan kelas lainnya. Secara keseluruhan, hanya terdapat 1 kesalahan klasifikasi dari total 30 data uji. Hasil ini mendukung nilai akurasi yang tinggi dan menunjukkan bahwa model memiliki kemampuan generalisasi yang baik meskipun dataset yang digunakan terbatas.

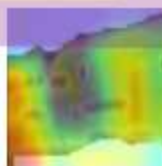
Perhitungan akurasi dilakukan menggunakan rumus akurasi multi-kelas:

$$Akurasi = \frac{Prediksi\ Benar}{Total\ Sampel}$$

Dengan nilai prediksi benar (TP) = 29 dan total sampel uji = 30, diperoleh:

$$Akurasi = \frac{29}{30} = 0,96$$

Hasil ini menunjukkan bahwa fine-tuning pada layer akhir mampu menghasilkan performa klasifikasi yang sangat baik dengan akurasi sebesar 96,66%. Berdasarkan hasil evaluasi, pada data uji terdapat sampel yang mengalami kesalahan klasifikasi pada kelas Brown spot. Visualisasi Grid-CAM digunakan untuk menganalisis area atensi model pada prediksi yang tidak sesuai dengan label sebenarnya.



Gambar 4.35 Grid-CAM misclassification Brown spot.

Gambar 4.35 Grid-CAM pada citra dengan label sebenarnya Brown spot yang diprediksi sebagai Leaf smut. Area atensi model terlihat menyebar dan tidak terlokalisasi secara jelas pada bercak penyakit utama, sehingga ciri khas Brown spot kurang menjadi fokus model dan menyebabkan terjadinya kesalahan klasifikasi.

Hasil menjadi lebih baik ketika 250 layer terakhir dibuka karena layer tersebut berada paling dekat dengan bagian output, sehingga fitur yang dipelajari bersifat lebih spesifik terhadap tugas klasifikasi pada penelitian ini. Dengan membuka 250 layer terakhir, model tetap mempertahankan fitur umum dari layer awal, seperti pola dasar dan tekstur, sekaligus mampu menyesuaikan fitur tingkat tinggi agar lebih relevan dengan karakteristik dataset yang digunakan.

4.3.2.2 Evaluasi Skenario 3 dengan Optimizer AdamW

Pada bagian ini dilakukan evaluasi Skenario 3 dengan menerapkan fine tuning pada beberapa layer awal model InceptionV3 menggunakan optimizer AdamW, sementara layer lainnya tetap bersifat non-trainable. Penggunaan AdamW bertujuan untuk menguji pengaruh mekanisme weight decay yang lebih terkontrol terhadap proses pembelajaran model, khususnya pada kondisi dataset yang berukuran terbatas.

A. 150 Layer Trainable

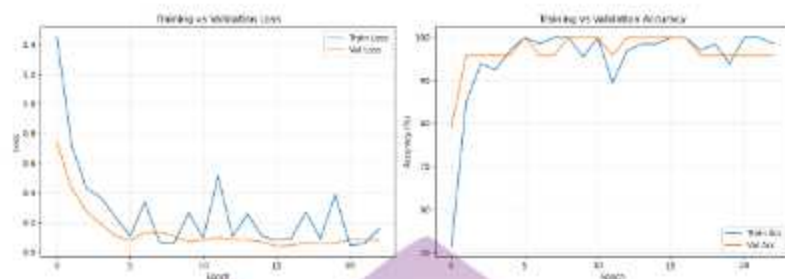
Pada tahap ini dilakukan evaluasi Skenario 3 dengan menerapkan fine-tuning pada 150 layer akhir model InceptionV3 menggunakan optimizer AdamW, yang dilakukan sebanyak 10 percobaan. sementara layer lainnya tetap bersifat non-trainable. Tabel 4.14 memperlihatkan akurasi data testing dari masing-masing

percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.14 Akurasi Testing 10 Percobaan Skenario 3 layer 150 (Optimizer AdamW)

Percobaan	Akurasi	F1-Score
Percobaan 1	96,66	96,66
Percobaan 2	93,33	93,32
Percobaan 3	93,33	93,12
Percobaan 4	80	79,37
Percobaan 5	93,33	93,12
Percobaan 6	86,66	86,44
Percobaan 7	96,66	96,66
Percobaan 8	86,66	86,24
Percobaan 9	90	89,82
Percobaan 10	93,33	93,46

Dari 10 percobaan yang dilakukan pada Skenario 3, diperoleh variasi nilai akurasi pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan dengan nilai akurasi tertinggi sebagai representasi performa maksimum model. Ditampilkan grafik training dan validation accuracy serta training dan validation loss pada Gambar 4.36.



Gambar 4.36 Grafik Layer 150 Training Val Acc, Loss Skenario 3 (AdamW)

Grafik menunjukkan bahwa proses pelatihan pada skenario fine-tuning layer akhir berjalan cepat dan stabil. Nilai loss training turun tajam dari sekitar 1,4 pada epoch awal dan mencapai nilai rendah di bawah 0,2 sebelum epoch 5. Loss validasi mengikuti pola yang sama dan cenderung stabil di kisaran 0,05–0,1 setelah epoch 5. Kondisi ini menunjukkan bahwa model dengan fine-tuning pada layer akhir mampu mencapai konvergensi lebih cepat dibanding skenario layer awal.

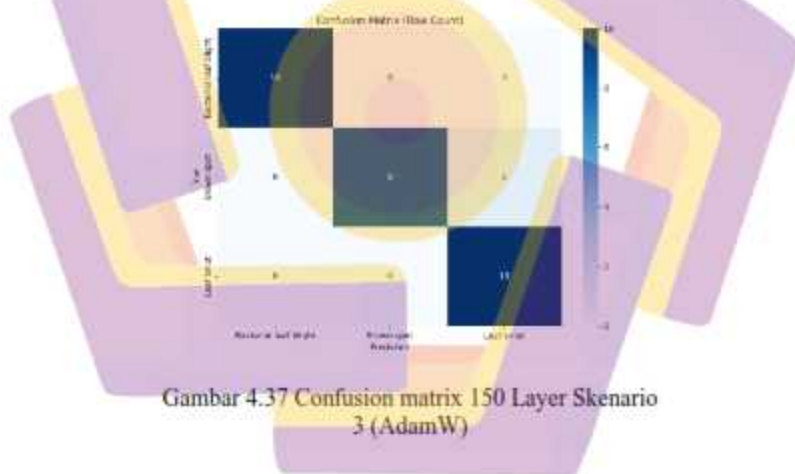
Pada grafik akurasi, akurasi training meningkat signifikan sejak epoch awal dan mencapai 100% pada sekitar epoch 5. Akurasi validasi juga meningkat cepat dan berada pada kisaran 95–100% sejak epoch 3 hingga akhir pelatihan. Perbedaan antara akurasi training dan validasi relatif kecil, menandakan bahwa model tidak mengalami overfitting yang signifikan meskipun hanya sebagian layer akhir yang bersifat trainable.

Meskipun jumlah maksimum epoch ditetapkan sebanyak 100, mekanisme early stopping menghentikan proses pelatihan lebih awal, yaitu pada sekitar epoch 22. Early stopping terjadi karena performa validasi sudah stabil dan tidak

menunjukkan peningkatan yang berarti dalam beberapa epoch berturut-turut. Hal ini menegaskan bahwa pelatihan tambahan setelah titik tersebut tidak lagi memberikan kontribusi signifikan terhadap peningkatan kinerja model.

Untuk melihat pola kesalahan klasifikasi pada masing-masing kelas penyakit daun padi secara lebih rinci, hasil evaluasi model selanjutnya ditunjukkan melalui confusion matrix pada gambar berikutnya.

Untuk melihat distribusi kesalahan klasifikasi pada masing-masing kelas penyakit daun padi secara lebih detail, hasil evaluasi model selanjutnya ditunjukkan pada confusion matrix pada Gambar 4.37.



Gambar 4.37 Confusion matrix 150 Layer Skenario 3 (AdamW)

Confusion matrix menunjukkan hasil klasifikasi untuk tiga kelas penyakit daun padi. Pada kelas Bacterial leaf blight, seluruh 10 data uji berhasil diprediksi dengan benar tanpa kesalahan. Pada kelas Brown spot, sebanyak 9 data diprediksi dengan benar, sedangkan 1 data salah diklasifikasikan sebagai Leaf smut. Pada

kelas Leaf smut, seluruh 10 data uji juga berhasil diklasifikasikan dengan benar. Kesalahan klasifikasi hanya terjadi pada kelas Brown spot dan jumlahnya sangat terbatas.

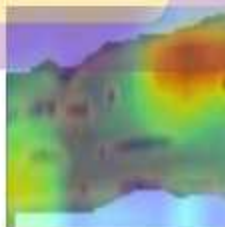
Perhitungan akurasi dilakukan menggunakan rumus akurasi multi-kelas:

$$\text{Akurasi} = \frac{\text{Prediksi Benar}}{\text{Total Sampel}}$$

Dengan nilai prediksi benar (TP) = 27 dan total sampel uji = 30, diperoleh:

$$\text{Akurasi} = \frac{27}{30} = 0,96$$

Hasil ini menunjukkan bahwa fine-tuning pada layer akhir mampu menghasilkan performa klasifikasi yang sangat baik dengan akurasi 96,66%. Model mampu mengenali kelas Bacterial leaf blight dan Leaf smut secara sempurna, sedangkan kesalahan hanya muncul satu kali pada kelas Brown spot. Berdasarkan hasil evaluasi, pada data uji terdapat sampel yang mengalami kesalahan klasifikasi pada kelas Brown spot. Visualisasi Grid-CAM digunakan untuk menganalisis area atensi model pada prediksi yang tidak sesuai dengan label sebenarnya.



Gambar 4.38 Grid-CAM misclassification Brown spot.

Gambar 4.38 Grid-CAM pada citra dengan label sebenarnya Brown spot yang diprediksi sebagai Leaf smut. Area atensi model terlihat menyebar dan tidak terlokalisasi secara spesifik pada bercak penyakit khas Brown spot, melainkan lebih terfokus pada area daun yang relatif homogen. Pola atensi yang tidak spesifik ini menunjukkan bahwa model kurang menekankan fitur diskriminatif utama Brown spot, sehingga menghasilkan prediksi yang keliru.

B. 250 Layer Trainable

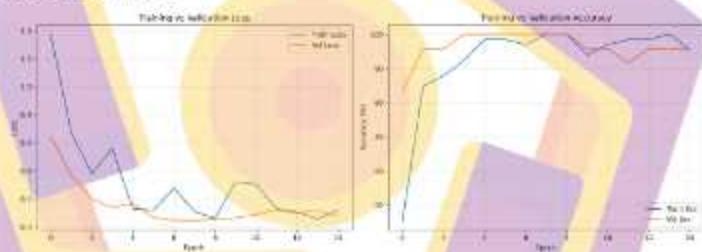
Pada tahap ini dilakukan evaluasi Skenario 3 dengan menerapkan fine-tuning pada 250 layer akhir model InceptionV3 menggunakan optimizer AdamW, yang dilakukan sebanyak 10 percobaan. sementara layer lainnya tetap bersifat non-trainable. Tabel 4.15 memperlihatkan akurasi data testing dari masing-masing percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.15 Akurasi Testing 10 Percobaan Skenario 3 layer 250 (Optimizer AdamW)

Percobaan	Akurasi	F1-Score
Percobaan 1	90	89.50
Percobaan 2	90	89.50
Percobaan 3	86,66	85.61
Percobaan 4	86,66	86.44
Percobaan 5	90	89.50
Percobaan 6	90	83.07
Percobaan 7	96,66	96.66

Percobaan	Akurasi	F1-Score
Percobaan 8	90	89.93
Percobaan 9	90	79.80
Percobaan 10	90	89.93

Dari 10 percobaan yang dilakukan pada Skenario 3, diperoleh variasi nilai akurasi pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan dengan nilai akurasi tertinggi sebagai representasi performa maksimum model. Ditampilkan grafik training dan validation accuracy serta training dan validation loss pada Gambar 4.39.



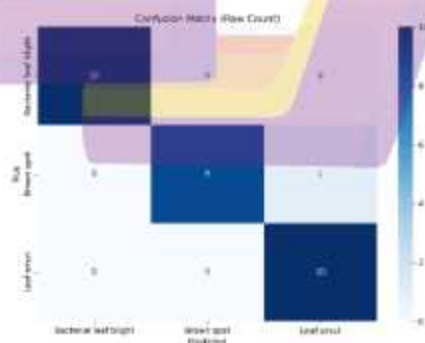
Gambar 4.39 Grafik Layer 250 Training Val Acc dan Loss Skenario 3 (AdamW)

Grafik pada Gambar 4.39, menunjukkan bahwa nilai loss training dan validasi turun sangat cepat pada awal proses pelatihan. Loss training awal berada di kisaran 1,4 dan menurun tajam hingga berada di bawah 0,2 pada sekitar epoch 4–5. Loss validasi juga menunjukkan pola serupa, turun dari sekitar 0,6 dan mulai stabil di bawah 0,1 setelah epoch 5. Pola ini menandakan bahwa model dengan fine-tuning layer akhir mampu mempelajari fitur penting secara cepat dan efisien.

Pada grafik akurasi, akurasi training meningkat drastis dari sekitar 45% pada epoch awal dan mencapai lebih dari 95% pada epoch 3–4. Akurasi validasi juga meningkat cepat dan mencapai 95–100% sejak epoch ke-3. Setelah titik tersebut, baik akurasi training maupun validasi cenderung stabil dengan fluktuasi kecil, menunjukkan bahwa model telah mencapai kondisi konvergen lebih awal.

Meskipun jumlah maksimum epoch ditetapkan sebesar 100, mekanisme early stopping menghentikan proses pelatihan lebih awal, yaitu pada sekitar epoch 14. Early stopping terjadi karena performa validasi, khususnya loss dan akurasi, tidak lagi menunjukkan peningkatan yang berarti dalam beberapa epoch berturut-turut. Hal ini menunjukkan bahwa pelatihan lanjutan setelah titik tersebut tidak memberikan peningkatan performa yang signifikan dan berpotensi meningkatkan risiko overfitting pada dataset berukuran kecil.

Untuk melihat pola kesalahan klasifikasi pada masing-masing kelas penyakit daun padi secara lebih rinci, hasil evaluasi model selanjutnya ditunjukkan pada confusion matrix pada Gambar 4.40.



Gambar 4.40 Confusion matrix 250 Layer Skenario 3 (AdamW)

Confusion matrix menunjukkan performa klasifikasi model pada tiga kelas penyakit daun padi. Kelas Bacterial leaf blight seluruh data uji sebanyak 10 citra berhasil diklasifikasikan dengan benar tanpa kesalahan. Kelas Brown spot memiliki 9 citra yang terklasifikasi dengan benar dan 1 citra salah diklasifikasikan sebagai Leaf smut. Kelas Leaf smut seluruh 10 citra berhasil dikenali dengan benar tanpa kesalahan klasifikasi. Kesalahan hanya terjadi pada kelas Brown spot dan bersifat terbatas. mirip.

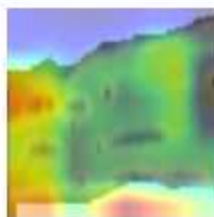
Perhitungan akurasi dilakukan menggunakan rumus akurasi multi-kelas:

$$Akurasi = \frac{Prediksi\ Benar}{Total\ Sampel}$$

Dengan nilai prediksi benar (TP) = 28 dan total sampel uji = 30, diperoleh:

$$Akurasi = \frac{29}{30} = 0,96$$

Hasil ini menunjukkan bahwa model mampu melakukan klasifikasi dengan sangat baik dengan tingkat akurasi sebesar 96,66%. Berdasarkan hasil evaluasi, pada data uji terdapat sampel yang mengalami kesalahan klasifikasi pada kelas Brown spot. Visualisasi Grid-CAM digunakan untuk menganalisis area atensi model pada prediksi yang tidak sesuai dengan label sebenarnya.



Gambar 4.41 Grid-CAM misclassification Brown spot.

Gambar 4.41 Grid-CAM pada citra dengan label sebenarnya Brown spot yang diprediksi sebagai Leaf smut. Area atensi model tampak menyebar dan tidak terfokus pada bercak penyakit utama, melainkan pada area daun yang relatif seragam. Kondisi ini mengindikasikan bahwa fitur visual khas Brown spot belum sepenuhnya menjadi perhatian utama model, sehingga menyebabkan terjadinya kesalahan klasifikasi.

4.3.2.3 Evaluasi Skenario 3 dengan Optimizer SGD

Pada tahap ini dilakukan evaluasi Skenario 3 dengan menerapkan fine-tuning pada model InceptionV3 menggunakan optimizer SGD, dengan variasi jumlah layer awal yang dijadikan trainable. Konfigurasi yang diuji meliputi fine-tuning pada 150 layer awal, hingga 250 layer awal, sementara layer lainnya tetap bersifat non-trainable. Variasi ini bertujuan untuk mengamati pengaruh kedalaman fine-tuning terhadap kemampuan model dalam mempelajari pola penyakit daun padi.

Proses pelatihan dijalankan dengan batas maksimum 100 epoch dan dilengkapi mekanisme early stopping. Early stopping diterapkan karena pada dataset berukuran kecil, pelatihan yang terlalu panjang sering kali tidak lagi meningkatkan performa validasi dan berpotensi menyebabkan overfitting. Dengan mekanisme ini, proses pelatihan dihentikan ketika tidak terjadi peningkatan kinerja validasi dalam beberapa epoch berturut-turut. peningkatan yang signifikan. Evaluasi pada skenario ini difokuskan pada pengamatan perubahan nilai loss dan akurasi pada data training dan validasi, serta dampaknya terhadap performa klasifikasi pada data uji.

A. 150 Layer Trainable

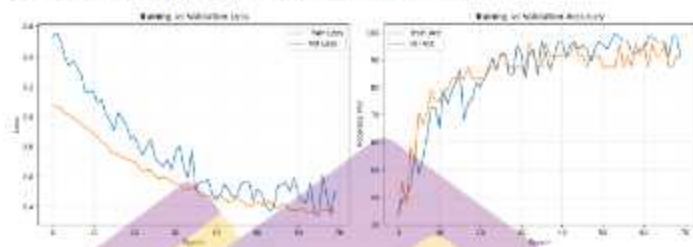
Pada tahap ini dilakukan evaluasi Skenario 3 dengan menerapkan fine-tuning pada 150 layer akhir model InceptionV3 menggunakan optimizer SGD, yang dilakukan sebanyak 10 percobaan. sementara layer lainnya tetap bersifat non-trainable. Tabel 4.16 memperlihatkan akurasi data testing dari masing-masing percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.16 Akurasi Testing-10 Percobaan Skenario 3 layer-150 (Optimizer SGD)

Percobaan	Akurasi	F1-Score
Percobaan 1	80	79.37
Percobaan 2	73,33	72.28
Percobaan 3	83,33	83.07
Percobaan 4	70	69.90
Percobaan 5	76,66	76.19
Percobaan 6	80	79.96
Percobaan 7	76,66	74.10
Percobaan 8	76,66	75.26
Percobaan 9	73,33	72.28
Percobaan 10	76,66	74.10

Dari 10 percobaan yang dilakukan pada Skenario 3, diperoleh variasi nilai akurasi pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan dengan nilai akurasi tertinggi sebagai representasi performa maksimum model. Ditampilkan grafik training dan validation accuracy serta training dan

validation loss pada Perubahan loss dan akurasi selama training dan validasi untuk konfigurasi ini ditunjukkan pada grafik pada Gambar 4.42.

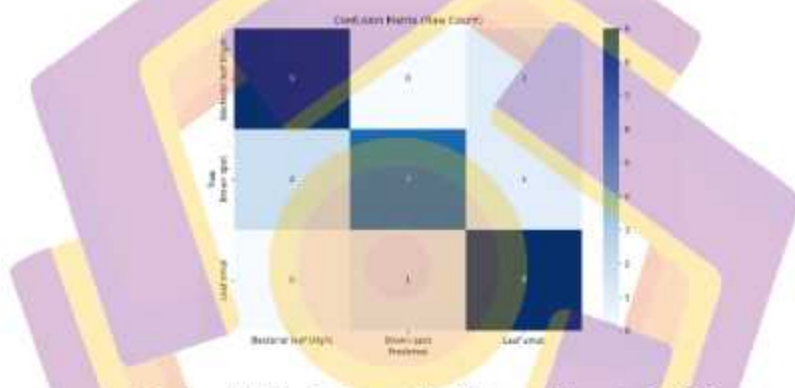


Gambar 4.42 Grafik Layer 150 Training Val Acc dan Loss Skenario 3 (SGD)

Grafik pada Gambar 4.42, menunjukkan bahwa nilai loss training dan loss validasi mengalami penurunan bertahap selama proses pelatihan. Loss training turun dari kisaran 1,5 pada awal epoch dan mulai stabil di bawah 0,6 setelah sekitar epoch 40. Loss validasi mengikuti pola yang serupa dan menunjukkan kestabilan di kisaran 0,4 sejak sekitar epoch 45. Pola ini menandakan bahwa proses pembelajaran berlangsung konsisten dan model mampu menyesuaikan bobot tanpa lonjakan loss yang signifikan.

Pada grafik akurasi, akurasi training dan validasi meningkat secara progresif pada fase awal pelatihan. Akurasi validasi mencapai kisaran 80% pada sekitar epoch 8–10 dan terus meningkat hingga berada pada rentang 90–95% setelah epoch 25. Akurasi training menunjukkan tren yang sejalan dan mencapai kestabilan di kisaran 95–100% mulai sekitar epoch 30. Perbedaan antara akurasi training dan validasi relatif kecil, yang menunjukkan bahwa model tidak mengalami overfitting yang berat dan masih memiliki kemampuan generalisasi yang baik.

Meskipun jumlah maksimum epoch ditetapkan sebanyak 100, proses pelatihan dihentikan lebih awal oleh mekanisme early stopping pada sekitar epoch 68–70. Early stopping terjadi karena performa validasi tidak lagi menunjukkan peningkatan yang signifikan dalam beberapa epoch berturut-turut. Untuk melihat distribusi kesalahan klasifikasi pada masing-masing kelas penyakit daun padi secara lebih detail, hasil evaluasi model selanjutnya ditunjukkan pada confusion matrix pada Gambar 4.43.



Gambar 4.43 Confusion matrix 150 Layer Skenario 3 (SGD)

Pada kelas Bacterial leaf blight, sebanyak 9 data berhasil diklasifikasikan dengan benar, sementara 1 data salah diklasifikasikan sebagai Leaf smut. Tidak terdapat kesalahan ke kelas Brown spot. Pada kelas Brown spot, terdapat 7 data yang diklasifikasikan dengan benar. Sebanyak 2 data salah diprediksi sebagai Bacterial leaf blight dan 1 data salah diprediksi sebagai Leaf smut. Pada kelas Leaf smut, sebanyak 9 data berhasil diklasifikasikan dengan benar dan 1 data salah diprediksi sebagai Brown spot. Tidak terdapat kesalahan ke kelas Bacterial leaf blight.

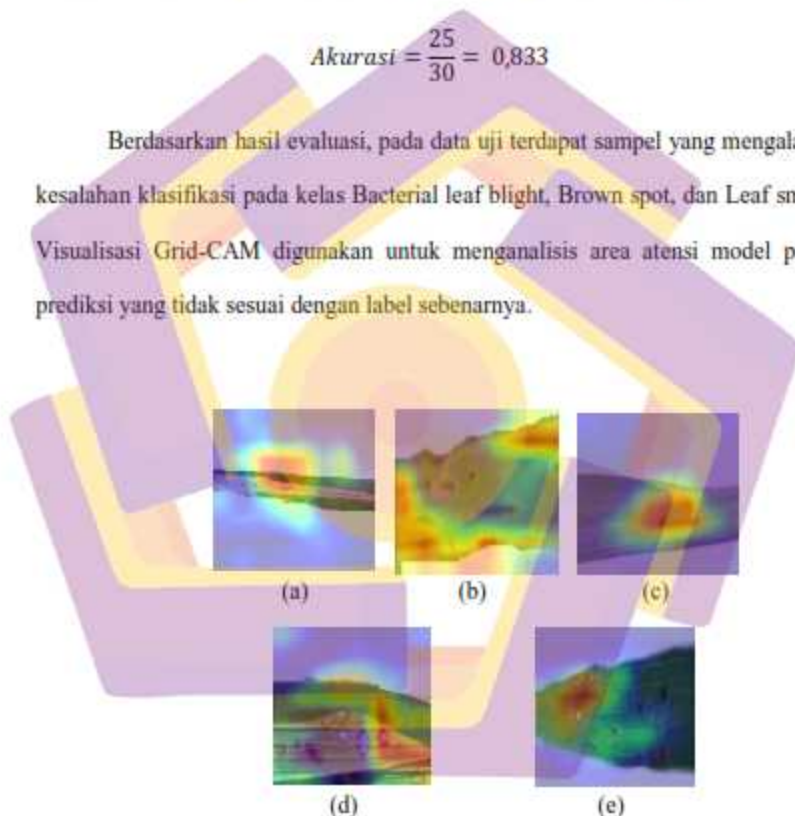
Perhitungan akurasi dilakukan menggunakan rumus akurasi multi-kelas:

$$Akurasi = \frac{Prediksi\ Benar}{Total\ Sampel}$$

Dengan nilai prediksi benar (TP) = 25 dan total sampel uji = 30, diperoleh:

$$Akurasi = \frac{25}{30} = 0,833$$

Berdasarkan hasil evaluasi, pada data uji terdapat sampel yang mengalami kesalahan klasifikasi pada kelas Bacterial leaf blight, Brown spot, dan Leaf smut. Visualisasi Grid-CAM digunakan untuk menganalisis area atensi model pada prediksi yang tidak sesuai dengan label sebenarnya.



Gambar 4.44 Grid-CAM misclassification: (a) Bacterial leaf blight, (b) Brown spot, (c) Brown spot, (d) Brown spot, (e) Leaf smut.

Gambar 4.44 Grid-CAM pada beberapa sampel data uji yang mengalami kesalahan klasifikasi. Pada Gambar (a), citra dengan label sebenarnya Bacterial leaf

blight diprediksi sebagai Leaf smut, di mana area atensi model lebih terfokus pada bagian daun yang memanjang dan relatif homogen, sehingga ciri lesi khas Bacterial leaf blight kurang mendapat perhatian. Pada Gambar (b), citra Brown spot diprediksi sebagai Leaf smut, dengan pola atensi yang menyebar dan tidak terlokalisasi pada bercak penyakit utama. Selanjutnya, pada Gambar (c) dan (d), citra Brown spot diprediksi sebagai Bacterial leaf blight, di mana aktivasi Grid-CAM terkonsentrasi pada area tertentu pada permukaan daun yang memiliki kemiripan pola visual dengan kelas tersebut. Gambar (e), citra Leaf smut diprediksi sebagai Brown spot, dengan perhatian model yang lebih dominan pada area daun secara umum, sehingga fitur spesifik Leaf smut tidak menjadi fokus utama model.

B. 250 Layer Trainable

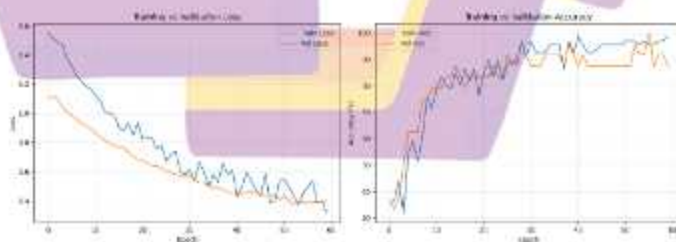
Pada tahap ini dilakukan evaluasi Skenario 3 dengan menerapkan fine-tuning pada 250 layer akhir model InceptionV3 menggunakan optimizer SGD, yang dilakukan sebanyak 10 percobaan. sementara layer lainnya tetap bersifat non-trainable. Tabel 4.17 memperlihatkan akurasi data testing dari masing-masing percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.17 Akurasi Testing 10 Percobaan Skenario 3 layer 250 (Optimizer SGD)

Percobaan	Akurasi	F1-Score
Percobaan 1	76,66	75.45
Percobaan 2	86,66	86.64

Percobaan	Akurasi	F1-Score
Percobaan 3	76,66	74.13
Percobaan 4	83,33	82.78
Percobaan 5	90	89.93
Percobaan 6	80	79.19
Percobaan 7	73,33	69.32
Percobaan 8	86,66	86.44
Percobaan 9	83,33	83.13
Percobaan 10	76,66	75,38

Dari 10 percobaan yang dilakukan pada Skenario 3, diperoleh variasi nilai akurasi pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan dengan nilai akurasi dan F1-Score tertinggi sebagai representasi performa maksimum model. Ditampilkan grafik training dan validation accuracy serta training dan validation loss pada Perubahan loss dan akurasi selama training dan validasi untuk konfigurasi ini ditunjukkan pada Gambar 4.45.



Gambar 4.45 Grafik Layer 250 Training Val Acc dan Loss Skenario 3 (SGD)

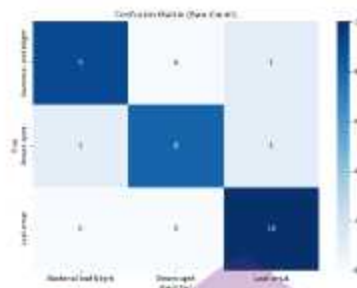
Berdasarkan grafik loss, nilai training loss dan validation loss menurun secara konsisten sejak awal pelatihan. Training loss turun dari kisaran sekitar 1,5

dan mulai stabil di bawah 0,6 setelah sekitar epoch 30–35. Validation loss mengikuti pola yang serupa dan menunjukkan kestabilan setelah epoch 35–40. Kondisi ini menunjukkan bahwa proses pembelajaran berlangsung dengan baik dan model mulai mencapai fase konvergensi.

Pada grafik akurasi, akurasi training dan validasi meningkat tajam pada fase awal pelatihan. Akurasi validasi telah mencapai kisaran 75–80% pada sekitar epoch 8–10, kemudian meningkat secara bertahap hingga berada pada rentang 90–95% setelah epoch 25. Akurasi training menunjukkan tren yang sejalan dan berada sedikit di atas akurasi validasi, dengan selisih yang relatif kecil. Hal ini mengindikasikan kemampuan generalisasi model yang cukup baik meskipun hanya layer akhir yang dilatih.

Meskipun jumlah maksimum epoch ditetapkan sebesar 100, early stopping menghentikan proses pelatihan lebih awal, yaitu pada sekitar epoch 55–60, ketika akurasi validasi tidak lagi menunjukkan peningkatan yang signifikan dalam beberapa epoch berturut-turut. Penghentian ini menunjukkan bahwa pelatihan tambahan tidak memberikan kontribusi berarti terhadap peningkatan performa dan berpotensi meningkatkan risiko overfitting.

Untuk melihat pola kesalahan klasifikasi secara lebih rinci pada masing-masing kelas penyakit daun padi, hasil evaluasi model selanjutnya ditunjukkan pada confusion matrix pada gambar berikut.



Gambar 4.46 Confusion matrix 250 Layer Skenario 3 (SGD)

Pada kelas Bacterial leaf blight, sebanyak 9 data berhasil diklasifikasikan dengan benar. Terdapat 1 data yang salah diklasifikasikan sebagai Leaf smut. Tidak ada data yang keliru diprediksi sebagai Brown spot. Hal ini menunjukkan bahwa model cukup baik mengenali pola Bacterial leaf blight, meskipun masih terdapat sedikit kebingungan dengan Leaf smut. Pada kelas Brown spot, sebanyak 8 data berhasil diprediksi dengan benar. Terdapat 1 data yang salah diprediksi sebagai Bacterial leaf blight dan 1 data lainnya salah diprediksi sebagai Leaf smut. Pola ini menunjukkan bahwa karakteristik Brown spot memiliki kemiripan dengan dua kelas lainnya, sehingga masih menimbulkan kesalahan klasifikasi. Pada kelas Leaf smut, seluruh data berhasil diklasifikasikan dengan benar sebanyak 10 data. Tidak terdapat kesalahan prediksi ke kelas lain. Hal ini menunjukkan bahwa model sangat konsisten dalam mengenali ciri Leaf smut. relatif mirip.

Perhitungan akurasi dilakukan menggunakan rumus akurasi multi-kelas:

$$Akurasi = \frac{Prediksi\ Benar}{Total\ Sampel}$$

Dengan nilai prediksi benar (TP) = 27 dan total sampel uji = 30, diperoleh:

$$Akurasi = \frac{27}{30} = 0,90$$

Analisis hasil menunjukkan bahwa model memiliki kinerja yang baik dengan akurasi keseluruhan sebesar 90%. Berdasarkan hasil evaluasi, pada data uji terdapat sampel yang mengalami kesalahan klasifikasi pada kelas Bacterial leaf blight dan Brown spot. Visualisasi Grid-CAM digunakan untuk menganalisis area atensi model pada prediksi yang tidak sesuai dengan label sebenarnya.



Gambar 4.47 Grid-CAM misclassification: (a) Bacterial leaf blight, (b) Brown spot, (c) Brown spot.

Berdasarkan pada Gambar 4.47, Grid-CAM pada sampel data uji yang mengalami kesalahan klasifikasi. Pada Gambar (a), citra dengan label sebenarnya Bacterial leaf blight diprediksi sebagai Leaf smut, di mana area atensi model lebih terfokus pada bagian daun yang memanjang dan relatif homogen, sehingga ciri lesi khas Bacterial leaf blight kurang menjadi fokus utama. Pada Gambar (b), citra kelas Brown spot diprediksi sebagai Leaf smut, dengan aktivasi Grid-CAM yang menyebar dan tidak terlokalisasi secara jelas pada bercak penyakit utama. Sementara itu, pada Gambar (c), citra Brown spot diprediksi sebagai Bacterial leaf blight, di mana perhatian model terkonsentrasi pada area tertentu pada permukaan daun yang memiliki

kemiripan pola visual dengan kelas tersebut, sehingga menyebabkan kesalahan klasifikasi.

4.3.4 Evaluasi Skenario 4 – Full Fine Tuning (Semua Layer Trainable)

Pada skenario ini, seluruh layer pada arsitektur InceptionV3 dibuat trainable sehingga proses pembelajaran dilakukan dari layer awal hingga layer akhir. Evaluasi pada skenario ini menampilkan hasil pelatihan model secara menyeluruh yang meliputi grafik loss dan akurasi training serta validation, confusion matrix pada data uji, dan perbandingan kinerja model menggunakan tiga optimizer yaitu Adam, AdamW, dan SGD. Setiap optimizer diuji dengan konfigurasi parameter yang sama agar perbedaan performa yang dihasilkan benar-benar mencerminkan pengaruh pemilihan optimizer pada full fine tuning.

4.3.1.1 Evaluasi Skenario 4 dengan Optimizer Adam

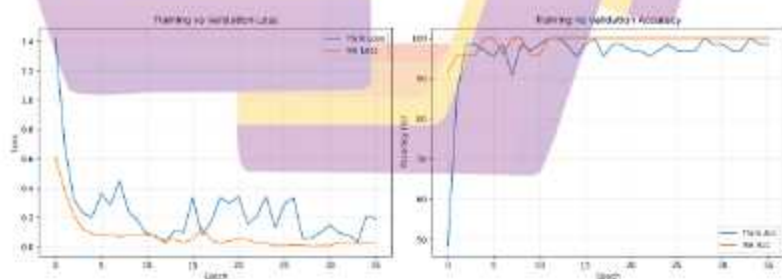
Pada bagian ini ditampilkan hasil pelatihan model pada Skenario 4 menggunakan optimizer Adam, yang dilakukan sebanyak 10 percobaan. Tabel 4.18 memperlihatkan akurasi data testing dari masing-masing percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.18 Akurasi Testing 10 Percobaan Skenario 4 (Optimizer Adam)

Percobaan	Akurasi	F1-Score
Percobaan 1	86,66	86,44
Percobaan 2	90	89,95

Percobaan	Akurasi	F1-Score
Percobaan 3	93,33	93.32
Percobaan 4	90	89.93
Percobaan 5	86,66	85.73
Percobaan 6	83,33	83.07
Percobaan 7	86,66	86.60
Percobaan 8	86,66	86.23
Percobaan 9	93,33	93.46
Percobaan 10	90	89.50

Dari 10 percobaan yang dilakukan pada Skenario 4, diperoleh variasi nilai akurasi pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan dengan nilai akurasi dan F1-Score tertinggi sebagai representasi performa maksimum model. Visualisasi ini mencakup grafik training accuracy, validation accuracy, training loss, dan validation loss, sebagaimana ditunjukkan pada Gambar 4.48.



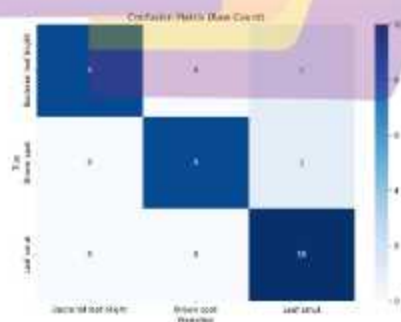
Gambar 4.48 Grafik Training Val Acc dan Loss Skenario 4 (Adam)

Berdasarkan grafik training dan validation loss pada Gambar 4.48, menunjukkan bahwa pada Skenario 4 dengan optimizer Adam, nilai training loss

dan validation loss menurun tajam pada fase awal pelatihan. Validation loss sudah berada pada nilai sangat rendah sejak sekitar epoch 5–7 dan cenderung stabil hingga akhir pelatihan. Training loss juga terus menurun meskipun sempat berfluktuasi ringan, yang masih wajar karena seluruh layer dilatih secara penuh.

Pada grafik akurasi, akurasi training dan validasi meningkat sangat cepat dan mencapai nilai di atas 95% sejak sekitar epoch 3–4. Akurasi validasi bahkan mencapai 100% pada beberapa epoch dan bertahan stabil tanpa penurunan signifikan. Selisih antara akurasi training dan validasi relatif kecil, menunjukkan proses pembelajaran yang stabil dan kemampuan generalisasi model yang baik.

Meskipun jumlah maksimum epoch ditetapkan 100, mekanisme early stopping menghentikan pelatihan lebih awal, yaitu sekitar epoch 35, karena tidak ditemukan lagi peningkatan berarti pada kinerja validasi. Hal ini menunjukkan bahwa model telah mencapai kondisi konvergen lebih cepat pada konfigurasi full fine tuning dengan optimizer Adam. Untuk melihat detail kesalahan klasifikasi pada masing-masing kelas, hasil evaluasi selanjutnya disajikan dalam bentuk confusion matrix pada Gambar 4.49.



Gambar 4.49 Confusion matrix Skenario 4 (Adam)

Berdasarkan confusion matrix pada Gambar 4.49, menunjukkan hasil klasifikasi model pada Skenario 4 dengan optimizer Adam untuk tiga kelas penyakit daun padi. Pada kelas Bacterial leaf blight, sebanyak 9 citra berhasil diklasifikasikan dengan benar, sementara 1 citra salah diprediksi sebagai Leaf smut. Kelas Brown spot memiliki 9 citra yang terklasifikasi dengan benar dan 1 citra salah diprediksi sebagai Leaf smut. Pada kelas Leaf smut, seluruh 10 citra berhasil diklasifikasikan dengan benar tanpa kesalahan.

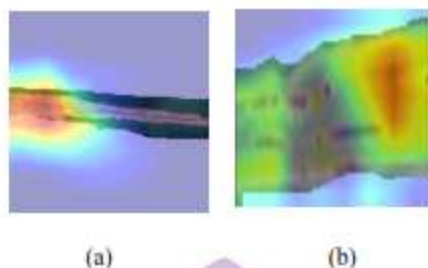
Perhitungan akurasi pada klasifikasi multi-kelas dilakukan berdasarkan jumlah prediksi benar dibandingkan jumlah keseluruhan sampel uji. Rumus akurasi yang digunakan adalah:

$$\text{Akurasi} = \frac{\text{Prediksi Benar}}{\text{Total Sampel}}$$

Dengan prediksi benar sebanyak 28 sampel dan total sampel uji sebanyak 30, diperoleh nilai akurasi:

$$\text{Akurasi} = \frac{28}{30} = 0,93$$

Berdasarkan hasil evaluasi, pada data uji terdapat sampel yang mengalami kesalahan klasifikasi pada kelas Bacterial leaf blight dan Brown spot. Visualisasi Grid-CAM digunakan untuk menganalisis area atensi model pada prediksi yang tidak sesuai dengan label sebenarnya.



Gambar 4.50 Grid-CAM misclassification: (a) Bacterial leaf blight, (b) Brown spot.

Gambar 4.50 Grid-CAM pada sampel data uji yang mengalami kesalahan klasifikasi. Pada Gambar (a), citra dengan label sebenarnya Bacterial leaf blight diprediksi sebagai Leaf smut, di mana atensi model terfokus pada bagian daun yang memanjang dan relatif homogen, sehingga area lesi khas Bacterial leaf blight kurang menjadi fokus. Pada Gambar (b), citra Brown spot diprediksi sebagai Leaf smut, dengan pola atensi yang menyebar dan tidak terlokalisasi pada bercak penyakit utama, yang mengindikasikan keterbatasan model dalam mengekstraksi fitur diskriminatif Brown spot.

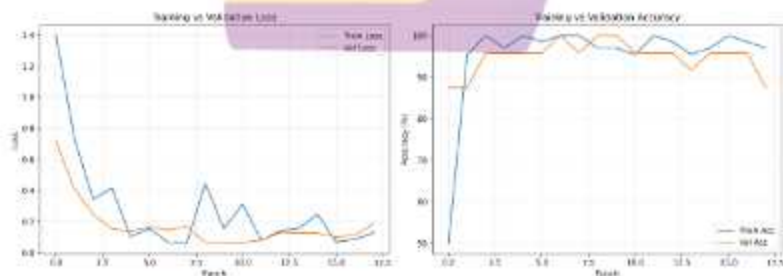
4.3.1.2 Evaluasi Skenario 4 dengan Optimizer AdamW

Pada bagian ini ditampilkan hasil pelatihan model pada Skenario 4 menggunakan optimizer AdamW, yang dilakukan sebanyak 10 percobaan. Tabel 4.19 memperlihatkan akurasi data testing dari masing-masing percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.19 Akurasi Testing 10 Percobaan Skenario 4 (Optimizer AdamW)

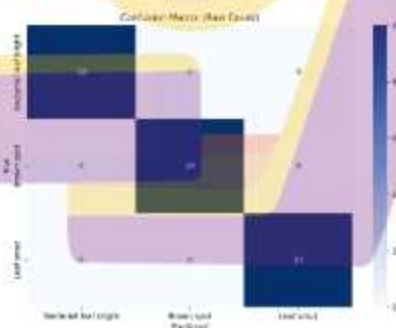
Percobaan	Akurasi	F1-Score
Percobaan 1	90	89.97
Percobaan 2	90	89.50
Percobaan 3	83,33	82.56
Percobaan 4	96,66	96.66
Percobaan 5	100	100.00
Percobaan 6	83,33	83.07
Percobaan 7	90	89.50
Percobaan 8	86,66	86.44
Percobaan 9	83,33	81.51
Percobaan 10	86,66	85.73

Dari 10 percobaan yang dilakukan pada Skenario 4, diperoleh variasi nilai akurasi pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan dengan nilai akurasi tertinggi sebagai representasi performa maksimum model. Visualisasi ini mencakup grafik training accuracy, validation accuracy, training loss, dan validation loss, sebagaimana ditunjukkan pada Gambar 4.51.



Gambar 4.51 Grafik Training Val Acc dan Loss Skenario 4 (AdamW)

Berdasarkan grafik training dan validation loss pada Gambar 4.51, Grafik terlihat bahwa nilai loss training dan loss validasi mengalami penurunan yang sangat cepat pada awal pelatihan dan telah mencapai nilai yang relatif rendah sejak sekitar epoch ke-4 hingga ke-6. Setelah fase tersebut, kedua nilai loss cenderung stabil dengan fluktuasi kecil hingga akhir pelatihan, yang menandakan proses optimasi berjalan dengan baik dan konsisten. Pada grafik akurasi, akurasi training meningkat tajam dari sekitar 50% pada epoch awal hingga berada pada kisaran 95–100% sejak epoch ke-2, sementara akurasi validasi juga meningkat dan stabil pada rentang 87–100% sepanjang proses pelatihan. Meskipun maksimum epoch ditetapkan lebih besar, mekanisme early stopping menghentikan pelatihan lebih awal, yaitu sekitar epoch ke-18, karena performa validasi tidak lagi menunjukkan peningkatan yang signifikan. Evaluasi lanjutan dilakukan menggunakan confusion matrix ditunjukkan pada Gambar 4.52.



Gambar 4.52 Confusion matrix Skenario 4 (AdamW)

Berdasarkan confusion matrix pada Gambar 4.52, hasil klasifikasi model pada Skenario 4 dengan optimizer AdamW menunjukkan performa yang sangat

optimal. Pada kelas Bacterial leaf blight, seluruh 10 data uji berhasil diprediksi dengan benar tanpa adanya kesalahan klasifikasi. Hal yang sama juga terjadi pada kelas Brown spot, di mana 10 data uji terklasifikasi dengan benar dan tidak terdapat data yang salah diprediksi ke kelas lain. Sementara itu, kelas Leaf smut juga menunjukkan performa sempurna dengan 10 data uji berhasil diklasifikasikan secara tepat tanpa kesalahan prediksi. Dengan demikian, seluruh kelas penyakit daun padi berhasil dikenali dengan sangat baik oleh model.

Perhitungan akurasi pada klasifikasi multi-kelas dilakukan berdasarkan jumlah prediksi benar dibandingkan jumlah keseluruhan sampel uji. Rumus akurasi yang digunakan adalah:

$$Akurasi = \frac{\text{Prediksi Benar}}{\text{Total Sampel}}$$

Dengan prediksi benar sebanyak 30 sampel dan total sampel uji sebanyak 30, diperoleh nilai akurasi:

$$Akurasi = \frac{30}{30} = 100$$

Hasil ini menunjukkan bahwa model pada Skenario 4 dengan optimizer AdamW mampu mencapai akurasi 100% pada data pengujian. Tidak adanya kesalahan klasifikasi mengindikasikan bahwa penerapan full fine-tuning, di mana seluruh layer InceptionV3 bersifat trainable, memungkinkan model menyesuaikan bobot secara sangat efektif terhadap karakteristik visual masing-masing penyakit daun padi. Secara keseluruhan, hasil ini menegaskan bahwa optimizer AdamW

memberikan kemampuan generalisasi yang sangat baik, bahkan ketika digunakan pada dataset dengan ukuran yang relatif kecil.

4.3.1.3 Evaluasi Skenario 4 dengan Optimizer SGD

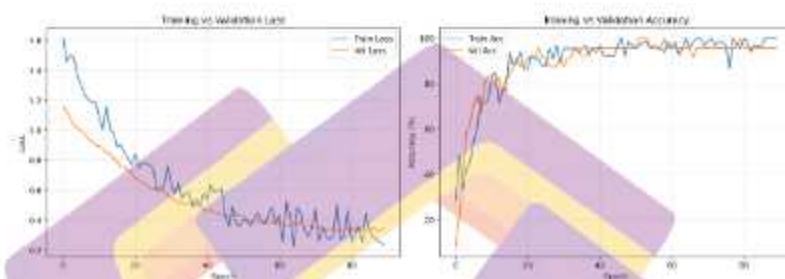
Pada bagian ini ditampilkan hasil pelatihan model pada Skenario 4 menggunakan optimizer SGD, yang dilakukan sebanyak 10 percobaan. Tabel 4.20 memperlihatkan akurasi data testing dari masing-masing percobaan. Nilai rata-rata digunakan sebagai performa akhir model, sementara akurasi tertinggi dicatat sebagai informasi tambahan.

Tabel 4.20 Akurasi Testing 10 Percobaan Skenario 4 (Optimizer SGD)

Percobaan	Akurasi	F1-Score
Percobaan 1	73,33	75.56
Percobaan 2	76,66	71.21
Percobaan 3	76,66	78.57
Percobaan 4	83,33	78.67
Percobaan 5	73,33	71.35
Percobaan 6	80	68.15
Percobaan 7	80	89.93
Percobaan 8	70	79.37
Percobaan 9	60	75.38
Percobaan 10	86,66	86.44

Dari 10 percobaan yang dilakukan pada Skenario 4, diperoleh variasi nilai akurasi pada data testing. Untuk tujuan visualisasi proses pelatihan, dipilih percobaan

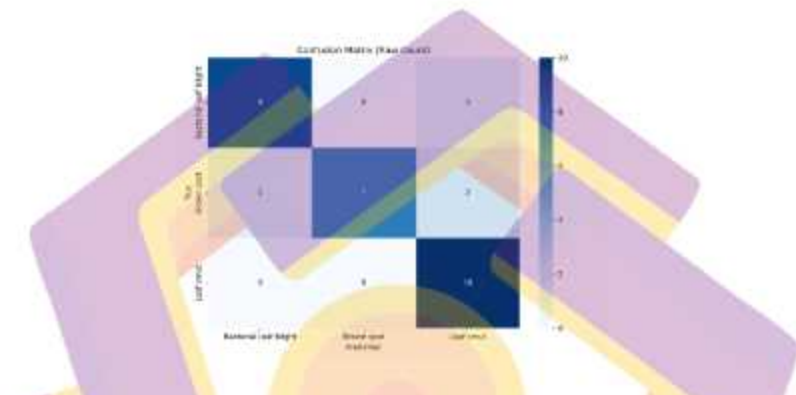
dengan nilai akurasi tertinggi sebagai representasi performa maksimum model. Visualisasi ini mencakup grafik training accuracy, validation accuracy, training loss, dan validation loss, sebagaimana ditunjukkan pada Gambar 4.53.



Gambar 4.53 Grafik Training Val Acc dan Loss Skenario 4 (SGD)

Berdasarkan grafik training dan validation loss pada 4.53, Grafik terlihat bahwa nilai loss training dan loss validasi mengalami penurunan yang konsisten sejak awal pelatihan. Loss training menurun dari sekitar 1,6 pada epoch awal hingga berada di kisaran 0,25–0,35, sementara loss validasi turun dari sekitar 1,15 dan kemudian stabil pada rentang 0,33–0,35. Penurunan yang relatif seimbang antara loss training dan validasi menunjukkan bahwa proses pembelajaran berjalan dengan baik tanpa indikasi overfitting yang signifikan. Pada grafik akurasi, akurasi training meningkat tajam dari sekitar 30% hingga mencapai kisaran 95–100% mulai sekitar epoch ke-25, sedangkan akurasi validasi juga meningkat dari sekitar 10% dan stabil pada kisaran 95–100% hingga akhir pelatihan. Meskipun jumlah maksimum epoch ditetapkan lebih besar, mekanisme early stopping menghentikan proses pelatihan pada sekitar epoch ke-90, karena performa validasi sudah tidak menunjukkan

peningkatan yang berarti. Hal ini menunjukkan bahwa model telah mencapai kondisi konvergen dan mampu mempertahankan performa yang stabil dengan kemampuan generalisasi yang baik. Evaluasi lanjutan dilakukan menggunakan confusion matrix sebagaimana ditunjukkan pada Gambar 4.54.



Gambar 4.54 Confusion matrix Skenario 4 (SGD)

Berdasarkan confusion matrix pada Gambar 4.54, Model mampu melakukan klasifikasi dengan cukup baik pada ketiga kelas penyakit daun padi, meskipun masih terdapat beberapa kesalahan. Pada kelas Bacterial leaf blight, dari total 10 data uji terdapat 9 data yang berhasil diprediksi dengan benar, sementara 1 data salah diprediksi sebagai Leaf smut. Pada kelas Brown spot, performa model relatif lebih rendah dibandingkan kelas lainnya, di mana 7 data berhasil diklasifikasikan dengan benar, sedangkan 3 data mengalami kesalahan, yaitu 1 data diprediksi sebagai Bacterial leaf blight dan 2 data diprediksi sebagai Leaf smut. Sementara itu, kelas Leaf smut menunjukkan performa yang sangat baik karena seluruh 10 data uji berhasil diprediksi dengan benar tanpa kesalahan klasifikasi.

Kesalahan prediksi terutama terjadi pada kelas Brown spot, yang kemungkinan disebabkan oleh kemiripan karakteristik visual dengan dua kelas lainnya, namun secara keseluruhan model tetap menunjukkan kemampuan klasifikasi yang cukup baik dan seimbang.

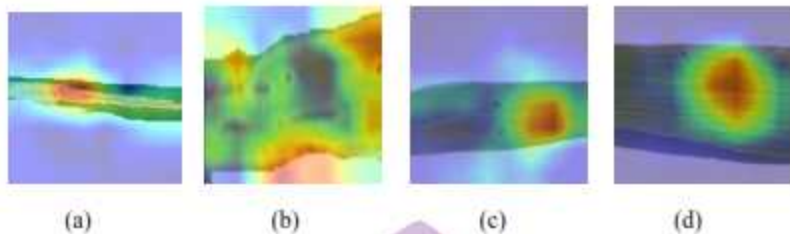
Jumlah prediksi benar diperoleh dari penjumlahan nilai pada diagonal utama confusion matrix, yaitu :

$$\text{Akurasi} = \frac{\text{Prediksi Benar}}{\text{Total Sampel}}$$

Dengan prediksi benar sebanyak 26 sampel dan total sampel uji sebanyak 30, diperoleh nilai akurasi:

$$\text{Akurasi} = \frac{26}{30} = 0,86$$

Hasil ini menunjukkan bahwa konfigurasi fine-tuning dengan optimizer SGD mampu memberikan performa klasifikasi yang cukup baik. Berdasarkan hasil evaluasi, pada data uji terdapat sampel yang mengalami kesalahan klasifikasi pada kelas Bacterial leaf blight dan Brown spot. Visualisasi Grid-CAM digunakan untuk menganalisis area atensi model pada prediksi yang tidak sesuai dengan label sebenarnya.



Gambar 4.55 Grid-CAM misclassification: (a) Bacterial leaf blight, (b) Brown spot, (c) Brown spot, (d) Brown spot.

Gambar 4.55 menampilkan visualisasi Grid-CAM pada sampel data uji yang mengalami kesalahan klasifikasi. Pada Gambar (a), citra dengan label sebenarnya Bacterial leaf blight diprediksi sebagai Leaf smut, di mana atensi model lebih terfokus pada area daun yang memanjang dan relatif homogen sehingga ciri lesi khas penyakit kurang menjadi fokus. Pada Gambar (b) dan (c), citra kelas Brown spot diprediksi sebagai Leaf smut, dengan pola atensi yang menyebar dan tidak terlokalisasi secara jelas pada bercak penyakit utama. Sementara itu, pada Gambar (d), citra Brown spot diprediksi sebagai Bacterial leaf blight, di mana aktivasi Grid-CAM terkonsentrasi pada area tertentu pada permukaan daun yang memiliki kemiripan pola visual dengan kelas tersebut, sehingga menyebabkan kesalahan klasifikasi.

4.4 Komparasi Performa Model dan Kesimpulan

4.4.1 Perbandingan Performa Antar Skenario

Pada tahap ini, setiap skenario pengujian dievaluasi melalui 10 kali percobaan atau trial yang dijalankan secara terpisah. Setiap trial menggunakan konfigurasi skenario yang sama agar hasil yang diperoleh konsisten dan dapat dibandingkan secara objektif. Nilai akurasi dari masing masing trial kemudian

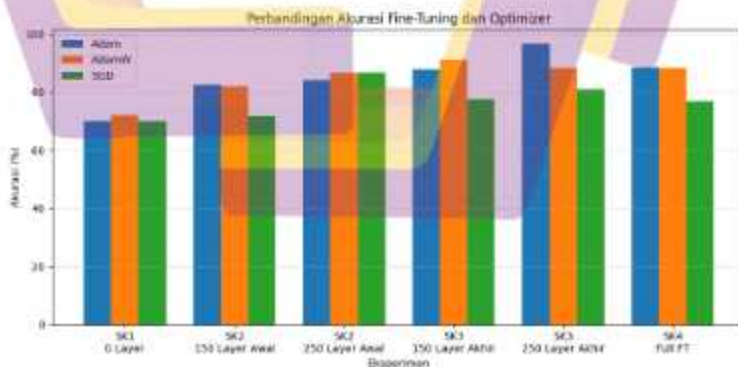
dicatat dan dihitung nilai rata ratanya. Perhitungan nilai average akurasi ini digunakan untuk merepresentasikan kinerja keseluruhan dari setiap skenario. Dengan pendekatan ini, hasil perbandingan akurasi antar skenario menjadi lebih stabil dan tidak bergantung pada satu kali percobaan saja.

Tabel 4.21 Hasil Perbandingan Skenario Fine Tuning Optimizer

Skenario	Pendekatan Fine Tuning	Layer Trainable	Akurasi (Adam)	Akurasi (AdamW)	Akurasi (SGD)
SK 1	Non trainable	0 Layer	70%	72%	70%
SK 2	Fine tuning layer awal	150 Layer	82,59%	82%	71,66%
	Fine tuning layer awal	250 Layer	84,07%	86,66%	86,66%
SK 3	Fine tuning layer akhir	150 Layer	87,77%	91%	77,5%
	Fine tuning layer akhir	250 Layer	96,66%	88,33%	80,83%
SK 4	Full fine tuning	Semua layer trainable	88,51%	88,33%	76,66%

Berdasarkan Tabel 4.21, hasil pengujian pada tabel tersebut, dapat disimpulkan bahwa penerapan fine-tuning memberikan dampak signifikan terhadap peningkatan performa model dibandingkan skenario non-trainable yang hanya mencapai akurasi 70–72%. Pada fine-tuning pada layer awal menunjukkan peningkatan akurasi seiring bertambahnya jumlah layer trainable, di mana penggunaan optimizer juga berpengaruh terhadap hasil, dengan AdamW dan SGD mampu mencapai akurasi tertinggi sebesar 86,66% pada konfigurasi 250 layer. Peningkatan performa yang lebih signifikan terjadi ketika fine-tuning dilakukan dengan membuka 250 lapisan terakhir model. Lapisan-lapisan ini berada paling dekat dengan lapisan output sehingga berperan dalam mengekstraksi fitur tingkat tinggi yang lebih spesifik terhadap tugas klasifikasi pada penelitian ini.

Dengan konfigurasi tersebut, model tetap mempertahankan fitur umum yang telah dipelajari pada lapisan awal, seperti pola dasar dan tekstur, sekaligus menyesuaikan fitur tingkat lanjut agar lebih selaras dengan karakteristik dataset penyakit daun padi. Kondisi ini memungkinkan proses pembelajaran menjadi lebih optimal dan berdampak pada peningkatan akurasi model. Sementara itu, full fine-tuning menghasilkan performa yang relatif stabil namun tidak melampaui skenario terbaik, sehingga menunjukkan bahwa melatih seluruh layer dan penggunaan optimizer yang berbeda tidak selalu memberikan peningkatan signifikan. Meskipun akurasi global model sudah cukup tinggi, hasil evaluasi menunjukkan bahwa kelas Brown Spot masih menjadi kelas yang paling sulit terdeteksi secara konsisten akibat kemiripan karakteristik visual dengan kelas penyakit lain, sehingga diperlukan strategi tambahan seperti augmentasi data yang lebih spesifik dan pemilihan optimizer yang lebih adaptif untuk meningkatkan performa pada kelas tersebut.



Gambar 4.56 Grafik perbandingan akurasi fine tuning dan optimizer

Dari ketiga optimizer Adam, AdamW, dan SGD menunjukkan bahwa strategi fine tuning dan pemilihan optimizer memberikan pengaruh yang berbeda terhadap akurasi model InceptionV3. Pada optimizer Adam, hasil terbaik diperoleh pada skenario fine tuning layer akhir dengan 250 layer trainable yang menghasilkan akurasi tertinggi sebesar 96,66%. Pada optimizer AdamW, performa terbaik dicapai pada fine tuning layer akhir dengan 150 layer trainable dengan akurasi 91%. Sementara itu, pada optimizer SGD, konfigurasi terbaik diperoleh pada fine tuning layer awal dengan 250 layer trainable dengan akurasi 86,66%. Secara keseluruhan, fine tuning layer akhir cenderung memberikan kinerja lebih optimal pada optimizer Adam dan AdamW, sedangkan SGD lebih efektif pada fine tuning layer awal dengan jumlah layer trainable yang lebih besar. Berdasarkan hasil tersebut, model dengan konfigurasi akurasi tertinggi dipilih sebagai model terbaik. Model dari skenario ini digunakan pada tahap berikutnya untuk komparasi dengan penelitian sebelumnya yang memakai dataset yang sama.

Selanjutnya setelah akurasi dan $f1$ -score di dapatkan lalu perbandingan beberapa penelitian terdahulu yang menggunakan dataset yang sama, yaitu dataset penyakit daun padi. Setiap penelitian menerapkan pendekatan yang berbeda sesuai metode yang dikembangkan, sehingga menghasilkan variasi performa pada tugas klasifikasi. Komparasi ini bertujuan menunjukkan bagaimana setiap metode bekerja pada kondisi dataset yang sama serta melihat posisi kinerja model dalam penelitian ini terhadap studi sebelumnya. Informasi yang ditampilkan berfokus pada model yang digunakan, skenario pengujian, dan akurasi yang dilaporkan. Ringkasan komparasi ditampilkan pada tabel berikut.

Tabel 4.22 Perbandingan dengan Penelitian Sebelumnya pada Dataset yang Sama

No	Author	Method	Akurasi
1	Yogesh Kumar Rathore et al. (2023)	CNN ringan LW17, 17 layer, augmentasi rotasi & zoom, Adam optimizer	93,75%
2	Muhammad Anwarul Azim et al. (2021)	Fitur warna, bentuk, tekstur + segmentasi HSV, XGBoost & SVM RBF	86,58%
3	Anam Islam et al. (2021)	Segmentasi local threshold + CNN (DenseNet121)	91,67%
4	Jody Ririt Krido Suseno et al. (2023)	VGG16 pretrained + segmentasi thresholding & k-means	91,66%
5	Ike Verawati & Ridwan Aunurrohm (2024)	KNN + GLCM + Canny Edge Detection	91,67%
7	Fadel Muhammad Fernando et al. (2024)	InceptionV3	90,00%
8	Proposed Skenario 3	InceptionV3 Fine-Tuning	96,66%

Berdasarkan Tabel 4.22, terlihat bahwa setiap penelitian menghasilkan akurasi yang berbeda meskipun menggunakan dataset yang sama. Studi [28] menggunakan model CNN ringan LW17 dan memperoleh akurasi 93,75%. Penelitian [25] menggunakan metode ekstraksi fitur manual berupa warna, bentuk, dan tekstur yang dikombinasikan dengan XGBoost dan SVM, dengan akurasi 86,58%. Pendekatan [26] menggunakan segmentasi dan CNN melalui penggunaan VGG16, ResNet50, dan DenseNet121 menghasilkan akurasi 91,67%, sedangkan penelitian [27] menggunakan model VGG16 pretrained dengan segmentasi k-means dan mencapai 91,66%. Studi [56] Metode KNN juga memberikan akurasi 91,67% melalui kombinasi GLCM dan Canny Edge Detection. Studi [20] menggunakan InceptionV3 pada dataset daun padi yang terbatas dan menghasilkan akurasi 90%.

Perbedaan akurasi tersebut menunjukkan variasi performa setiap metode. Pendekatan CNN umumnya memberikan hasil yang lebih stabil, sedangkan metode berbasis fitur manual tetap dapat bekerja dengan baik selama proses ekstraksi

dilakukan secara tepat. Pada penelitian ini, scenario 3 full fine-tuning pada InceptionV3 menghasilkan akurasi 96,66%, tertinggi dibandingkan penelitian sebelumnya pada dataset yang sama. Hasil ini menunjukkan bahwa optimasi melalui fine-tuning dapat meningkatkan performa model, terutama ketika bekerja dengan jumlah data yang terbatas.



BAB 5

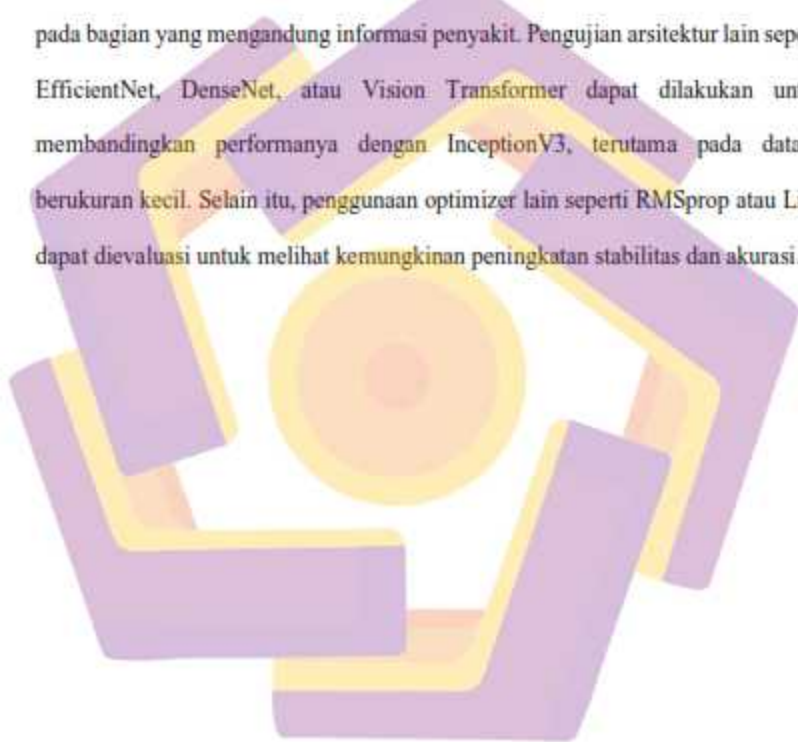
PENUTUP

5.1 Kesimpulan

1. Penerapan fine tuning terbukti meningkatkan akurasi klasifikasi penyakit daun padi secara signifikan pada dataset berukuran kecil. Model dengan seluruh layer non trainable hanya menghasilkan akurasi 70%. Fine tuning parsial memberikan peningkatan bertahap, dengan hasil terbaik diperoleh pada fine tuning layer akhir sebanyak 250 layer trainable menggunakan optimizer Adam yang mencapai akurasi 96,66%. Peningkatan ini menunjukkan adanya selisih akurasi sebesar 26,66% dibandingkan model tanpa fine tuning. Sementara itu, full fine tuning tidak selalu memberikan hasil optimal dan pada beberapa konfigurasi justru menghasilkan akurasi lebih rendah dibandingkan fine tuning parsial.
2. Pemilihan optimizer berpengaruh langsung terhadap efektivitas proses fine tuning pada InceptionV3. Optimizer Adam menunjukkan performa paling optimal dengan akurasi tertinggi 96,66%, diikuti oleh AdamW dengan akurasi terbaik 91%, dan SGD dengan akurasi maksimum 86,66%. Hasil ini menunjukkan bahwa kombinasi fine tuning layer akhir dan optimizer Adam merupakan konfigurasi paling efektif untuk klasifikasi penyakit daun padi pada dataset terbatas dan memberikan peningkatan performa dibandingkan penelitian sebelumnya pada dataset yang sama.

5.2 Saran

Penelitian selanjutnya dapat mempertimbangkan penambahan jumlah data melalui pengumpulan citra baru atau teknik augmentasi yang lebih beragam agar model memperoleh variasi pola yang lebih luas. Pendekatan segmentasi daun juga dapat diterapkan untuk memisahkan area latar belakang sehingga model lebih fokus pada bagian yang mengandung informasi penyakit. Pengujian arsitektur lain seperti EfficientNet, DenseNet, atau Vision Transformer dapat dilakukan untuk membandingkan performanya dengan InceptionV3, terutama pada dataset berukuran kecil. Selain itu, penggunaan optimizer lain seperti RMSprop atau Lion dapat dievaluasi untuk melihat kemungkinan peningkatan stabilitas dan akurasi.



DAFTAR PUSTAKA

- [1] H. Ali dan H. Kim, "Utilizing Convolutional Neural Networks for the Effective Classification of Rice Leaf Diseases Through a Deep Learning Approach," 2024.
- [2] G. Zhou, W. Zhang, A. Chen, M. He, dan X. Ma, "Rapid Detection of Rice Disease Based on FCM-KM and Faster R-CNN Fusion," *IEEE Access*, vol. 7, hal. 143190–143206, 2019, doi: 10.1109/ACCESS.2019.2943454.
- [3] I. Verawati dan R. A. A. Aunurrohm, "Klasifikasi Penyakit Daun Padi Menggunakan KNN dengan GLCM dan Canny Edge Detection," *J. Media Inform. Budidarma*, vol. 8, no. 1, hal. 517, 2024, doi: 10.30865/mib.v8i1.6906.
- [4] A. Jinan, B. H. Hayadi, dan U. P. Utama, "Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Convolutional Neural Network Melalui Citra Daun (Multilayer Perceptron)," *J. Comput. Eng. Sci.*, vol. 1, no. 2, hal. 37–44, 2022.
- [5] M. Sharma, C. J. Kumar, J. Talukdar, T. P. Singh, G. Dhiman, dan A. Sharma, "Identification of rice leaf diseases and deficiency disorders using a novel DeepBatch technique," 2023.
- [6] Y. Cao *et al.*, "Applying spectral fractal dimension index to predict the SPAD value of rice leaves under bacterial blight disease stress," *Plant Methods*, hal. 1–15, 2022, doi: 10.1186/s13007-022-00898-8.
- [7] T. P. Bhutia, Y. D. Rao, P. Kharel, P. Rai, M. Rana, dan S. Srivastava, "A comprehensive review on brown spot disease of rice: Etiology , epidemiology , management strategies and future directions," vol. x, no. x, 2024.
- [8] S. Poonguzhali, "A Hybrid Approach for Prediction and Classification in Rice Plant Disease Using Machine Learning Algorithm," vol. 45, no. 2, hal. 468–478, 2024.
- [9] P. Pai *et al.*, "Deep learning-based automatic diagnosis of rice leaf diseases

- using ensemble CNN models,” hal. 1–13, 2025.
- [10] A. R. Al-shamasneh dan R. W. Ibrahim, “Classification of tomato leaf images for detection of plant disease using conformable polynomials image features,” vol. 13, no. April, 2024, doi: 10.1016/j.mex.2024.102844.
- [11] R. H. Bakhtiar, Y. Dani, dan D. Suandi, “MACHINE LEARNING BASED CLASSIFICATION OF RICE LEAF DISEASES : A COMPARATIVE STUDY,” hal. 1–23, 2025.
- [12] M. E. Marlier, K. I. Brenner, S. J. Koepke, J. J. Watkins, dan S. D. Minter, “Classification of Corn Diseases using Random Forest , Neural Network , and Naive Bayes Methods,” hal. 0–10, 2022, doi: 10.1088/1742-6596/2406/1/012023.
- [13] K. Adil, D. Idress, O. Alsadig, A. Gadalla, dan G. P. Baitu, “Machine Learning-based for Automatic Detection of Corn-Plant Diseases Using Image Processing,” vol. 30, no. 3, hal. 464–476, 2024, doi: 10.15832/ankutbd.1288298.
- [14] C. D. N. Kiran S M, “Plant Leaf Disease Detection Using Efficient Image Processing and Machine Learning Algorithms,” vol. 4, no. 6, hal. 840–848, 2023, doi: 10.18196/jrc.v4i6.20342.
- [15] B. K. Hatuwal, A. Shakya, dan B. Joshi, “Plant Leaf Disease Recognition Using Random Forest, KNN, SVM and CNN,” vol. 62, hal. 13–19, 2020.
- [16] M. S. Krishna, P. Machado, R. I. Otuka, S. W. Yahaya, F. Neves, dan I. K. Ihianle, “Plant Leaf Disease Detection Using Deep Learning : A Multi-Dataset Approach,” hal. 1–24, 2025.
- [17] M. Dutta *et al.*, “Rice Leaf Disease Classification—A Comparative Approach Using Convolutional Neural Network (CNN), Cascading Autoencoder with Attention Residual U-Net (CAAR-U-Net), and MobileNet-V2 Architectures,” *Technologies*, vol. 12, no. 11, 2024, doi: 10.3390/technologies12110214.
- [18] B. W. Sari, “Transfer Learning-Based Convolutional Neural Network for

- Accurate Detection of Rice Leaf Disease in Precision Agriculture,” vol. 11, no. 3, hal. 420–432, 2025.
- [19] T. Ahad, Y. Li, B. Song, dan T. Bhuiyan, “Comparison of CNN-based deep learning architectures for rice diseases classification,” *Artif. Intell. Agric.*, vol. 9, hal. 22–35, 2023, doi: 10.1016/j.aiaa.2023.07.001.
- [20] F. M. Firmando, D. R. I. M. Setiadi, A. R. Muslikh, dan S. W. Iriananda, “Analyzing InceptionV3 and InceptionResNetV2 with Data Augmentation for Rice Leaf Disease Classification,” *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 1, hal. 1–11, 2024, doi: 10.62411/faith.2024-4.
- [21] E. H. I. Eliwa, “Enhancing Skin Cancer Diagnosis Through Fine-Tuning of Pretrained Models: A Two-Phase Transfer Learning Approach.” *International Journal of Breast Cancer*, 2025. doi: <https://doi.org/10.1155/ijbc/4362941>.
- [22] A. T. Teweldemedhin, Mahtem, “Harnessing deep learning for wheat variety classification: a convolutional neural network and transfer learning approach,” vol. 2025, no. January, 2025, doi: 10.1002/jsfa.14378.
- [23] and V. D. J. Shah, H. Prajapati, “Rice Leaf Diseases,” *UCI Machine Learning Repository*, 2017.
- [24] H. B. Prajapati, J. Shah, D. Technologies, dan V. Dabhi, “Detection and Classification of Rice Plant Diseases,” no. August 2018, 2017, doi: 10.3233/IDT-170301.
- [25] M. A. Azim, M. K. Islam, M. M. Rahman, dan F. Jahan, “An effective feature extraction method for rice leaf disease classification,” *Telkommnika (Telecommunication Comput. Electron. Control.*, vol. 19, no. 2, hal. 463–470, 2021, doi: 10.12928/TELKOMNIKA.v19i2.16488.
- [26] A. Islam, R. Islam, S. M. R. Haque, dan S. M. M. Islam, “Rice Leaf Disease Recognition using Local Threshold Based Segmentation and Deep CNN,” no. October, hal. 35–45, 2021, doi: 10.5815/ijisa.2021.05.04.
- [27] J. Ririt, K. Suseno, Y. Azhar, dan A. E. Minarno, “Implementation of

- pretrained VGG16 model for rice leaf disease classification using image segmentation," vol. 4, no. 1, 2023.
- [28] T. Rathore, Y. K., Janghel, R. R., Swarup, C., Pandey, S. K., Kumar, A., Singh, K. U., & Singh, "Detection of rice plant disease from RGB and grayscale images using an LW17 deep learning model," *Electron. Res. Arch.*, vol. 31(5), hal. 2813–2833, 2023.
- [29] W. Setiawan, M. M. Suhadi, dan Y. D. Pramudita, "Inception-v3 with Reduce Learning Rate for Optimization of Lung Cancer Histopathology Classification," vol. 29, no. 2, hal. 561–570, 2024.
- [30] A. Eko, L. Aripa, Y. Azhar, dan Y. Munarko, "Classification of Malaria Cell Image Using Inception-V3 Architecture," vol. 7, no. June, hal. 273–278, 2023.
- [31] S. ABROL *et al.*, "Effect of Agro-Met Conditions on the Progression of Brown Leaf Spot Disease in Basmati-370 Rice," *Asian J. Microbiol. Biotechnol. Environ. Sci.*, vol. 24, no. 2, hal. 335–340, 2022, doi: 10.53550/ajmbes.2022.v24i02.020.
- [32] H. B. Prajapati, J. P. Shah, dan V. K. Dabhi, "Detection and classification of rice plant diseases," *Intell. Decis. Technol.*, vol. 11, no. 3, hal. 357–373, 2017, doi: 10.3233/IDT-170301.
- [33] L. Alzubaidi *et al.*, *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*. Springer International Publishing, 2021. doi: 10.1186/s40537-021-00444-8.
- [34] M. Halim, M. Noor, dan A. Olalekan, "A Survey on State-of-the-art Deep Learning Applications and Challenges".
- [35] A. Shrestha, "Review of Deep Learning Algorithms and Architectures," *IEEE Access*, vol. 7, hal. 53040–53065, 2019, doi: 10.1109/ACCESS.2019.2912200.
- [36] X. Zhao, L. Wang, Y. Zhang, X. Han, dan M. Deveci, "A review of convolutional neural networks in computer vision," *Artif. Intell. Rev.*, vol.

- 57, no. 4, hal. 1–43, 2024, doi: 10.1007/s10462-024-10721-6.
- [37] M. Krichen, “Convolutional Neural Networks : A Survey,” hal. 1–41, 2023.
- [38] A. Khan, A. Sohail, U. Zahoor, dan A. S. Qureshi, “A Survey of the Recent Architectures of Deep Convolutional Neural Networks,” hal. 1–70.
- [39] V. Risojevi, “DO WE STILL NEED IMAGENET PRE-TRAINING IN REMOTE SENSING SCENE CLASSIFICATION ?,” vol. XLIII, no. June, hal. 6–11, 2022.
- [40] Y. Alkhalifi dan A. Subekti, “Bell Pepper Leaf Disease Classification Using Fine-Tuned Transfer Learning,” vol. 23, no. 1, hal. 55–61, 2023, doi: 10.55981/jet.546.
- [41] W. Li, Q. Liu, Y. Wang, dan H. Li, “TRANSFER LEARNING WITH LIMITED SAMPLES FOR THE SAME SOURCE HYPERSPECTRAL REMOTE SENSING IMAGES CLASSIFICATION,” vol. XLIII, no. June, hal. 6–11, 2022.
- [42] A. T. Balarabe dan I. Jordanov, “A Deeper Look Into Remote Sensing Scene Image Misclassification by CNNs,” *IEEE Access*, vol. 12, no. September, hal. 123078–123098, 2024, doi: 10.1109/ACCESS.2024.3354976.
- [43] N. I. R. Red dan N. I. R. Red, “Study of CNN deep learning model for temporal remote sensing data processing to map rabi crops CBSI □ NDVI □,” vol. 16, no. 2, hal. 167–175, 2022.
- [44] J. Siddamsetty, M. Stricker, M. Charfuelan, M. Nuske, dan A. Dengel, “INTER-REGION TRANSFER LEARNING FOR LAND USE LAND COVER CLASSIFICATION,” vol. X, no. September, hal. 2–7, 2023.
- [45] G. Latif, S. E. Abdelhamid, R. E. Mallouhy, J. Alghazo, dan Z. A. Kazimi, “Deep Learning Utilization in Agriculture : Detection of Rice Plant Diseases Using an Improved CNN Model,” 2022.
- [46] K. Alam, M. Kabir, dan A. Rahman, “DenseNet201Plus : Cost-effective transfer-learning architecture for rapid leaf disease identification with

- attention mechanisms,” *Heliyon*, vol. 10, no. 15, hal. e35625, 2024, doi: 10.1016/j.heliyon.2024.e35625.
- [47] A. Pratondo, E. Elfahmi, dan A. Novianty, “Classification of *Curcuma longa* and *Curcuma zanthorrhiza* using transfer learning Agus,” hal. 1–15, 2022, doi: 10.7717/peerj-cs.1168.
- [48] M. R. Shoaib, H. M. Emara, J. Zhao, dan W. El-shafai, “Deep Learning Innovations in Diagnosing Diabetic Retinopathy : The Potential of Transfer Learning and the DiaCNN Model,” hal. 1–51.
- [49] Z. N. I. Zailan, S. A. Mostafa, A. Idrees, dan Z. Baharum, “Deep Learning Approach for Prediction of Brain Tumor from Small Number of MRI Images,” vol. 6, no. August, 2022.
- [50] F. A. Sierra-liñan, O. Iparraguirre-villanueva, V. Guevara-ponce, O. R. Paredes, dan F. Sierra-liñan, “Convolutional Neural Networks with Transfer Learning for Pneumonia Detection,” no. April 2023, 2022, doi: 10.14569/IJACSA.2022.0130963.
- [51] S. N. Yousafzai *et al.*, “Advanced clustering and transfer learning based approach for rice leaf disease segmentation and classificatio,” hal. 1–23, 2025, doi: 10.7717/peerj-cs.3018.
- [52] P. A. S. Rani dan N. S. Singh, “Paddy Leaf Symptom-based Disease Classification Using Deep CNN with ResNet-50,” *Int. J. Adv. Sci. Comput. Eng.*, vol. 4, no. 2, hal. 88–94, 2022, doi: 10.62527/ijasce.4.2.83.
- [53] S. X. C. Journal, “Leveraging the performance of Deep Learning Models for Corn Leaf Disease Diagnosis using DenseNet201 and Xception,” vol. 2, hal. 83–95, 2023.
- [54] R. Sharma, A. Singh, N. Z. Jhanjhi, M. Masud, E. S. Jaha, dan S. Verma, “Plant Disease Diagnosis and Image Classification Using Deep Learning,” 2022, doi: 10.32604/cmc.2022.020017.
- [55] A. Azis, A. Fadlil, T. Sutikno, I. S. Program, U. A. Dahlan, dan U. A. Purwokerto, “Optimization of Convolutional Neural Network (CNN) Using

Transfer Learning for Disease Identification in Rice Leaf Image Abdul,” vol. 8, no. 2, hal. 504–515, 2024.

- [56] I. Verawati dan R. A. A. Aunurrohim, “Klasifikasi Penyakit Daun Padi Menggunakan KNN dengan GLCM dan Canny Edge Detection,” *J. MEDIA Inform. BUDIDARMA*, vol. 8, no. 1, hal. 517, Jan 2024, doi: 10.30865/mib.v8i1.6906.

