

TESIS

**KLASIFIKASI PENYAKIT PADA TANAMAN TEH BERDASARKAN
CITRA DAUN MENGGUNAKAN CONVOLUTIONAL NEURAL
NETWORK**



Disusun oleh:

Nama : Liana Trihardlaningsih
NIM : 21.51.2098
Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2023

TESIS

**KLASIFIKASI PENYAKIT PADA TANAMAN TEH BERDASARKAN
CITRA DAUN MENGGUNAKAN CONVOLUTIONAL NEURAL
NETWORK**

**CLASSIFICATION OF DISEASES IN TEA PLANT BASED ON LEAF
IMAGES USING CONVOLUTIONAL NEURAL NETWORK**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Liana Trihardianingsih
NIM : 21.51.2098
Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2023

HALAMAN PENGESAHAN

**KLASIFIKASI PENYAKIT PADA TANAMAN TEH BERDASARKAN
CITRA DAUN MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK**

**CLASSIFICATION OF DISEASES IN TEA PLANT BASED ON LEAF
IMAGES USING CONVOLUTIONAL NEURAL NETWORK**

Dipersiapkan dan Disusun oleh

Liana Trihardianingsih

21.51.2098

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Rabu, 6 Desember 2023

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 6 Desember 2023

Rektor

Prof. Dr. M. Suyanto, M.M.

NIK. 190302001

HALAMAN PERSETUJUAN

**KLASIFIKASI PENYAKIT PADA TANAMAN TEH BERDASARKAN CITRA
DAUN MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK**

**CLASSIFICATION OF DISEASES IN TEA PLANT BASED ON LEAF
IMAGES USING CONVOLUTIONAL NEURAL NETWORK**

Dipersiapkan dan Disusun oleh

Liana Trihardianingsih

21.51.2098

Telah Ditujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Rabu, 6 Desember 2023

Pembimbing Utama

Anggota Tim Penguji

Dr. Andi Sunyoto, M.Kom

NIK. 190302052

Hanafi, S.Kom., M.Eng., Ph.D

NIK. 190302024

Pembimbing Pendamping

Dr. Kumara Ari Yuana, S.T., M.T.

NIK. 190302575

Tonny Hidayat, M.Kom., Ph.D

NIK. 190302182

Dr. Andi Sunyoto, M.Kom

NIK. 190302052

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 6 Desember 2023

Direktur Program Pascasarjana

Prof. Dr. Kusriani, M.Kom.

NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Liana Trihardianingsih
NIM : 21.51.2098
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
Klasifikasi Penyakit pada Tanaman Teh Berdasarkan Citra Daun Menggunakan Convolutional Neural Network

Dosen Pembimbing Utama : Dr. Andi Sunyoto, M.Kom
Dosen Pembimbing Pendamping : Tonny Hidayat, 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 Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 6 Desember 2023

Yang Menyatakan,

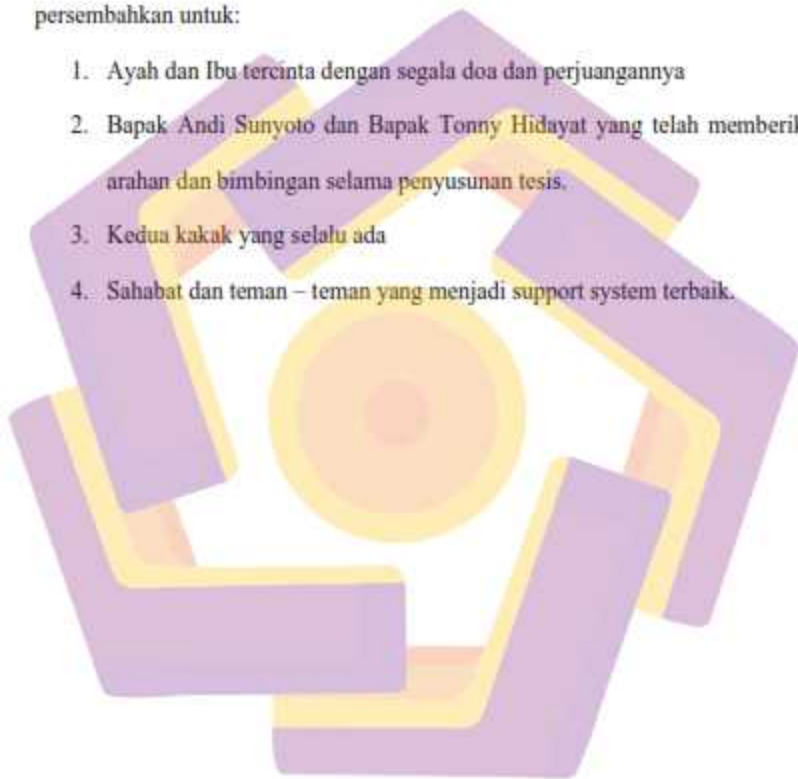


Liana Trihardianingsih

HALAMAN PERSEMBAHAN

Alhamdulillah, segala puji bagi Allah SWT yang telah melimpahkan rahmat serta hidayahNya sehingga tesisi ini dapat terselesaikan. Hasil ini saya persembahkan untuk:

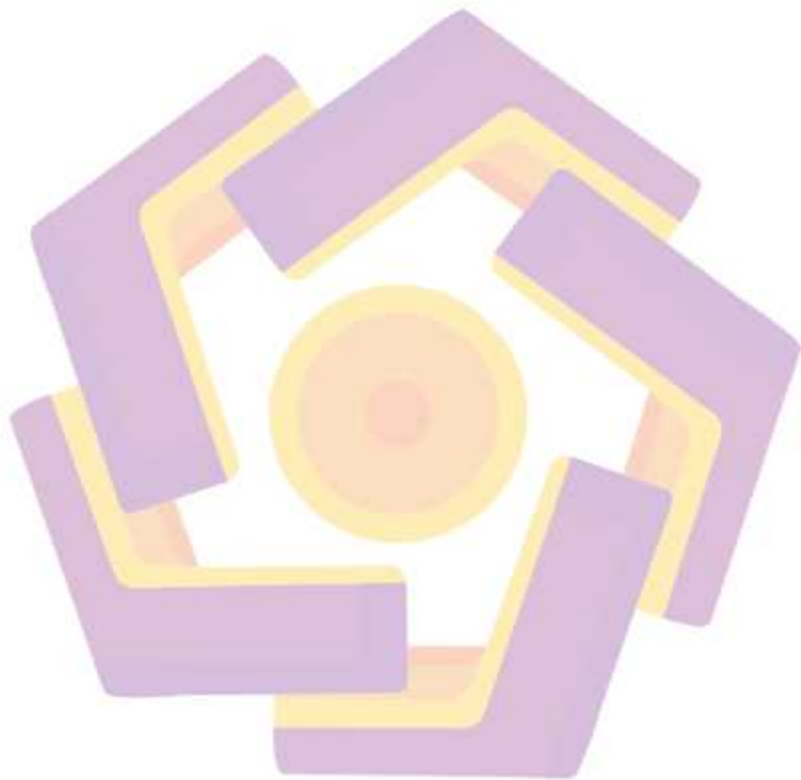
1. Ayah dan Ibu tercinta dengan segala doa dan perjuangannya
2. Bapak Andi Sunyoto dan Bapak Tonny Hidayat yang telah memberikan arahan dan bimbingan selama penyusunan tesis.
3. Kedua kakak yang selalu ada
4. Sahabat dan teman – teman yang menjadi support system terbaik.



HALAMAN MOTTO

"Spread your wings. Even if you break and get hurt again. You are stronger than anyone."

-Hello Future, NCT Dream



KATA PENGANTAR

Puji syukur dipanjatkan kehadiran Allah ta'ala yang telah melimpahkan segala kebutuhan yang diperlukan selama penyusunan tesis ini, sehingga penulis dapat menyelesaikan penyusunan tesis dengan judul "Klasifikasi Penyakit Tanaman Teh Berdasarkan Citra Daun Menggunakan Convolutional Neural Network".

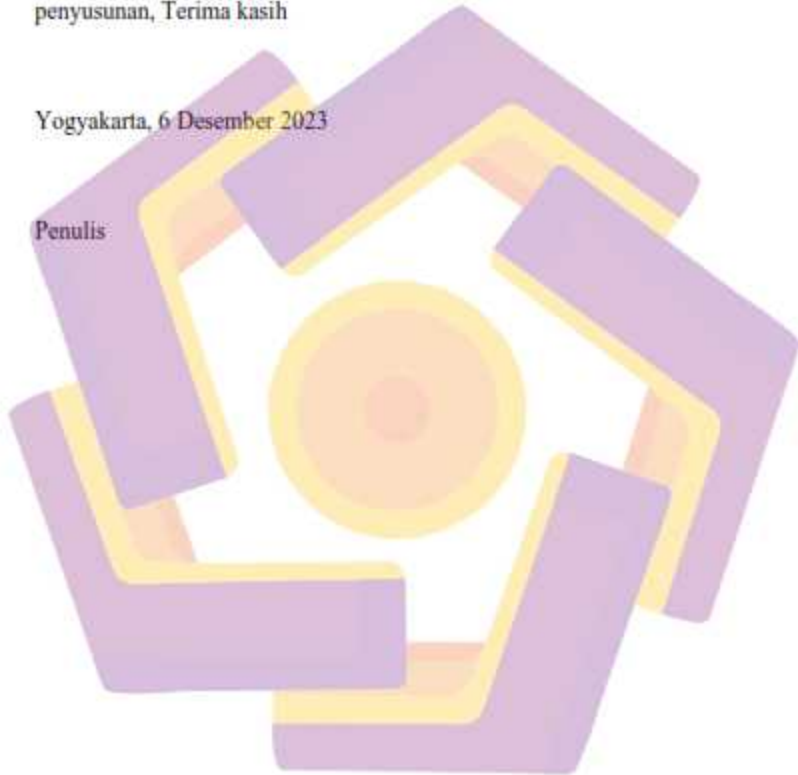
Bimbingan dan bantuan berharga dari berbagai pihak tidak terlepas dari penyusunan tesis ini. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. M. Suyanto, MM., selaku Rektor Universitas Amikom Yogyakarta,
2. Bapak Dr. Andi Sunyoto, M.Kom., sebagai pembimbing utama yang telah mendampingi dalam penyusunan tesis ini.
3. Pak Tonny Hidayat, M.Kom., Ph.D sebagai pendamping yang tak lelah membimbing penulis dalam pengerjaan tesis.
4. Bapak/ibu dosen Magister Teknik Informatika Universitas Amikom Yogyakarta yang telah memberikan ilmunya kepada penulis selama menempuh studi Magister Teknik Informatika
5. Ayah, ibu dan kedua kakak yang memberikan dukungan serta doa.
6. Teman – teman MTI-2021 kelas A Universitas Amikom Yogyakarta yang telah memberikan dukungan dan doa.
7. Terakhir, kepada semua pihak yang telah membantu yang tidak dapat penulis sampaikan satu-persatu.

Penulis berharap penelitian ini dapat membantu bagi pihak yang membutuhkan. Kekurangan dan ketidaksempurnaan masih dapat ditemukan agar dapat menjadi patokan kearah yang lebih baik lagi dimasa yang akan datang. Selibhnya permohonan maaf apabila terdapat salah kata dan salah dalam penyusunan, Terima kasih

Yogyakarta, 6 Desember 2023

Penulis



DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TESIS.....	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xv
INTISARI.....	xx
<i>ABSTRACT</i>	xxi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	7
1.3. Batasan Masalah.....	8
1.4. Tujuan Penelitian.....	9
1.5. Manfaat Penelitian.....	9
BAB II TINJAUAN PUSTAKA.....	10
2.1. Tinjauan Pustaka.....	10
2.2. Keaslian Penelitian.....	14

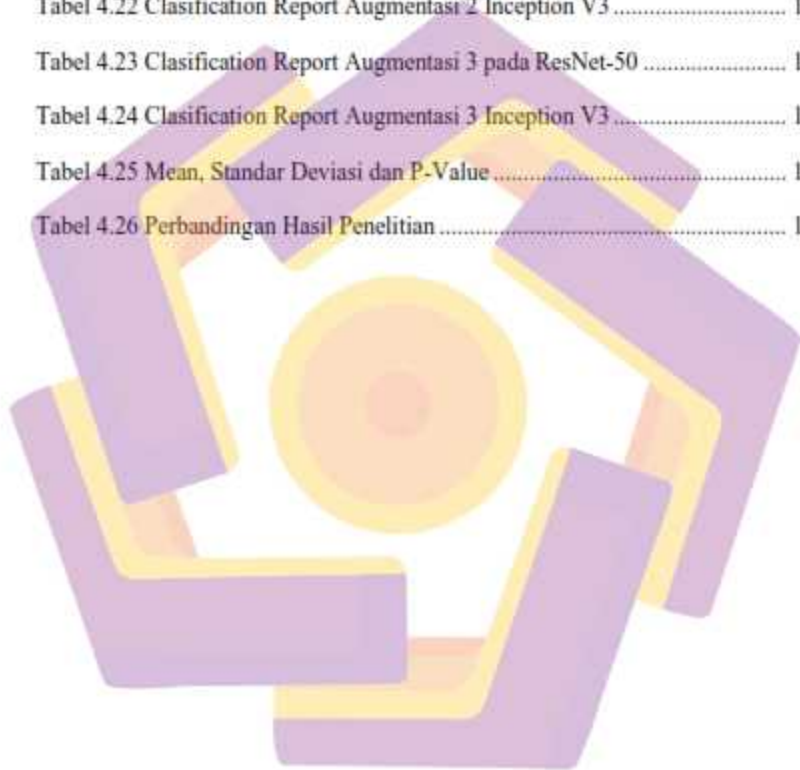
2.3. Landasan Teori.....	18
2.3.1 Klasifikasi.....	18
2.3.2 Deep Learning.....	18
2.3.3 Convolutional Neural Network.....	19
2.3.4 <i>Layer Convolutional</i>	21
2.3.5 Activation Function.....	21
2.3.6 Pooling Layer.....	22
2.3.7 Fully Connected Layer.....	22
2.3.8 Data Augmentasi.....	23
2.3.9 Transfer Learning.....	27
2.3.10 Inception V3.....	28
2.3.11 ResNet-50.....	29
2.3.12 Optimizer.....	31
2.3.13 Learning Rate.....	31
2.3.14 Batch Size.....	31
2.3.15 <i>Confusion Matrix</i>	32
2.3.16 Grad-CAM.....	34
2.3.17 Mean, Standar Deviasi, dan P-Value.....	35
BAB III METODE PENELITIAN.....	37
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	37

3.2. Metode Pengumpulan Data.....	37
3.3. Metode Analisis Data.....	39
3.4. Alur Penelitian.....	40
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	46
4.1. Pengumpulan Data.....	46
4.2. Preprocessing Data.....	46
4.3. Model Klasifikasi.....	48
4.4. Skenario Percobaan.....	49
4.5. Hasil dan Evaluasi Model.....	53
4.5.1 Hasil Pelatihan Model.....	53
4.5.2 Hasil Pengujian Model.....	58
4.5.3 Analisis Kesalahan Hasil <i>Confusion Matrix</i>	120
4.5.4 Faktor Yang Mempengaruhi Akurasi.....	140
4.6. Kondisi Terbaik Kedua Model.....	143
4.7. Perbandingan dengan Penelitian Sebelumnya.....	144
BAB V PENUTUP.....	146
5.1. Kesimpulan.....	146
5.2. Saran.....	147
DAFTAR PUSTAKA.....	148

DAFTAR TABEL

Tabel 2.1 Matrix Literature Review dan Posisi Penelitian	14
Tabel 2.2 Detail Arsitektur Inception V3.....	29
Tabel 2.3 Visualisasi Confusion Matrix.....	32
Tabel 3.1 Detail Dataset.....	38
Tabel 4.1 Detail Parameter Augmentasi	48
Tabel 4.2 Detail Skenario Percobaan.....	50
Tabel 4.3 Perbedaan Jumlah Data Training Skenario 1	50
Tabel 4.4 Perbandingan Jumlah Data Training Skenario 3.....	52
Tabel 4.5 Clasification Report Tanpa Augmentasi ResNet-50.....	62
Tabel 4.6 Clasification Report Tanpa Augmentasi Inception V3.....	63
Tabel 4.7 Clasification Report dengan Augmentasi ResNet-50	66
Tabel 4.8 Clasification Report dengan Augmentasi Inception V3	68
Tabel 4.9 Clasification Report <i>Optimizer</i> SGD pada ResNet-50.....	74
Tabel 4.10 Clasification Report <i>Optimizer</i> SGD Inception V3	76
Tabel 4.11 Clasification Report <i>Learning Rate</i> 0,0005 pada ResNet-50.....	82
Tabel 4.12 Clasification Report <i>Learning Rate</i> 0,0005 Inception V3	84
Tabel 4.13 Clasification Report <i>Learning Rate</i> 0,00075 pada ResNet-50.....	87
Tabel 4.14 Clasification Report <i>Learning Rate</i> 0,00075 Inception V3	88
Tabel 4.15 Clasification Report <i>Learning Rate</i> 0,0001 ResNet-50	91
Tabel 4.16 Clasification Report <i>Learning Rate</i> 0,0001 Inception V3	93
Tabel 4.17 Clasification Report <i>Batch Size</i> 16 pada ResNet-50.....	98

Tabel 4.18 Clasification Report <i>Batch size</i> 16 Inception V3	100
Tabel 4.19 Clasification Report <i>Batch size</i> 64 pada ResNet-50	103
Tabel 4.20 Clasification Report <i>Batch size</i> 64 Inception V3	105
Tabel 4.21 Clasification Report Augmentasi 2 pada ResNet-50	111
Tabel 4.22 Clasification Report Augmentasi 2 Inception V3	113
Tabel 4.23 Clasification Report Augmentasi 3 pada ResNet-50	116
Tabel 4.24 Clasification Report Augmentasi 3 Inception V3	117
Tabel 4.25 Mean, Standar Deviasi dan P-Value	141
Tabel 4.26 Perbandingan Hasil Penelitian	145



DAFTAR GAMBAR

Gambar 2.1 Cara Kerja Deep Learning.....	19
Gambar 2.2 Cara Kerja CNN.....	20
Gambar 2.3 Proses Layer Convolution.....	21
Gambar 2.4 Contoh Pooling Layer.....	22
Gambar 2.5 Proses Fully Connected Layer.....	23
Gambar 2.6 Teknik Augmentasi Data Rotasi.....	24
Gambar 2.7 Teknik Augmentasi Data Zoom.....	24
Gambar 2.8 Teknik Augmentasi Data Horizontal Flip.....	25
Gambar 2.9 Teknik Augmentasi Data Vertical Flip.....	25
Gambar 2.10 Teknik Augmentasi Data Shear.....	26
Gambar 2.11 Teknik Augmentasi Data Widht Shift.....	26
Gambar 2.12 Teknik Augmentasi Data Height Shift.....	27
Gambar 2.13 Arsitektur Inception V3.....	29
Gambar 2.14 Residual Bloks.....	30
Gambar 2.15 ArsitektulArsitektur ResNet-50.....	30
Gambar 3. 1 Alur penelitian.....	40
Gambar 4.1 Proses Load Image.....	47
Gambar 4.2 Custom Layer.....	49
Gambar 4.3 Grafik Rata-Rata Akurasi Training dan Validasi ResNet-50.....	53
Gambar 4.4 Grafik Rata Rata Nilai Loss Training dan Validasi ResNet-50.....	55
Gambar 4.5 Grafik Rata-Rata Akurasi Training dan Validasi Inception V3.....	56

Gambar 4.6 Grafik Rata-Rata Loss Training dan Validasi Inception V3	57
Gambar 4.7 <i>Confusion Matrix</i> Tanpa Augmentasi pada ResNet-50	60
Gambar 4.8 <i>Confusion Matrix</i> Tanpa Augmentasi pada Inception V3.....	60
Gambar 4.9 <i>Confusion Matrix</i> dengan Augmentasi pada ResNet-50	64
Gambar 4.10 <i>Confusion Matrix</i> dengan Augmentasi pada Inception V3	65
Gambar 4.11 Rata Rata Pengujian Model Skenario 1.....	69
Gambar 4.12 Perbandingan Rata-Rata Waktu dan Memori Skenario 1	70
Gambar 4.13 <i>Confusion Matrix Optimizer</i> SGD pada ResNet-50.....	72
Gambar 4.14 <i>Confusion Matrix Optimizer</i> SGD pada Inception V3.....	73
Gambar 4.15 Rata Rata Pengujian Model Skenario 2 <i>Optimizer</i>	77
Gambar 4.16 Perbandingan Rata-Rata Waktu dan Memori Skenario 2 <i>Optimizer</i>	78
Gambar 4.17 <i>Confusion Matrix Learning Rate</i> 0,0005 pada ResNet-50	80
Gambar 4.18 <i>Confusion Matrix Learning Rate</i> 0,0005 pada Inception V3	81
Gambar 4.19 <i>Confusion Matrix Learning Rate</i> 0,00075 pada ResNet-50.....	85
Gambar 4.20 <i>Confusion Matrix Learning Rate</i> 0,00075 pada Inception V3	85
Gambar 4.21 <i>Confusion Matrix Learning Rate</i> 0,0001 pada ResNet-50	89
Gambar 4.22 <i>Confusion Matrix Learning Rate</i> 0,0001 pada Inception V3	90
Gambar 4.23 Rata Rata Pengujian Model Skenario 2 <i>Learning Rate</i>	94
Gambar 4.24 Perbandingan Rata-Rata Waktu dan Memori Skenario 2 <i>Learning</i> <i>Rate</i>	95
Gambar 4.25 <i>Confusion Matrix Batch Size</i> 16 pada ResNet-50	97
Gambar 4.26 <i>Confusion Matrix Batch Size</i> 16 pada Inception V3	97

Gambar 4.27 <i>Confusion Matrix Batch size 64</i> pada ResNet-50.....	101
Gambar 4.28 <i>Confusion Matrix Batch size 64</i> pada Inception V3.....	102
Gambar 4.29 Rata Rata Pengujian Model Skenario 2 <i>Batch size</i>	106
Gambar 4.30 Perbandingan Rata-Rata Waktu dan Memori Skenario 2 <i>Batch size</i>	107
Gambar 4.31 <i>Confusion Matrix</i> Augmentasi 2 pada ResNet-50.....	109
Gambar 4.32 <i>Confusion Matrix</i> Augmentasi 2 pada Inception V3.....	110
Gambar 4.33 <i>Confusion Matrix</i> Augmentasi 3 pada ResNet-50.....	114
Gambar 4.34 <i>Confusion Matrix</i> Augmentasi 3 pada Inception V3.....	114
Gambar 4.35 Rata-Rata Akurasi Model Skenario 3.....	118
Gambar 4.36 Perbandingan Rata-Rata Waktu dan Memori Skenario 3	119
Gambar 4.37 Rata-Rata <i>Confusion Matrix</i> ResNet50	121
Gambar 4.38 Perbedaan Penyakit <i>Brown Blight(a)</i> dengan Penyakit <i>Gray Blight(b)</i>	122
Gambar 4.39 Perbedaan Penyakit <i>Gray Blight(a)</i> , <i>Algal Spot(b)</i> dan <i>Brown Blight(c)</i>	122
Gambar 4.40 Perbedaan Penyakit <i>Healthy(a)</i> , <i>Helopeltis(b)</i> dan <i>Red Spot(c)</i> ...	123
Gambar 4.41 Perbedaan Penyakit <i>Helopeltis(a)</i> , <i>Healthy(b)</i> dan <i>Gray Blight(c)</i>	124
Gambar 4.42 Perbedaan Penyakit <i>Red Spot(a)</i> , <i>Brown Blight(b)</i> , <i>Gray Blight(c)</i> dan <i>Healthy(d)</i>	124
Gambar 4.43 Rata-Rata <i>Confusion Matrix</i> ResNet50	126
Gambar 4.44 Perbedaan Penyakit <i>Algal Spot(a)</i> , dan penyakit <i>Gray Blight(b)</i> ..	126

Gambar 4.45 Perbedaan Penyakit <i>Brown Blight</i> (a), <i>Gray Blight</i> (b), dan <i>Helopeltis</i> (c)	127
Gambar 4.46 Perbedaan Penyakit <i>Gray Blight</i> (a), <i>Algal Spot</i> (b) dan <i>Brown Blight</i> (c)	128
Gambar 4.47 Perbedaan Penyakit <i>Helopeltis</i> (a), <i>Healthy</i> (b) dan <i>Gray Blight</i> (c).....	129
Gambar 4.48 Perbedaan Penyakit <i>Red Spot</i> (a), <i>Brown Blight</i> (b), dan <i>Healthy</i> (c)	130
Gambar 4.49 Kesalahan Inception V3 Memprediksi Kelas <i>Algal spot</i> Sebagai <i>Gray blight</i>	131
Gambar 4.50 Kesalahan ResNet-50 Memprediksi <i>Brown blight</i> Sebagai <i>Gray blight</i>	132
Gambar 4.51 Kesalahan Inception V3 Memprediksi <i>Brown blight</i> Sebagai <i>Gray blight</i>	132
Gambar 4.52 Kesalahan Inception V3 Memprediksi Kelas <i>Brown blight</i> Sebagai <i>Helopeltis</i>	133
Gambar 4.53 ResNet-50 Memprediksi Kelas <i>Brown blight</i> dengan Benar	134
Gambar 4.54 Kesalahan ResNet-50 Memprediksi Kelas <i>Gray blight</i> Sebagai <i>Helopeltis</i>	135
Gambar 4.55 ResNet-50 Memprediksi Kelas <i>Gray blight</i> dengan Benar.....	135
Gambar 4.56 Kesalahan ResNet-50 Memprediksi Kelas <i>Healthy</i> Sebagai <i>Helopeltis</i>	136
Gambar 4.57 Kesalahan Inception V3 Memprediksi Kelas <i>Healthy</i> Sebagai <i>Helopeltis</i>	136

Gambar 4.59 Kesalahan Inception V3 Memprediksi <i>Helopeltis</i> Sebagai <i>Gray blight</i>	137
Gambar 4.60 Kesalahan ResNet-50 Memprediksi Kelas <i>Red spot</i> Sebagai <i>Brown blight</i>	138
Gambar 4.61 Kesalahan ResNet-50 Memprediksi Kelas <i>Red spot</i> Sebagai <i>Healthy</i>	138
Gambar 4.62 Kesalahan Inception V3 Memprediksi <i>Red spot</i> Sebagai <i>Gray blight</i>	139
Gambar 4.63 ResNet-50 Memprediksi Kelas <i>Red Spot</i> dengan Benar	140
Gambar 4.64 Kondisi Terbaik Kedua Model	144



INTISARI

Kemajuan teknologi telah memberikan kontribusi besar dalam pengendalian penyakit tanaman. Salah satu metode untuk menyelesaikan masalah identifikasi penyakit tanaman adalah penggunaan deep learning untuk pengolahan citra digital. Penyakit daun teh merupakan penyakit tanaman yang memerlukan pengendalian cepat dan efektif. Penelitian ini mengadopsi arsitektur Convolutional Neural Network (CNN) yaitu ResNet-50 dan Inception V3 untuk mengklasifikasikan enam jenis penyakit yang menyerang daun teh. Untuk mengoptimalkan performa ResNet-50 dan Inception V3, teknik augmentasi akan ditambahkan pada proses training agar menambah variasi data serta mengatur hyperparameter untuk meningkatkan akurasi

Penelitian ini memiliki empat skenario untuk mendapatkan hasil yang optimal. Tiga dari empat skenario yang diusulkan adalah dengan melakukan penyesuaian hyperparameter yang meliputi *optimizer*, *learning rate*, dan *batch size*. Penelitian ini kemudian melakukan klasifikasi dengan menggunakan 3 kombinasi penggunaan augmentasi data yaitu zoom, rotasi, flip; zoom, rotasi, shear; dan zoom, rotasi, shift. Penyesuaian hyperparameter dan penggunaan augmentasi data yang tepat dapat meningkatkan performa model ResNet-50 dan Inception V3.

Hasil akhir dari penelitian ini adalah kedua model klasifikasi, ResNet-50 dan Inception V3 memiliki performa yang baik dalam melakukan klasifikasi penyakit daun teh. ResNet-50 memperoleh rata-rata akurasi 98,30% dengan menggunakan *optimizer* Adam, *learning rate* 0,0001, *batch size* 16 dan augmentasi berupa rotasi, zoom, dan flip. Sedangkan Inception V3 memperoleh rata-rata akurasi 97,61% dengan menggunakan *optimizer* Adam, *learning rate* 0,0001, *batch size* 64, dan augmentasi berupa rotasi, zoom, dan shear.

Kata kunci: Augmentasi, Hyperparameter, Inception V3, Penyakit Daun Teh, ResNet-50

ABSTRACT

Technological advances have made a major contribution to controlling plant diseases. One method to solve the problem of identifying plant diseases is the use of deep learning for digital image processing. Tea leaf disease is a plant disease that requires fast and effective control. This research adopts a Convolutional Neural Network (CNN) architecture, namely ResNet-50 and Inception V3 to classify six types of diseases that attack leaves. To optimize the performance of ResNet-50 and Inception V3, augmentation techniques will be added to the training process to increase data variation and adjust hyperparameters to increase accuracy

This research has four scenarios to obtain optimal results. Three of the four proposed scenarios involve adjusting hyperparameters which include optimizer, learning rate, and batch size. This research then carried out classification using 3 combinations of data augmentation, namely zoom, rotation, flip; zoom, rotation, shear; and zoom, rotation, shift. Hyperparameter tuning and appropriate use of data augmentation can improve the performance of ResNet-50 and Inception V3 models.

The final result of this research is that the two classification models, ResNet-50 and Inception V3 have good performance in classifying tea leaf diseases. ResNet-50 obtained an average accuracy of 98.30% using the Adam optimizer, learning rate 0.0001, batch size 16 and augmentation in the form of rotation, zoom and flip. Meanwhile, Inception V3 obtained an average accuracy of 97.61% using the Adam optimizer, learning rate 0.0001, batch size 64, and augmentation in the form of rotation, zoom, and shear.

Keywords: *Augmentation, Hyperparameters, Inception V3, ResNet-50, Tea Leaf Disease*

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Sektor pertanian memiliki peran yang penting dalam menunjang perekonomian sebuah negara. Salah satunya adalah sektor perkebunan teh. Perkebunan teh menjadi salah satu sektor industri yang berkembang pesat karena selalu terjadi peningkatan permintaan produksi setiap tahunnya (Pandian et al., 2023). Teh merupakan minuman yang populer di seluruh dunia karena rasanya yang enak, cita rasanya yang istimewa, dan mengandung berbagai manfaat untuk kesehatan (Soeb et al., 2023). Sehingga tidak aneh jika teh telah menjadi minuman yang paling banyak dikonsumsi setelah air (Sanlier et al., 2018).

Menurut penelitian yang dilakukan oleh (Ahmed et al., 2021), konsumsi teh global diperkirakan akan mencapai 7,4 juta metrik ton pada tahun 2025, dari sebelumnya sekitar 7,3 juta metrik ton di tahun 2020. Sehingga dapat diprediksi bahwa permintaan dan persaingan pasar dunia terhadap produksi teh akan segera meningkat dalam kurun waktu yang cukup singkat. Namun, permintaan yang dinilai akan meningkat tersebut tidak berbanding lurus dengan produksi teh yang terjadi saat ini. Kondisi cuaca dan perubahan iklim membuat adanya penurunan produksi teh (Soeb et al., 2023). Selain fenomena global tersebut, berbagai penyakit dan hama memang sering menyerang tanaman teh selama proses pertumbuhan.

Daun teh yang bermutu adalah daun teh yang memiliki tingkat kematangan pucuk daun yang tepat. Penentuan kematangan pucuk pada daun teh diperoleh

berdasarkan perhitungan dari periode pemetikan yang dilakukan sebelumnya. Periode pemetikan ini bergantung pada umur, kecepatan pembentukan tunas, ketinggian, iklim dan kesehatan tanaman (Ibrahim et al., 2022). Faktor kesehatan pada tanaman membuat tidak semua daun teh bisa dipanen dengan baik karena ada beberapa penyakit daun dan hama yang bisa menyerang pertumbuhan tanaman teh itu sendiri (Hardi, 2022; Jahagirdar & Budihal, 2020).

Penyakit yang menjangkit daun teh selalu menjadi perhatian besar karena secara langsung berdampak pada kualitas dan kuantitas produk ketika memasuki masa panen (Pandian et al., 2023). Penelitian yang dilakukan (Hu et al., 2019) mengungkapkan bahwa terdapat lebih dari seratus penyakit yang telah teridentifikasi merusak daun teh di seluruh dunia. Sedangkan sejauh ini para ahli masih mengidentifikasi beberapa gejala penyakit yang menyerang daun teh dengan mengamati visual daun yang meliputi ukuran fisik, tekstur daun, struktur tulang, dan warna dari daun itu sendiri (Ramdan et al., 2019). Hal ini sangat sulit untuk dilakukan sendiri oleh petani teh, sedangkan untuk menyediakan tenaga ahli yang cukup untuk menangani seluruh area perkebunan sangatlah tidak mungkin karena akan membutuhkan biaya yang sangat mahal. Sehingga petani hanya mampu mengandalkan pengalaman yang dimiliki untuk mendiagnosa penyakit yang menyerang daun teh, hasilnya pun menjadi sangat subjektif (Bao et al., 2022). Padahalantisipasi yang tidak tepat dalam mendeteksi penyakit dengan penggunaan pestisida atau bahan kimia lain yang tidak tepat akan menyebabkan patogen semakin kuat dan kekebalan tanaman teh untuk melawan penyakit akan menurun

(Gayathri et al., 2020). Oleh karena itu, diperlukan diagnosis penyakit pada daun teh yang lebih tepat dan andal (Mathew & Mahesh, 2022).

Penanganan masalah penyakit daun teh untuk mengurangi dampak kerusakan dan kegagalan panen telah dilakukan beberapa kali dalam sebuah penelitian. Salah satu penelitian yang telah dilakukan adalah pemanfaatan bidang informatika untuk mengidentifikasi penyakit yang ada pada daun teh dengan menggunakan data gambar daun. Identifikasi dengan menggunakan citra ini merupakan proses image processing atau pengolahan citra digital (Suttapakti & Bupeng, 2019). Image processing dapat membantu para petani teh mengidentifikasi tanda-tanda awal penyakit sehingga bisa segera dilakukan penanganan yang efektif dan efisien untuk menghindari penyebaran penyakit pada daun teh.

Minat yang berkembang terhadap perkembangan teknologi menggunakan kecerdasan buatan, khususnya computer vision, untuk mengenali suatu objek perlahan mulai meningkat dan secara bertahap diterapkan baik untuk pengenalan objek. Terdapat beberapa algoritma yang dapat digunakan digunakan untuk pengenalan dan pemrosesan gambar. Salah algoritma yang sering digunakan adalah *Convolutional Neural Network (CNN)*. CNN merupakan algoritma deep learning yang terinspirasi oleh struktur dan fungsi jaringan saraf organik di otak yang dapat mengambil gambar, termasuk klasifikasi gambar, segmentasi, lokalisasi, dan tugas lain untuk membedakan objek satu sama lain (Sarvamangala & Kulkarni, 2022). CNN memiliki fitur utama, yaitu kombinasi lapisan konvolusional, ekstraksi bentuk visual, perceptron multi-lapisan, dan implementasi deteksi konvolusional.

Berkembangnya teknologi komputasi, *machine learning*, dan *image processing* (Suttapakti & Bunpeng, 2019) dapat digunakan untuk mendeteksi dan mengidentifikasi penyakit tanaman secara otomatis. Hal ini tentu saja akan memainkan peran penting untuk diagnosis otomatis penyakit tanaman (Zhao et al., 2016). Minat yang berkembang terhadap perkembangan teknologi menggunakan kecerdasan buatan, khususnya *computer vision*, untuk mengenali suatu objek perlahan mulai meningkat dan secara bertahap diterapkan baik untuk pengenalan objek. *Computer vision* terbukti mampu digunakan untuk mendiagnosis penyakit atau hama pada tanaman secara akurat berdasarkan gejala karakteristik yang ada (Pandian et al., 2022). Pendekatan tersebut dilakukan dengan mengandalkan ekstraksi fitur dari citra daun dan identifikasi serta klasifikasinya menggunakan jaringan syaraf tiruan (ChandraKarmokar et al., 2015).

Terdapat beberapa algoritma yang dapat digunakan digunakan untuk pengenalan dan pemrosesan gambar. Salah algoritma yang sering digunakan adalah *Convolutional Neural Network (CNN)*. CNN merupakan algoritma deep learning yang terinspirasi oleh struktur dan fungsi jaringan saraf organik di otak yang dapat mengambil gambar, termasuk klasifikasi gambar, segmentasi, lokalisasi, dan tugas lain untuk membedakan objek satu sama lain (Sarvamangala & Kulkarni, 2022). CNN memiliki fitur utama, yaitu kombinasi lapisan konvolusional, ekstraksi bentuk visual, perceptron multi-lapisan, dan implementasi deteksi konvolusional.

Perkembangan penelitian di bidang teknologi mengenai pengolahan citra digital saat ini telah banyak dilakukan. Salah satu penelitian yang telah

memanfaatkan citra digital adalah penelitian yang dilakukan (Chen et al., 2019), penelitian ini menggunakan salah satu model Convolutional Neural Network (CNN), yaitu LeafNet, dan melakukan perbandingan dengan dua model lain yaitu Support Vector Machine (SVM) dan Multi-Layer Perceptron (MLP) untuk klasifikasi dan identifikasi jenis penyakit pada teh dengan menggunakan citra daun yang terdiri dari 7 kelas. Pada ketiga metode tersebut dilakukan evaluasi secara individual dari setiap kelas dan diperoleh rata-rata hasil akurasi klasifikasi sebesar 90,19% untuk model LeafNet, 60,62% untuk SVM, dan 70,77% untuk MLP. Penelitian lain dilakukan oleh (Hardi, 2022), menggunakan Mobile Net dan NasNet mobile untuk melakukan klasifikasi penyakit daun teh. Hasil komparasi antara kedua metode mendapatkan hasil akurasi sebesar 88% untuk NasNet dan 95% pada MobileNet.

Penelitian (Yuwana et al., 2020) ini memberikan pendekatan berbasis deep learning untuk klasifikasi citra daun teh sebagai daun sehat, *blister blight*, *empoaasca sp.* dan *looper caterpillar*. Penelitian ini menguji performa pada data augmentasi yang dievaluasi menggunakan berbagai Deep Convolutional Neural Network (DCNN). Penelitian ini menerapkan augmentasi dilakukan pada seluruh kelas untuk mendapatkan data yang lebih bervariasi karena dataset yang digunakan sedikit. Hasil penelitian diperoleh akurasi tertinggi sebesar 88,86% dengan menggunakan teknik augmentasi GAN. Penelitian lain yang dilakukan oleh (Gayathri et al., 2020) melakukan klasifikasi penyakit tanaman teh pada 4 kelas menggunakan CNN yaitu model LeNet dan menggabungkan dengan metode Color Co-occurrence untuk mengekstraksi package warna. Penelitian menggunakan augmentasi data untuk

memperbanyak variasi data pada seluruh kelas yang ada. Dengan menggunakan fungsi aktivasi softmax, diperoleh akurasi sebesar 90,23% dan MCA sebesar 90,16%.

Klasifikasi citra menggunakan CNN terbukti memiliki performa yang baik untuk melakukan klasifikasi penyakit daun teh. Namun, untuk melatih sebuah sistem pembelajaran mendalam membutuhkan data dalam jumlah besar (Ramdan et al., 2020), sedangkan untuk mengumpulkan data dalam jumlah banyak akan memakan banyak waktu dan biaya (Yuwana et al., 2020). Sehingga, keterbatasan data akan menjadi kondisi yang tidak bisa dihindari. Salah satu cara yang dapat digunakan untuk menambah keragaman gambar tanpa memakan banyak biaya dan waktu adalah menerapkan teknik augmentasi. Penggunaan augmentasi data dapat membantu untuk menambah keberagaman gambar dan membuat model dapat bekerja lebih baik pada proses training. Namun, penggunaan augmentasi yang terlalu berlebihan juga dapat menyebabkan terjadinya penurunan akurasi karena gambar akan terlalu banyak mengalami perubahan dari citra asli sehingga dapat menyebabkan terjadi *overfitting* (Wu et al., 2017). Sehingga perlu diperhatikan penggunaan teknik augmentasi yang tepat untuk klasifikasi daun teh.

Selain penggunaan teknik augmentasi, performa CNN juga dapat ditingkatkan dengan mengatur hyperparameter yang digunakan (Julianto et al., 2022). Untuk melatih CNN dengan agar dapat mengklasifikasikan gambar, dibutuhkan banyak hyperparameter yang perlu disesuaikan karena penggunaan hyperparameter ini akan mempengaruhi kinerja jaringan disepanjangnya waktu

untuk konvergensi (Kandel & Castelli, 2020a). Beberapa hyperparameter yang bisa dioptimalkan diantaranya adalah *batch size*, *learning rate* dan *optimizer*.

Berdasarkan uraian diatas, peneliti tertarik untuk melakukan penelitian terkait klasifikasi penyakit pada daun teh menggunakan Inception-V3 dan Resnet-50. Identifikasi penyakit daun teh akan dilakukan pada enam kelas, yaitu *Normal leaf*, *Algal Spot*, *Brown blight*, *Gray Blight*, *Helopeltis*, dan *Red spot*. Dengan menggunakan Inception-V3 dan Resnet-50, penelitian ini akan melakukan analisis terhadap penggunaan teknik augmentasi serta paramater yang digunakan seperti penggunaan *batch size*, *learning rate*, dan *optimizer* dalam meningkatkan kinerja model arsitektur yang diusulkan dengan memperhatikan akurasi yang dihasilkan, waktu komputasi dan penggunaan memori. Model yang dihasilkan dari penelitian ini diharapkan dapat membantu para petani dalam melakukan identifikasi penyakit tanaman teh sehingga petani teh dapat melakukan perawatan tanaman teh dengan tepat.

1.2. Rumusan Masalah

Berdasarkan pada permasalahan yang diuraikan pada latar belakang, maka rumusan masalah pada penelitian ini adalah sebagai berikut:

- a. Bagaimana pengaruh dari penerapan penyesuaian hyperparameter dan augmentasi terhadap akurasi dari model ResNet-50 dan Inception V3?
- b. Berapa tingkat akurasi yang dihasilkan dalam melakukan klasifikasi penyakit daun teh pada model arsitektur Resnet-50 dan Inception-V3?

- c. Berapa *time consumption per step* dan *memory size* yang dihabiskan pada proses klasifikasi penyakit daun teh menggunakan model arsitektur Inception-V3 dan Resnet-50?

1.3. Batasan Masalah

Batasan masalah pada penelitian ini adalah:

- a. Model arsitektur yang digunakan adalah Resnet-50 dan Inception-V3.
- b. Data yang digunakan pada model klasifikasi terbagi ke dalam 6 class, yaitu *normal leaf*, *algal spot*, *brown blight*, *gray blight*, *helopeltis*, dan *red spot*.
- c. Pengujian kinerja model yang diusulkan dievaluasi menggunakan *Confusion Matrix Multi-Class*.
- d. Indikator yang dipakai untuk membandingkan hasil performa adalah nilai *accuracy*, *recall*, dan *F1-Score* seperti pada penelitian yang dilakukan oleh (Datta & Gupta, 2023).
- e. Faktor-faktor yang dianalisis adalah penggunaan *optimizer*, *learning rate*, *batch size*, dan *augmentasi*.
- f. Platform yang digunakan untuk melakukan pelatihan, pengujian dan kebutuhan visualisasi data adalah Google Collab.
- g. Library yang digunakan adalah library keras, tensorflow, dan scikit-learn. Library keras dan tensorflow digunakan untuk membangun dan melatih model arsitektur yang dipilih, sedangkan library scikit-learn digunakan untuk evaluasi model.

1.4. Tujuan Penelitian

Tujuan penelitian sangat penting untuk mengukur keberhasilan suatu penelitian. Tujuan penelitian ini adalah sebagai berikut:

- a. Mengetahui sejauh mana penyesuaian hyperparameter dan teknik augmentasi yang digunakan terhadap akurasi dari model ResNet-50 dan Inception V3.
- b. Mengetahui hasil performa dari model Inception-V3 dan Resnet-50 yang menghasilkan akurasi paling tinggi dengan menerapkan data augmentation.
- c. Mengetahui time consumption dengan menggunakan indikator durasi waktu dalam proses melakukan klasifikasi penyakit pada daun teh.

1.5. Manfaat Penelitian

Penelitian yang baik merupakan hasil penelitian yang memiliki manfaat dan dedikasi kepada ilmu pengetahuan untuk selanjutnya bisa diimplementasikan secara langsung di lapangan. Berdasarkan tujuan yang ingin dicapai pada uraian diatas, penelitian ini memberikan manfaat sebagai berikut:

- a. Mampu melakukan analisis dan klasifikasi mengenai penyakit daun teh dengan menerapkan data augmentation.
- b. Mengetahui pengaruh peningkatan dari model Inception-V3 dan Resnet-50 terhadap analisis klasifikasi penyakit daun teh menggunakan CNN.
- c. Menjadi referensi terhadap penelitian di masa mendatang yang berkaitan dengan pemanfaatan model Inception-V3 dan Resnet-50 pada citra daun teh dengan menggunakan CNN.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Sebelum melakukan penelitian terkait identifikasi penyakit pada daun teh, peneliti terlebih dahulu melakukan peninjauan pustaka terkait penelitian-penelitian yang telah dilakukan oleh para peneliti sebelumnya sebagai acuan dalam pengembangan penelitian yang akan dilakukan. Dari peninjauan pustaka yang dilakukan diperoleh beberapa kajian penelitian terdahulu yang berkaitan dengan penelitian yang akan dilakukan.

Penelitian yang dilakukan oleh (Datta & Gupta, 2023) menerapkan penggunaan Deep CNN untuk mengklasifikasikan penyakit pada daun teh ke dalam enam kelas yang terdiri dari lima jenis penyakit daun teh dan satu kelas untuk daun normal (sehat). Deep CNN digunakan karena memiliki banyak lapisan tersembunyi sehingga dapat dipertimbangkan untuk mengklasifikasikan daun teh yang sakit ke dalam kategori yang berbeda. Penggunaan Deep CNN membantu jaringan dalam mendeteksi lebih banyak fitur sehingga akurasi dalam deteksi penyakit akan meningkat. Berdasarkan hasil percobaan yang dilakukan, metode yang diusulkan mampu mendeteksi jenis penyakit daun teh dengan akurasi yang cukup tinggi yaitu 96,56%. Ketepatan kelas penyakit berikut ini adalah sebagai berikut. *Algal spot* memiliki akurasi sebesar 98,23%, *Brown blight* memiliki akurasi sebesar 97,98%, Grey Blight memiliki akurasi sebesar 93,46%, kelas Daun Sehat memiliki akurasi

sebesar 99,10%, kelas penyakit *Helopeltis* memiliki akurasi sebesar 98,98% dan *Red spot* memiliki akurasi 92%.

Penelitian lain yang dilakukan oleh (Hu et al., 2021) mengusulkan metode deep learning untuk meningkatkan kinerja deteksi dan analisis keparahan penyakit daun tea leaf blight. Daun tea leaf blight dideteksi menggunakan kerangka deep learning yaitu Faster R-CNN, untuk meningkatkan kinerja pendeteksian daun berpenyakit yang kabur, tersumbat, dan kecil. Daun tea leaf blight yang terdeteksi akan dimasukkan ke dalam jaringan VGG16 untuk dilakukan analisis. Pada tahap pengujian, diperoleh nilai recall sebesar 93.92%, precision 95.74%, dan AP value reaches 91.22%. Hasil eksperimen menunjukkan bahwa presisi rata-rata akurasi dari metode yang diusulkan masing-masing meningkat lebih dari 6% dan 9%, dibandingkan dengan metode pembelajaran mesin klasik.

Selanjutnya penelitian yang dilakukan oleh (Lee et al., 2020) melakukan klasifikasi terhadap penyakit daun teh serta hama yang menyerang. Penelitian ini menggunakan 1822 gambar teh yang disebabkan oleh tiga penyakit (*brown blight*, *blister blight*, and *algal leaf spot*) dan empat hama (*leaf miner*, *tea thrip*, *tea leaf roller*, and *tea mosquito bug*). Metode yang diusulkan pada penelitian ini adalah Faster R-CNN. Hasil penelitian menggunakan Faster R-CNN yang terlatih diperoleh nilai precision 77.5%, recall 70.6%, F1 score 73.91%, dan mean average precision sebesar 66,02%. Secara keseluruhan, akurasi yang diperoleh untuk identifikasi pada tujuh kelas penyakit dan hama teh adalah 89,4%.

Penelitian terkait klasifikasi penyakit daun teh juga dilakukan oleh (Pandian et al., 2023). Penelitian ini menggunakan Deep CNN untuk mengidentifikasi

penyakit hawar abu-abu pada daun teh. Deep CNN yang diusulkan terdiri dari tiga blok bottleneck, dua pasang lapisan convolutional (Conv), dan tiga dense layer. Teknik augmentasi juga digunakan untuk meningkatkan ukuran data set. Sebanyak 5016 data dilatih menggunakan Deep CNN yang meliputi gambar daun sehat, penyakit hawar abu-abu, dan penyakit lainnya. Performa dari metode yang digunakan menghasilkan akurasi 98.99%, precision 98.51%, recall 98.48%, F measure 98.49%, dan misclassification rates 1.01% pada data uji.

Penelitian oleh (Ramdan et al., 2020) menggunakan model pre-trained pada data ImageNet dan akan mengulang proses training untuk mendeteksi penyakit daun teh dari dataset yang dimiliki. Untuk mengurangi perbedaan tugas ImageNet dan dataset yang dimiliki, dilakukan penyempurnaan pada model pre-trained dengan mengganti beberapa bagian model pre-trained dengan struktur baru. Penelitian ini dilakukan dengan menerapkan transfer learning dengan fine-tuning untuk memproses data training. Model yang digunakan untuk proses training adalah VGGNet, ResNet, dan Xception. Hasil penelitian yang dilakukan diketahui bahwa penerapan fine-tuning pada model yang digunakan terbukti efektif dan diperoleh akurasi menggunakan ResNet-50 94.05%, VGG16 91.26%, dan Exception 91.71%.

Penelitian yang dilakukan oleh (Yuwana et al., 2020) menggunakan augmentasi untuk menambah jumlah data. Augmentasi data dilakukan pada empat kelas penyakit daun teh dengan menggunakan Generative Adversarial Networks (GAN) dan Deep Convolutional GAN (DCGAN). Performa deteksi penyakit teh pada data augmented dievaluasi menggunakan berbagai Deep Convolutional Neural Network (DCNN) termasuk AlexNet, DenseNet, ResNet, dan Xception.

Hasil percobaan menunjukkan bahwa akurasi GAN tertinggi diperoleh oleh arsitektur DenseNet yaitu sebesar 88,84%, akurasi baseline pada arsitektur yang sama sebesar 86,30%. Hasil akurasi DCGAN pada penggunaan arsitektur yang sama menunjukkan kecenderungan yang sama, yaitu sebesar 88,86%.

Penelitian lain yang dilakukan oleh (Hardi, 2022) melakukan klasifikasi penyakit daun teh dengan mengkomparasikan dua arsitektur yaitu algoritma MobileNet dan NasNet Mobile. Dataset yang digunakan pada penelitian ini terdiri dari daun teh yang sehat dan lima jenis penyakit daun teh yaitu *algal spot*, *brown blight*, *gray blight*, *helopeltis*, dan *red spot*. Dari hasil Komparasi antara metode MobilNet dan NasNet Mobile dalam klasifikasi penyakit daun teh, NasNet Mobile mendapatkan hasil akurasi sebesar 88% sedangkan MobileNet sebesar 95%.



2.2. Keaslian Penelitian

Tabel 2.1 Matrix Literature Review dan Posisi Penelitian

Klasifikasi Penyakit pada Tanaman Teh Berdasarkan Citra Daun Menggunakan Convolutional Neural Network

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	<i>A Novel Approach for the Neural Detection of Tea Leaf Disease Using Deep Network Deep Datta Neural Network.</i>	Saikat Datta dan Nitin Gupta, Procedia Computer Science, 2023	Melakukan klasifikasi dalam proses diagnosa penyakit daun teh menggunakan Convolutional Neural Networks (CNN)	Berdasarkan hasil yang diperoleh pada tahap pengujian, rata-rata tingkat akurasi yang diperoleh adalah 96.56% untuk klasifikasi 6 kelas penyakit daun teh	Mencoba menerapkan berbagai teknik augmentasi dan parameter yang lain.	Menggunakan beberapa teknik augmentasi untuk meningkatkan akurasi serta dilakukan beberapa skenario dengan mengubah <i>learning rate</i> , <i>batch size</i> , dan <i>optimizer</i>
2	<i>Detection and severity analysis of tea leaf blight based on deep learning</i>	Gensheng Hu, Huaiyu Wang, Yan Zhang, dan Mingzhu Wan Computers and Electrical Engineering 2021	Melakukan klasifikasi penyakit leaf blight pada daun teh menggunakan metode deep learning yaitu VGG16	Proses klasifikasi pada penelitian dilakukan pada dua kelas yaitu <i>healthy</i> dan leaf blight. Pada tahap pengujian, diperoleh nilai recall sebesar 93.92%, precision 95.74%, dan AP value reaches 91.22%	Mencoba menerapkan dataset yang lebih bervariasi dan dibandingkan dengan metode yang lain untuk mengukur sejauh mana performa VGG16 dibanding dengan metode yang lain	Melakukan klasifikasi terhadap penyakit daun teh menggunakan model Resnet-50 dan Inception-V3 dan menggunakan dataset yang terdiri dari 6 kelas, yaitu <i>normal leaf</i> , <i>algal spot</i> , <i>brown blight</i> , <i>gray blight</i> , <i>helopeltis</i> , dan <i>red spot</i> .
3	<i>Identification of tea foliar diseases and</i>	Sheng-Hung Lee, Shiou-Ruei	Melakukan klasifikasi terhadap	Proses klasifikasi dengan menggunakan	Jumlah sampel pelatihan harus	Menggunakan berbagai teknik augmentasi untuk

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	<i>pest damage under practical field conditions using a convolutional neural network</i>	Lin, dan Shih-Fang Chen Plant Pathology 2020	tiga penyakit dan 4 hama yang menyercang daun teh menggunakan metode deep learning yaitu Faster R-CNN	Faster R-CNN diperoleh akurasi sebesar 89,4% untuk identifikasi tujuh kelas penyakit dan hama teh	ditingkatkan lebih lanjut dan menambah kategori baru untuk sampel negatif untuk mengurangi kesalahan klasifikasi.	meningkatkan akurasi dan menerapkan 2 arsitektur CNN yaitu Resnet-50 dan Inception-v3.
4	<i>Grey Blight Disease Detection on Tea Leaves Using Improved Deep Convolutional Neural Network</i>	J. Arun Pandian, Sum Nirmala Nisha, K. Kanchanasudhi, Abhay K. Pandey, dan Samira Kabir Rima Computational Intelligence and Neuroscience 2023	Menggunakan DCNN untuk diagnosis otomatis penyakit hawar abu-abu pada tanaman teh serta meningkatkan nilai akurasi menggunakan teknik augmentasi data untuk menambah sampel pada proses training	Hasil penelitian menunjukkan bahwa algoritma yang diusulkan (DCNN) dapat mendeteksi jenis penyakit yang menetap pada daun teh dengan nilai accuracy sebesar 98,99% pada data uji	Walaupun memiliki tingkat akurasi yang tinggi, identifikasi hanya terfokus pada satu jenis penyakit yaitu penyakit Grey blight. Jenis penyakit lain dijadikan satu kelas dan hanya diberi label 'penyakit lain'	Melakukan klasifikasi terhadap penyakit daun teh menggunakan model Resnet-50 dan Inception-V3 dan menggunakan dataset yang terdiri dari 6 kelas, yaitu <i>normal leaf, algal spot, brown blight, gray blight, helopeltis</i> , dan <i>red spot</i>
5	<i>Transfer Learning and Fine-Tuning for Deep Learning-Based Tea Diseases Detection on Small Datasets.</i>	Ade Ramdan, Ana Heryana, Andria Arisal, R. Budiarianto S. kusumo, dan Hilman F Pardede.	Melakukan klasifikasi penyakit daun teh menggunakan beberapa arsitektur CNN yaitu ResNet-50, VGG16, dan	Proses klasifikasi dengan menerapkan fine-tuning pada model DCNN pra-terlatih terbukti efektif. Hasil eksperimen menunjukkan akurasi	Data antar kelas tidak seimbang dan ukuran data yang kecil membuat penerapan ResNet-50, VGG16, dan Xception dari awal menjadi tidak optimal.	Menggunakan dataset yang berbeda dengan kuantitas yang lebih besar dan menambahkan arsitektur Inception V3 untuk klasifikasi penyakit daun teh

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
		Proceeding - 2020 International Conference on Radar, Antenna, Microwave, Electronics and Telecommunications, ICRAMET 2020	Xception dan menerapkan fine tuning	yang diperoleh dengan ResNet-50: adalah 94.05%, VGG16 adalah: 91.26%, dan Exception adalah 91.71%	Perlu mencoba untuk menggabungkan berbagai teknik augmentasi data dan transfer learning	
6	<i>Data Augmentation using Adversarial Networks for Tea Diseases Detection</i>	R. Sandra Yuwana, Fani Fauziah, Ana Heryana, Dikdik Krisnandi, R. Budiarianto Suryo Kusumo, dan Hilman F. Pardede Jurnal Elektronika dan Telekomunikasi 2020	Mengusulkan penggunaan augmentasi data untuk mendeteksi penyakit pada daun teh. Peningkatan jumlah dan variasi citra sampel berdampak pada peningkatan akurasi deteksi penyakit pada daun teh.	Performa deteksi penyakit teh pada data augmented dievaluasi menggunakan berbagai deep convolutional neural network (DCNN) termasuk AlexNet, DenseNet, ResNet, dan Xception. Hasil percobaan menunjukkan bahwa akurasi tertinggi diperoleh oleh arsitektur DenseNet yaitu sebesar 88,84%.	Menggunakan data yang tidak seimbang sehingga ketika dilakukan proses augmentasi, augmentasi terjadi pada semua kelas, sehingga data yang paling sedikit tetap memiliki perbedaan yang cukup jauh dengan kelas yang memiliki data banyak.	Menggunakan dataset yang berbeda dengan kuantitas yang lebih besar serta melakukan beberapa percobaan skenario pada preprocessing untuk mendapatkan akurasi terbaik yaitu mencoba beberapa teknik augmentasi untuk menambah variasi gambar.

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
7.	Komparasi Algoritma MobileNet Dan Nasnet Mobile Pada Klasifikasi Penyakit Daun Teh	Nila Hardi Reputasi: Jurnal Rekayasa Perangkat Lunak 2022	Membandingkan dua algoritma yaitu MobileNet dan Nasnet Mobile untuk klasifikasi penyakit daun teh	Dari hasil Komparasi antara metode MobiNet dan NasNet Mobile dalam klasifikasi penyakit daun teh, NasNet Mobile mendapatkan hasil akurasi sebesar 88% sedangkan MobileNet sebesar 95%.	Perlu mencoba untuk menggabungkan berbagai teknik augmentasi data pada percobaan yang dilakukan	Menggunakan beberapa teknik augmentasi untuk meningkatkan akurasi serta dilakukan beberapa skenario dengan mengubah <i>learning rate</i> , <i>batch size</i> , dan <i>optimizer</i> . Klasifikasi dilakukan dengan menggunakan model Resnet-50 dan Inception-V3

Berdasarkan Tabel 2.1, perbedaan penelitian ini dengan penelitian sebelumnya terletak pada penggunaan arsitektur, dataset, teknik augmentasi yang digunakan, dan penggunaan parameter pada proses training. Arsitektur yang digunakan untuk mengekstraksi ciri daun teh adalah Resnet-50 dan Inception-V3. Dataset yang digunakan lebih beragam dan terdiri dari 6 kelas, yaitu normal leaf, *algal spot*, *brown blight*, *gray blight*, *helopeltis*, dan *red spot*. Augmentasi data yang digunakan meliputi rotasi, zoom, flip, shear, dan shift yang akan saling dikombinasikan menjadi tiga skenario untuk dianalisis teknik apa yang menghasilkan akurasi terbaik. Selanjutnya, penggunaan parameter pada proses training dilakukan dengan mengubah *learning rate*, *batch size*, dan *optimizer* untuk mencapai akurasi terbaik.

2.3. Landasan Teori

2.3.1 Klasifikasi

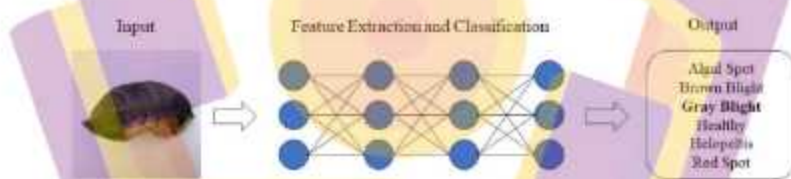
Klasifikasi merupakan salah satu bentuk penerapan pengolahan citra digital (Borman et al., 2022). Klasifikasi citra dilakukan dengan mengelompokkan citra pada sebuah kelas dimana setiap kelasnya dapat menjelaskan karakter dari objek tersebut agar dapat dikenali. Definisi lainnya menurut (Mayatopani et al., 2021). Klasifikasi juga diartikan oleh (Putry, 2022) sebagai proses pembelajaran yang berfungsi untuk mengelompokkan suatu data atau objek ke dalam kelas yang memiliki fungsi sama. Klasifikasi bertujuan untuk meningkatkan kehandalan dari hasil yang telah diperoleh dari data dengan mencoba untuk menemukan hubungan antara atribut input dengan atribut yang ditargetkan (Hendrian, 2018).

2.3.2 Deep Learning

Deep Learning adalah sebuah keilmuan baru dari machine learning dengan memiliki kemampuan yang sangat baik dalam melakukan visi komputer yang salah satunya adalah klasifikasi objek menggunakan citra (Marifatul Azizah et al., 2018). Menurut (Howard & Gugger, 2020) dalam buku *Deep Learning for Coders with Fastai and PyTorch* mendefinisikan deep learning sebagai teknik komputer yang digunakan untuk mengekstrak dan mengubah data mulai dari pengenalan ucapan manusia hingga klasifikasi citra menggunakan beberapa layer jaringan saraf dimana setiap layer akan mengambil inputan dari layer sebelumnya dan secara bertahap menyempurnakannya.

Metode pendekatan deep learning mengklasifikasikan data menjadi dua sesi yaitu sesi training dan sesi testing. Sesi training mempelajari ekstraksi fitur dari

suatu label dengan label yang lain. Pada tahapan ini tiap lapisan akan dilatih dengan algoritma tertentu yang bertujuan untuk meminimalkan kesalahan (error) dan meningkatkan akurasi. Sedangkan pada sesi testing, data-data akan diuji untuk mendapatkan dianalisis dari hasil sesi training. Menurut (Setiawan, 2020a) dalam buku *Deep Learning Menggunakan Convolutional Neural Network*, deep learning merupakan proses pembelajaran yang terdiri dari banyak layer yang dimulai dari layer awal yang sederhana hingga layer akhir yang menghasilkan fitur-fitur kompleks. Berbeda dengan machine learning, pada deep learning proses ekstraksi fitur dan klasifikasi dilakukan secara otomatis dengan menginputkan gambar mentah ke network kemudian network akan mengidentifikasi pola gambar ketika gambar tersebut melewati layer untuk menghasilkan fitur seperti pada Gambar 2.1

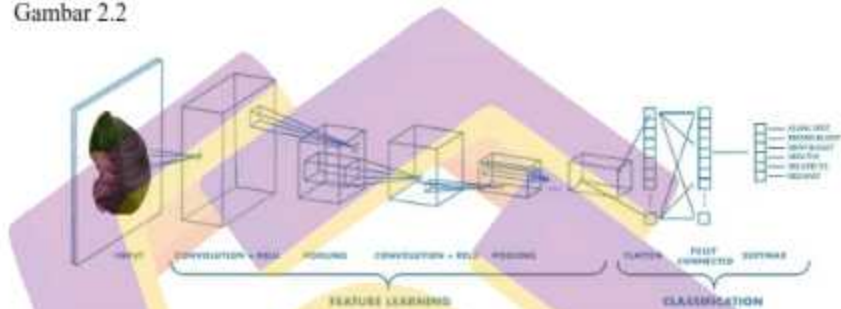


Gambar 2.1 Cara Kerja Deep Learning

2.3.3 Convolutional Neural Network

Convolutional Neural Networks (CNN) merupakan multilayer perceptron yang menjadi salah satu jenis dari deep natural network yang sering digunakan untuk mengklasifikasikan citra digital (Rahman et al., 2021). CNN memiliki arsitektur network yang terdiri dari puluhan hingga ratusan layer (Setiawan, 2020b). Selain itu (Setiawan, 2020a) juga menjelaskan bahwa CNN memproses citra melalui network layer dimana setiap layer akan melakukan pembelajaran kemudian

menghasilkan output pada kelas tertentu. Output dari tiap layer akan digunakan untuk memasukan layer selanjutnya. Di awal network, layer akan menghasilkan fitur yang sederhana seperti warna, kecerahan, dan *edges*. Selanjutnya, network akan menghasilkan fitur yang lebih kompleks. Arsitektur CNN dapat dilihat pada Gambar 2.2



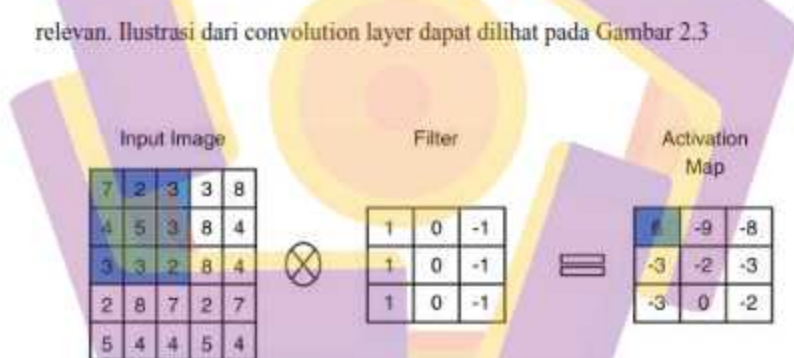
Gambar 2.2 Cara Kerja CNN

CNN memiliki dua bagian utama yaitu *Feature Learning* dan *Classification* seperti yang terlihat pada Gambar 2.2. *Feature Learning* terdiri dari lapisan konvolusi, fungsi aktivasi seperti ReLU (Rectified Linear Unit), dan Pooling. Lapisan ini sering disusun ulang untuk menjadi beberapa lapisan yang disesuaikan dengan kebutuhan arsitektur yang dibuat. Bagian *classification* terdiri dari *Flatten*, *Fully Connected*, dan *Softmax*. Pemilihan parameter dalam membangun CNN sangat mempengaruhi kinerja jaringan CNN. Pada Gambar 2.2 terlihat bahwa input CNN merupakan citra daun teh yang ingin diklasifikasikan kedalam 6 kelas. Untuk bisa mengenali daun teh tersebut termasuk ke dalam kelas *algal spot*, *brown blight*, *gray blight*, *healthy*, *helopeltis*, atau *red spot*, setiap lapisan CNN harus dikenali terlebih dahulu fungsi serta peran masing-masing agar memudahkan proses

membangun arsitektur CNN yang baru atau mengeksploitasi arsitektur CNN yang sudah tersedia (Existing CNN).

2.3.4 Layer Convolutional

Layer convolutional berfungsi untuk operasi convolution terhadap sejumlah node pada citra menggunakan beberapa filter untuk membantu neural network mengenali objek berdasarkan atribut-atributnya (Rohim et al., 2019). Output dari convolutional layer terdiri dari sejumlah matriks atau volume yang disebut feature map atau activation map. Setiap elemen dalam feature map merepresentasikan respons dari kernel terhadap suatu bagian input. Kernel tersebut akan dilatih selama proses pelatihan jaringan saraf sehingga dapat mengekstraksi fitur-fitur yang relevan. Ilustrasi dari convolution layer dapat dilihat pada Gambar 2.3



Gambar 2.3 Proses Layer Convolution

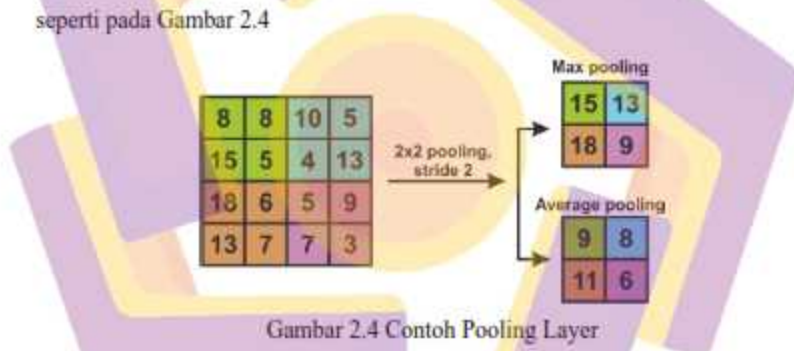
2.3.5 Activation Function

Activation Function adalah sebuah proses matematis yang berfungsi untuk mengaktifkan atau menonaktifkan neuron. Activation Function biasa disebut sebagai transfer function karena pada prosesnya terjadi perubahan fungsi linear dari jumlah berbobot menjadi model non-linear (Al-gaashani et al., 2022). Tujuan dari fungsi aktivasi adalah untuk memperkenalkan non-linearitas ke dalam jaringan

saraf sehingga jaringan saraf mampu mempelajari hubungan yang kompleks antara input dan output.

2.3.6 Pooling Layer

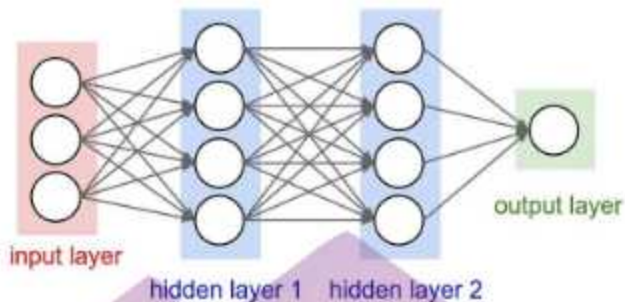
Poling merupakan proses mereduksi ukuran sebuah data gambar dengan tujuan untuk meningkatkan invariansi posisi dari fitur serta mempercepat komputasi dan mengontrol terjadinya *overfitting*. Pooling memiliki dua jenis yaitu max pooling dan average pooling (Elgendy, 2020). Max pooling merupakan nilai tertinggi dari sub patch pada konvolusi layer sebelumnya, sedangkan average pooling adalah rata-rata dari tiap bagian feature map. Proses poling dilakukan seperti pada Gambar 2.4



Gambar 2.4 Contoh Pooling Layer

2.3.7 Fully Connected Layer

Fully Connected Layer secara sederhana dapat diartikan sebagai sebuah neural network multilayer perceptron. Fully connected layer berfungsi untuk mendapatkan hasil dari proses ekstraksi fitur yang dilakukan sebelumnya dengan cara menentukan fitur mana yang saling berhubungan dan menyatukan semua node menjadi satu dimensi (Elgendy, 2020). Pada Gambar 2.5 terlihat bahwa setiap node dalam tiap layer akan saling terhubung dengan semua node di layer sebelumnya, itulah ilustrasi terjadinya fully connected layer.



Gambar 2.5 Proses Fully Connected Layer

2.3.8 Data Augmentasi

Data augmentasi merupakan teknik yang digunakan untuk mengurangi *overfitting* dengan meningkatkan ukuran dataset untuk proses pembelajaran sehingga mampu meningkatkan akurasi pada deep learning (Fadillah et al., 2021). Teknik ini membuat data citra yang digunakan menjadi lebih bervariasi dan sangat berguna ketika dataset yang dimiliki berjumlah sedikit (Shanmugaman, 2018). Augmentasi data dilakukan dengan melakukan transformasi pada data tanpa mengubah label dari setiap data yang ada karena proses augmentasi ini akan memiliki dimensi yang sama dengan kondisi citra sebelum dilakukan proses augmentasi. Beberapa teknik augmentasi data yang sering digunakan diantaranya:

A. Rotasi

Pada teknik rotasi, gambar akan diputar untuk menangani tingkat perubahan pada objek. Rentang rotasi akan ditentukan terlebih dahulu untuk kemudian gambar yang dihasilkan akan diputar secara acak sesuai dengan rentang yang diberikan. Gambar 2.6 merupakan gambar asli sebelum

mengalami augmentasi dan setelah mengalami augmentasi dengan teknik rotasi



Gambar 2.6 Teknik Augmentasi Data Rotasi

B. Zoom

Pada teknik zoom, gambar akan diperbesar atau diperkecil secara acak sesuai dengan rentang zoom yang diberikan. Gambar 2.7 merupakan perbandingan gambar sebelum di augmentasi dan setelah mengalami augmentasi dengan teknik zoom.



Gambar 2.7 Teknik Augmentasi Data Zoom

C. Horizontal Flip

Pada teknik horizontal flip, gambar akan dicerminkan atau dibalik dalam arah horizontal. Gambar 2.8 merupakan perbandingan gambar

sebelum diaugmentasi dan setelah mengalami augmentasi dengan teknik horizontal flip.



Gambar 2.8 Teknik Augmentasi Data Horizontal Flip

D. Vertical Flip

Pada teknik vertical flip, gambar akan dicerminkan atau dibalik dalam arah vertical. Gambar 2.9 merupakan perbandingan gambar sebelum diaugmentasi dan setelah mengalami augmentasi dengan teknik vertical flip.



Gambar 2.9 Teknik Augmentasi Data Vertical Flip

E. Shearing

Pada teknik shear, gambar akan digeser dan dimiringkan sesuai dengan rentang yang ditetapkan. Teknik ini berbeda dengan teknik rotasi karena teknik ini ditetapkan pada satu sumbu untuk kemudian gambar akan

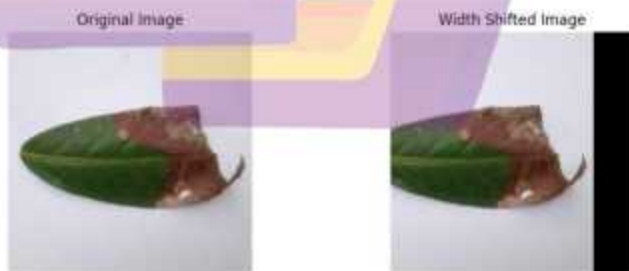
meregang pada sudut lain yang disebut sudut geser. Gambar 2.10 merupakan perbandingan gambar sebelum diaugmentasi dan setelah mengalami augmentasi dengan teknik shear



Gambar 2.10 Teknik Augmentasi Data Shear

F. Widht Shift

Pada teknik width shift, gambar akan megalami pergeseran horizontal secara acak dari gambar asli dengan memindahkan piksel-piksel gambar ke kanan atau ke kiri dengan jumlah piksel tertentu yang dipilih secara acak. Gambar 2.11 merupakan perbandingan gambar sebelum diaugmentasi dan setelah mengalami augmentasi dengan widht shift.



Gambar 2.11 Teknik Augmentasi Data Widht Shift

G. Height Shift

Pada teknik height shift, gambar akan mengalami pergeseran vertikal acak dari gambar asli dengan memindahkan piksel-piksel gambar ke atas atau ke bawah dengan jumlah piksel tertentu yang dipilih secara acak. Gambar 2.12 merupakan perbandingan gambar sebelum diaugmentasi dan setelah mengalami augmentasi dengan teknik height shift.



Gambar 2.12 Teknik Augmentasi Data Height Shift

2.3.9 Transfer Learning

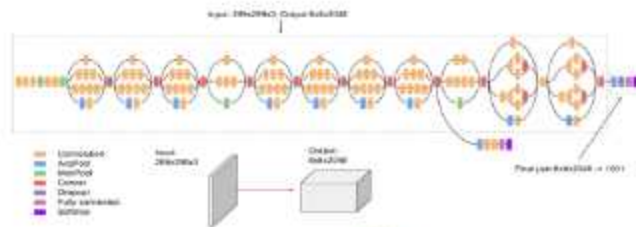
Transfer Learning menurut (Setiawan, 2020a) didefinisikan sebagai metode yang bekerja dengan memanfaatkan *network* yang sudah ada untuk kemudian dilakukan modifikasi atau pembaruan parameter-parameter pada *network* tersebut. Transfer learning menjadikan *network* yang sudah termodifikasi sebagai proses *training* dengan tugas yang berbeda. (Li et al., 2017) menjelaskan bahwa arsitektur CNN yang digunakan untuk proses transfer learning tidak memerlukan proses *training* dari awal karena arsitektur *network* telah mengenali fitur-fitur (warna, tekstur, dan bentuk) dari proses *training* yang telah dilakukan sebelumnya. Transfer

learning mampu melakukan proses training dengan baik walaupun data training yang dimiliki terbatas.

2.3.10 Inception V3

Inception-v3 menurut (Janakiramalah & Kalyani, 2021) adalah jaringan *deep learning* yang memiliki 48 layer sehingga mampu mempelajari peta fitur dengan efisien. Pada Inception V3, inputan gambar secara default akan berukuran 299×299 . Dibandingkan dengan Inception v1 dan v2, struktur jaringan pada Inception v3 menggunakan metode kernel splitting konvolusi untuk membagi integral volume besar menjadi konvolusi kecil. Pada Inception V3, konvolusi 3×3 dibagi menjadi konvolusi 3×1 dan 1×3 . Melalui metode pemisahan tersebut, jumlah parameter dapat dikurangi sehingga kecepatan pelatihan jaringan dapat dipercepat dan fitur spasial dapat diekstraksi dengan lebih efektif (Dong et al., 2020).

Dalam penelitian yang dilakukan (Jahagirdar & Budihal, 2020), menambahkan bahwa Inception V3 terdiri dari dua bagian. Bagian pertama merupakan ekstraksi fitur dan bagian kedua merupakan bagian klasifikasi. Ekstraksi fitur dilakukan dengan convolutional dan pooling layers, sedangkan bagian klasifikasi terdiri dari lapisan yang terhubung sepenuhnya dan softmax seperti yang ditunjukkan pada Gambar 2.13. dan detail arsitektur inception-v3 dapat dilihat pada Tabel 2.2.



Gambar 2.13 Arsitektur Inception V3

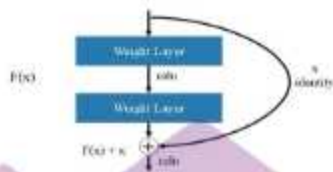
Tabel 2.2 Detail Arsitektur Inception V3

Layer (Type)	Path Size/stride or remarks	Input Size
Conv	3x3/2	200x200x3
Conv	3x3/1	149x149x32
Conv padded	3x3/1	147x147x32
Pool	3x3/2	147x147x64
Conv	3x3/1	63x73x64
Conv	3x3/2	71x71x80
Conv	3x3/1	35x35x192
3x Inception A		35x35x288
5x Inception B		17x17x768
2x Inception C		8x8x1280
Pool	8x8	8x8x2048
Linear	Logits	1x1x2048
Softmax	Classifier	1x1x1000

2.3.11 ResNet-50

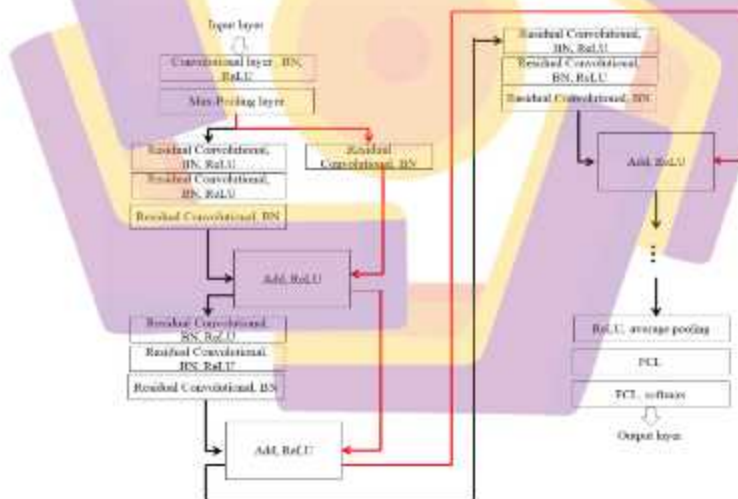
Resnet (Residual Network) merupakan jaringan deep learning yang diperkenalkan oleh He, Xiangyu Zhang, Shaoqing Ren, dan Jian Sun. Resnet memiliki arsitektur yang lebih mendalam dengan menumpuk beberapa lapisan tambahan pada deep neural network supaya mampu mempelajari fitur yang lebih kompleks sehingga dapat menghindari terjadinya *overfitting* (Rahman et al., 2021). Arsitektur Resnet-50 dibangun dari *Residual Bloks* dengan mengadopsi *residual learning* untuk diterapkan pada beberapa layer. *Residual Learning* melakukan skip

connection dari keluaran sebuah layer menjadi masukan dari layer yang dalam. *Residual Bloks* pada *residual learning* ditunjukkan pada Gambar 2.14.



Gambar 2.14 Residual Bloks

Arsitektur ResNet-50 memiliki 177 layer dan 16 *skip connection* (Setiawan, 2020a). Arsitektur ResNet-50 ditunjukkan pada Gambar 2.15, dimana BN merupakan Batch Normalization dan FCL adalah Fully Connected Network. Garis penghubung dengan warna merah menunjukkan *skip connection*.



Gambar 2.15 Arsitektur Arsitektur ResNet-50

2.3.12 Optimizer

Pada penerapan deep learning, optimizer memiliki peran yang penting dalam melatih model (Bera & Shrivastava, 2020). Tugas utama pengoptimal selama pelatihan jaringan adalah memperbarui bobot untuk mengurangi nilai fungsi kerugian. Pengoptimal dapat memberikan dampak besar pada konvergensi proses pelatihan, terutama untuk pembelajaran transfer (Kandel & Castelli, 2020b).

2.3.13 Learning Rate

Learning rate merupakan hyperparameter yang berjalan dalam sebuah model. learning rate menentukan seberapa besar atau kecil perubahan yang harus dilakukan pada bobot (weights) model berdasarkan nilai gradien (turunan) fungsi kerugian (loss function) terhadap bobot tersebut. Learning rate yang terlalu kecil dapat menyebabkan proses training berjalan lebih lama, sementara learning rate yang terlalu besar dapat menyebabkan pembelajaran set bobot menjadi tidak optimal karena proses berjalan terlalu cepat sehingga proses pelatihan tidak stabil (Kandel & Castelli, 2020a).

2.3.14 Batch Size

Batch size merupakan salah satu hyperparameter yang dapat dioptimalkan selama pelatihan model karena mempengaruhi sejumlah aspek dalam proses pelatihan, seperti kecepatan konvergensi model, penggunaan memori, dan waktu pelatihan. Secara garis besar, batch size merupakan jumlah sampel data yang digunakan dalam satu iterasi pelatihan. Batch size yang terlalu tinggi akan membuat jaringan membutuhkan waktu lebih lama untuk mencapai titik konvergensi, namun jika terlalu rendah akan membuat kinerja model menjadi tidak optimal (Kandel &

Castelli, 2020a). Sehingga, pemilihan ukuran batch yang tepat dapat memengaruhi performa dan kecepatan konvergensi model yang dibuat.

2.3.15 Confusion Matrix

Confusion Matrix merupakan pendekatan yang memberikan pernyataan singkat tentang hasil prediksi pada masalah klasifikasi biner dan multi kelas (Navlani et al., 2021). Sebuah *Confusion Matrix* digunakan untuk membuat analisis lengkap berupa data statistic dengan lebih cepat dan hasil yang diperoleh juga lebih mudah dibaca dan dimengerti melalui visualisasi data yang jelas. *Confusion Matrix* menunjukkan proses prediksi yang dibentuk oleh model yang digunakan (Yudono et al., 2022). Visualisasi *Confusion Matrix* pada proses klasifikasi dua kelas digambarkan dengan dua baris dan dua kolom seperti pada Tabel 2.3.

Tabel 2.3 Visualisasi Confuxion Matrix

Kelas	Prediksi <i>Positive</i>	Prediksi <i>Negative</i>
<i>Positive</i>	TP (<i>True Positive</i>)	FN (<i>False Negative</i>)
<i>Negative</i>	FP (<i>False Positive</i>)	TN (<i>True Negative</i>)

Pada Tabel 2.3 dapat dijelaskan bahwa nilai *True Positive* merupakan data positif yang terdeteksi benar, Nilai *True Positive* (TP) menyatakan kasus yang diperkirakan iya dan pada kenyataannya kasusnya adalah iya, nilai *False Negatif* (FN) menyatakan kasus yang diperkirakan sebagai tidak dan pada kenyataannya kasusnya adalah iya, nilai *False Positif* (FS) menyatakan kasus yang diperkirakan iya dan pada kenyataannya kasusnya adalah tidak, sedangkan nilai *True Negatif* (TN) menyatakan kasus yang diperkirakan tidak dan pada kenyataannya kasusnya

adalah tidak. Berdasarkan dari nilai TP, FN, FP, dan TN dapat diperoleh nilai akurasi, presisi, *recall*, dan *F-Score*.

a. Akurasi

Pada tahap ini akan menemukan akurasi model yang dihitung dari *confusion matrix*. Perhitungan ini akan memberikan hasil terkait seberapa akurat model prediksi yang telah dibuat (Navlani et al., 2021). Akurasi dirumuskan sebagai berikut:

$$\text{Akurasi} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100\% \quad (1)$$

b. Presisi

Persentase positif kasus dari total kasus yang diprediksi dalam dataset. Pada presisi, kita bisa mengetahui sampai tingkat apa model yang telah dibuat ketika dikatakan benar. Presisi dirumuskan sebagai berikut:

$$\text{Presisi} = \frac{TP}{(FP+TP)} \times 100\% \quad (2)$$

c. Recall

Persentase kasus positif dari total kasus aktual yang ada pada kumpulan data. Recall dirumuskan sebagai berikut:

$$\text{Recall} = \frac{TP}{(FN+TP)} \times 100\% \quad (3)$$

d. F1-Score

F1-score memberikan bobot yang sama untuk presisi dan recall, yang berarti itu menunjukkan keseimbangan antara keduanya. F1-Score adalah metrik yang lebih baik untuk data yang tidak seimbang dibandingkan dengan accuracy (Yutika et al., 2021). F1-Score dirumuskan sebagai berikut:

$$F1\ Score = \frac{2x\ (recall\ x\ presisi)}{(recall\ +\ presisi)} \times 100 \quad (4)$$

2.3.16 Grad-CAM

Model CNN secara umum disebut sebagai *black box* karena pemrosesan gambar yang terjadi didalamnya sulit dipahami secara rinci terkait bagaimana CNN membuat keputusan dan mengapa hasilnya seperti itu (Moujahid et al., 2022). Pada penelitian yang dilakukan oleh (Selvaraju et al., 2020), diusulkan Grad-CAM sebagai pendekatan menarik untuk memberikan wawasan lebih mendalam tentang bagian-bagian gambar yang paling berpengaruh terhadap keputusan klasifikasi model menggunakan CNN. Dengan memeriksa aktivasi gradien dari lapisan terakhir dari model yang diusulkan, penggunaan Grad-CAM menggambarkan area pada gambar yang berpengaruh terhadap keputusan yang dibuat model. Sehingga dapat memberikan pemahaman visual tentang bagaimana model yang digunakan memandang gambar dan area mana yang paling mempengaruhi keputusan kelasnya.

Pada penelitian ini, ResNet-50 menggunakan lapisan `conv5_block3_out` dan Inception V3 menggunakan lapisan `mixed10`. Lapisan tersebut merupakan lapisan konvolusi terakhir dari kedua arsitektur. Hasil Grad-CAM akan berupa sebuah *heatmap* yang dapat dihubungkan dengan gambar asli untuk visualisasi. *Heatmap* ini membantu memahami apakah model memprediksi daun teh berdasarkan wilayah daun yang berpenyakit dengan benar atau tidak.

2.3.17 Mean, Standar Deviasi, dan P-Value

Mean merupakan nilai rata-rata yang mendeskripsikan jumlah data untuk dibandingkan dengan jumlah masing-masing variabel (Gahagho et al., 2021). Rumus mean dapat dilihat pada persamaan (5).

$$\text{Mean} = \frac{\sum x}{n} \quad (5)$$

Pada persamaan (5), x merupakan nilai tiap data, dan n merupakan jumlah total data. Selanjutnya, standar deviasi merupakan hasil pengukuran penyebaran distribusi maupun variabilitas yang terdapat pada data (Gahagho et al., 2021). Persamaan standar deviasi dapat dilihat pada (6).

$$\text{Standar Deviasi} = \sqrt{\frac{\sum(x-\bar{x})^2}{n-1}} \quad (6)$$

Pada persamaan (6), x merupakan nilai tiap data, \bar{x} merupakan rerata dari data, dan n merupakan jumlah total data. Selanjutnya, p-value merupakan ukuran yang mampu menentukan seberapa signifikan perbedaan kondisi yang sedang dibandingkan (Stevani et al., 2021). Salah satu cara untuk menghitung p-value adalah dengan menggunakan uji Anova. Pada uji Anova terdapat *One Way Anova*. *One Way Anova* ini digunakan untuk menguji rata-rata sebuah percobaan menggunakan satu faktor dimana faktor tersebut terdiri dari tiga kelompok atau lebih (Kim, 2017). Semakin kecil nilai p-value dari tingkat signifikansi yang ditetapkan (biasanya 0,05), maka semakin besar perbedaan yang terjadi pada kondisi yang sedang dibandingkan, sebaliknya, semakin besar nilai p-value, maka semakin kecil perbedaan yang terjadi pada kondisi yang sedang dibandingkan. Untuk memperoleh nilai p-value menggunakan uji Anova, langkah pertama adalah

dengan mencari nilai F-statistic. Nilai F-statistic diperoleh dengan menggunakan rumus

$$F = \frac{\text{Varian antar Kelompok}}{\text{Varian dalam Kelompok}} = \frac{\sum_{i=1}^K n_i(\bar{Y}_i - \bar{Y})^2 / (K-1)}{\sum_{i,j=1}^N (Y_{ij} - \bar{Y})^2 / (N-K)} \quad (7)$$

Pada persamaan (7), \bar{Y}_i adalah mean dari grup i ; n_i adalah jumlah observasi kelompok i ; \bar{Y} adalah rata-rata keseluruhan; K adalah jumlah kelompok; Y_{ij} adalah nilai observasi ke- j kelompok i ; dan N adalah jumlah seluruh nilai observasi. Setelah memperoleh nilai F-statistik, selanjutnya adalah menentukan derajat kebebasan. Terdapat dua derajat kebebasan yang akan dihitung, yaitu derajat kebebasan antar kelompok ($df1$) dan derajat kebebasan dalam kelompok ($df2$). Derajat kebebasan dapat dihitung menggunakan rumus:

$$df1 = k - 1 \quad (8)$$

$$df2 = N - k \quad (9)$$

Selanjutnya, dapat dilakukan dengan menggunakan tabel distribusi F pada excel untuk menemukan probabilitas distribusi F yang sesuai dengan nilai F-statistic atau dapat menghitung menggunakan bahasa pemrograman python.

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian




Sifat penelitian ini termasuk ke dalam jenis eksperimen atau didasarkan pada percobaan dengan melakukan pengujian. Percobaan yang dilakukan pada penelitian ini adalah pemanfaatan teknik *data augmentation* untuk mengatasi jumlah data yang tidak seimbang pada data citra daun teh yang digunakan untuk bahan percobaan. Selain itu, penelitian akan memanfaatkan beberapa teknik augmentasi untuk memperbanyak variasi data yang dari data yang dimiliki pada proses training. Kemudian dilakukan percobaan pada model arsitektur Resnet-50 dan Inception V3 untuk mendapatkan tingkat akurasi yang terbaik. Jenis dan pendekatan penelitian adalah penelitian kuantitatif. Pada penelitian ini melakukan perhitungan secara matematis terhadap data-data penelitian.

3.2. Metode Pengumpulan Data

Pengumpulan data dilakukan untuk mendapatkan berbagai macam data yang diperlukan dalam kegiatan penelitian. Pada pengumpulan data, dataset yang digunakan berasal dari penelitian yang dilakukan (Datta & Gupta, 2023). Dataset diperoleh menggunakan kamera ponsel 12 megapiksel untuk menangkap 2.800 foto daun teh dari berbagai perkebunan teh di distrik Unakoti, Tripura, India. Setiap foto diambil dari jarak kurang lebih 30 cm di atas dedaunan dengan menggunakan setting kamera natural dan resolusi 3000 x 3000 pixel. Gambar yang diperoleh kemudian diperkecil menjadi 256px × 256px. Selanjutnya, untuk meningkatkan

kemampuan generalisasi pengklasifikasian ukuran database ditingkatkan menjadi 5867 menggunakan beberapa teknik augmentasi pada seluruh kelas. Sehingga, dataset yang digunakan terdiri dari 5867 data yang terbagi menjadi 6 kelas dengan rincian 1000 data daun sehat, 1000 data daun dengan penyakit *algal spot*, 867 data daun dengan penyakit *brown blight*, 1000 data daun dengan penyakit *gray blight*, 1000 data daun dengan penyakit *helopeltis*, dan 1000 data daun dengan penyakit *red spot*. Detail dataset yang digunakan beserta karakteristiknya dapat dilihat pada Tabel 3.1

Tabel 3.1 Detail Dataset

Kelas	Sampel Gambar	Jumlah	Karakteristik
<i>Algal spot</i>		1000	Penyakit ini disebabkan oleh alga parasit hijau <i>Cephaleuros virescens</i> . Penyakit ini memiliki bercak atau noda pada daun yang berwarna merah kecoklatan atau ungu tua. Selain itu, juga terjadi pembentukan struktur serabut atau bulu pada daun
<i>Brown blight</i>		867	Penyakit ini disebabkan oleh <i>Colletotrichum Camelliae Massee</i> . Penyakit ini memiliki bercak berwarna kecoklatan atau bahkan hitam. Bercak pada penyakit ini dapat memiliki tepi yang berbeda dan dapat bertambah seiring waktu.
<i>Gray blight</i>		1000	Penyakit ini disebabkan oleh <i>Pestalozzia theae Sawada</i> . Penyakit ini memiliki bercak berwarna keabuan, coklat, merah, atau hitam.

Tabel 3.1 Lanjutan

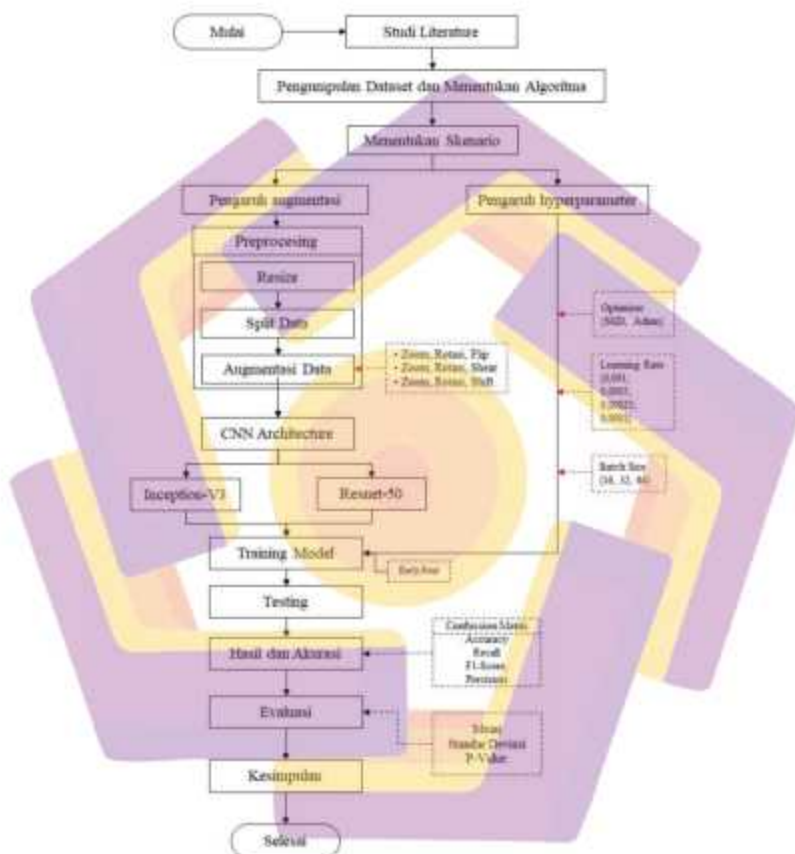
Kelas	Sampel Gambar	Jumlah	Karakteristik
<i>Healthy</i>		1000	Daun daun teh sehat cenderung berwarna hijau tua atau gelap dan tidak memiliki bercak atau deformasi (ada lipatan, goresan, atau bagian yang hilang)
<i>Helopeltis</i>		1000	Penyakit ini disebabkan oleh serangga <i>Helopeltis theivora</i> . Penyakit ini memiliki tanda-tanda cacat pada daun, yaitu adanya keriput pada daun atau bercak berwarna kemerahan atau kecoklatan.
<i>Red spot</i>		1000	Penyakit ini disebabkan oleh <i>Phyllosticta theicola</i> Petch. Penyakit ini memiliki bercak pada daun dengan warna kemerahan atau kecoklatan, terdapat pengeringan jaringan daun, dan terjadi pembentukan lesi (luka)

3.3. Metode Analisis Data

Metode analisis data merupakan tahapan pengelolaan data yang digunakan pada penelitian ini untuk mendapatkan hasil yang diharapkan dari permasalahan yang telah dirumuskan. Pada penelitian ini analisis yang digunakan berupa analisis kuantitatif dengan mengolah data yang telah dikumpulkan menggunakan model Resnet-50 dan Inception-V3. Data daun teh yang sudah terkumpul selanjutnya akan dilakukan analisis dengan beberapa proses tertentu secara bertahap sampai dapat digunakan proses klasifikasi antara daun yang sehat dan daun yang berpenyakit. Analisis dilakukan dengan menghitung *precision*, *recall* dan F1 Score yang dihasilkan melalui pengklasifikasian penyakit pada daun teh

3.4. Alur Penelitian

Alur penelitian yang digunakan pada penelitian ini ditunjukkan pada Gambar 3.1.



Gambar 3.1 Alur penelitian

Pada penelitian ini terdapat beberapa langkah utama pada penelitian seperti pada gambar 5 dengan rincian alur sebagai berikut:

- a. Studi Literature

Studi literature dilakukan dengan menelaah penelitian terdahulu. Pada tahap ini peneliti melakukan studi literature dari jurnal bereputasi dan beberapa sumber buku yang relevan dengan topik dan permasalahan yang diangkat dengan tujuan mendapat referensi tentang metode dan algoritma yang cocok untuk diterapkan pada penelitian ini.

b. Pengumpulan Data dan Penentuan Algoritma

Pada pengumpulan data, dataset yang digunakan berasal dari penelitian yang dilakukan (Datta & Gupta, 2023), dengan jumlah data sebesar 5867 data yang terbagi menjadi 6 kelas dengan rincian 1000 data daun dengan penyakit *algal spot*, 867 data daun dengan penyakit *brown blight*, 1000 data daun dengan penyakit *gray blight*, 1000 data daun sehat, 1000 data daun dengan penyakit *helopeltis*, dan 1000 data daun dengan penyakit *red spot*. Algoritma yang akan digunakan pada klasifikasi penyakit daun teh adalah ResNet-50 dan Inception V3.

c. Menentukan Skenario

Penentuan skenario percobaan pada penelitian ini berguna untuk menganalisa perbandingan hasil percobaan dengan memperhatikan penggunaan parameter dan teknik augmentasi yang digunakan pada proses training. Parameter yang digunakan dalam adalah *batch size*, *optimizer*, dan *learning rate*. Teknik augmentasi yang akan digunakan adalah zoom, rotasi, vertical flip, horizontal flip, *shear*, *widht shift*, dan *height shift*. Sedangkan parameter yang akan digunakan adalah *optimizer*, *learning rate*, dan *batch size*.

d. Preprocessing Data

Tahapan preprocessing data merupakan tahapan yang digunakan untuk mengolah dan mengelola data sehingga dapat digunakan, pada tahap ini terdapat beberapa proses yang dilakukan yaitu mulai dari resize, split data, dan data augmentasi.

i. Resize

Proses resize dilakukan pada setiap dataset yang telah dikumpulkan agar semua gambar memiliki ukuran yang sama. Proses ini akan menyesuaikan data masukan pada sistem agar tidak terjadi sebuah kesalahan karena ketidakseimbangan ukuran pada data. Pada tahap ini gambar akan diresize dari citra yang sebelumnya 256x256 pixel menjadi 224x224 pixel.

ii. Split data

Proses split data dilakukan pembagian dataset sebesar 80% untuk data training, 10% untuk data testing dan 10% untuk validasi data.

iii. Augmentasi Data

Augmentasi data dilakukan untuk memperluas jumlah data pelatihan dengan membuat variasi baru dari data yang sudah ada. Augmentasi yang dilakukan adalah *rotation*, *zoom*, *vertical flip*, dan *horizontal flip*; *rotation*, *zoom*, dan *shear*; *rotation*, *zoom*, *weight shift* dan *widht shift*.

e. CNN Arsitektur

Algoritma yang akan digunakan adalah CNN. CNN digunakan untuk mengetahui tingkat performa dari proses klasifikasi dan pengenalan citra dari dataset. Arsitektur model yang digunakan adalah ResNet-50 dan Inception V3.

i. ResNet-50

Pada model arsitektur ResNet-50 citra gambar akan melalui proses konvolusi dengan ukuran filter 7×7 dan stride 2 yang kemudian menghasilkan feature map. Selanjutnya dilakukan proses normalisasi dengan *batch normalization*. Hasil dari proses normalisasi selanjutnya masuk kedalam aktivasi layer, yang didalamnya terdapat fungsi ReLU untuk membuat hasil dari ekstraksi fitur menjadi non-linear. Setelah proses yang dilakukan oleh fungsi ReLU kemudian dilakukan proses reduksi pada setiap lapisan layer dengan max pooling sebelum diteruskan pada tahap konvolusi kedua. Diantara tahap convolution kedua sampai dengan kelima, proses feature extraction dilakukan oleh kombinasi convolutional block dan identity block. Setelah proses feature extraction selesai, kemudian feature map diproses ke dalam fully connected layer untuk melakukan proses prediksi dibantu dengan activation function sigmoid.

ii. Inception V3

Pada model arsitektur Inception V3 terdiri dari blok bangunan simetris dan asimetris building blocks seperti pada gambar 8. Tahapan Inception V3 meliputi konvolusi, average pooling, max pooling, concatenations, dropouts, dan fully connected layers. Batch normalization digunakan pada seluruh rangkaian model dan diterapkan pada input aktivasi. Loss terhadap proses yang terjadi pada model dihitung menggunakan softmax.

f. Training Model

Pada proses training dilakukan untuk melatih setiap model algoritma yang digunakan sehingga mendapatkan hasil evaluasi yang dapat memberikan nilai akurasi yang tinggi. Pada tahap ini akan dilakukan 2 skenario yaitu mengatur hyperparameter tuning dan augmentasi. *Hyperparameter tuning* yang akan diatur adalah *optimizer*, *batch size*, dan *learning rate*. Untuk menghindari terjadinya *overfitting*, akan digunakan *early stop* sehingga ketika akurasi tidak mengalami kenaikan dalam beberapa langkah, proses training akan berhenti. Setiap skenario akan dilakukan training sebanyak 3 kali untuk mendapatkan nilai rata-rata, standar deviasi, dan p value.

g. Hasil dan Akurasi

Tahapan ini digunakan untuk mengetahui hasil akurasi yang diperoleh di setiap skenario menggunakan-50 atau Inception V3. Metode

pengujian yang digunakan adalah *confusion matrix multi-class* untuk mendapatkan nilai *Accuracy*, *Recall*, *F1-Score*, dan *Precision*.

h. Evaluasi

Tahap ini merupakan tahap untuk melakukan evaluasi berdasarkan hasil akurasi yang diperoleh Resnet-50 dan Inception V3. Evaluasi yang dilakukan adalah mengetahui performa arsitektur Resnet-50 dan Inception V3 dengan mengatur beberapa hyperparameter selanjutnya membandingkan akurasi dari arsitektur Resnet-50 dan Inception V3 dengan penelitian serupa yang pernah dilakukan sebelumnya oleh (Datta & Gupta, 2023).

i. Kesimpulan

Tahap akhir dari alur penelitian ini adalah pengambilan kesimpulan setelah dilakukan beberapa tahapan mulai dari studi literature hingga evaluasi yang dilakukan. Tahapan ini memaparkan temuan yang dihasilkan dari setiap skenario

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

Bab ini menjelaskan hasil penelitian yang dilakukan berdasarkan alur penelitian pada Gambar 3.1. Penelitian ini diselesaikan melalui beberapa tahapan yaitu pengumpulan dataset, analisis data, menentukan algoritma dan skenario percobaan, dan yang terakhir melakukan evaluasi dengan melakukan pengujian skenario untuk mendapatkan hasil terbaik.

4.1. Pengumpulan Data

Data yang digunakan pada penelitian ini adalah kumpulan gambar daun teh sehat dan 5 daun teh berpenyakit. Data diperoleh melalui penelitian yang dilakukan oleh (Datta & Gupta, 2023), yang dapat diakses secara bebas melalui situs kaggle. Dataset yang digunakan terdiri dari 5867 data yang terbagi menjadi 6 kelas dengan rincian 1000 data daun sehat, 1000 data daun dengan penyakit *algae spot*, 867 data daun dengan penyakit *brown blight*, 1000 data daun dengan penyakit *gray blight*, 1000 data daun dengan penyakit *helopeltis*, dan 1000 data daun dengan penyakit *red spot*.

4.2. Preprocessing Data

Preprocessing data merupakan salah satu bagian terpenting pada proses analisis data. Tahapan ini akan mengelola data yang diperoleh agar dapat digunakan untuk meningkatkan kinerja model klasifikasi yang diusulkan. Setelah melakukan pengumpulan data, tahapan selanjutnya adalah menyimpan dataset pada satu

direktori. Data yang telah tersimpan pada direktori akan dimuat menggunakan 'load_files' ke dalam array NumPy. Proses ini dapat dilihat pada Gambar 4.1

```
# Load images into NumPy array
images = load_files(INPUT_PATH, random_state=3, shuffle=True)
X = np.array(images['filenames'])
y = np.array(images['target'])
labels = np.array(images['target_names'])

# Remove unnecessary .pyc and .py files
pyc_file = (np.where(file==X) for file in X if file.endswith(('.pyc', '.py')))
for i in pyc_file:
    X = np.delete(X, i)
    y = np.delete(y, i)
```

Gambar 4.1 Proses Load Image

Berdasarkan Gambar 4.1, terlihat bahwa fungsi `load_file` membuat file gambar disimpan ke dalam array `X`, label target disimpan ke dalam array `y`, sedangkan nama-nama kategori target disimpan ke dalam array `labels`. Selanjutnya, file dengan ekstensi `.pyc` atau `.py` akan dihapus dari array `X` dan `y` untuk memastikan bahwa data yang dimuat hanya terdiri dari file gambar yang benar-benar relevan dan siap digunakan dalam analisis atau pemodelan selanjutnya.

Tahapan berikutnya adalah *resize*. Tahapan ini dilakukan pada keseluruhan data agar gambar yang diproses memiliki ukuran yang sama. Pada penelitian ini, ukuran gambar yang digunakan untuk kedua arsitektur adalah 224 x 224 piksel. Perubahan ukuran ini dilakukan tanpa mengubah atau memotong bentuk gambar sehingga tidak akan menghilangkan detail gambar. Proses ini berguna untuk menyesuaikan data input pada sistem sehingga dapat mencegah kesalahan karena ukuran data yang tidak seimbang.

Tahapan selanjutnya adalah melakukan split data. Split data dilakukan dengan menggunakan library `folder split` dengan rasio 80% untuk data training, 10% untuk data validasi, dan 10% untuk data testing. Setelah data terbagi menjadi

training, validasi, dan testing, selanjutnya dilakukan proses augmentasi. Proses augmentasi ini hanya diterapkan pada data training dengan tujuan untuk membuat data training menjadi lebih bervariasi. Teknik augmentasi yang digunakan adalah rotasi, zoom, vertical flip, horizontal flip, shear, dan shift. Detail parameter pada teknik augmentasi yang digunakan disajikan pada Tabel 4.1

Tabel 4.1 Detail Parameter Augmentasi

Augmentasi	Parameter
Rotasi	25
Zoom	0,2
Vertical Flip	True
Horizontal Flip	True
Shear	0,2
Width shift	0,2
Weight shift	0,2

4.3. Model Klasifikasi

Metode klasifikasi yang digunakan pada penelitian ini adalah ResNet-50 dan Inception V3. Kedua arsitektur yang digunakan akan mendapat perlakuan yang sama. Pertama, pustaka Keras diimpor terlebih dahulu dari TensorFlow. Selanjutnya, `pre_trained_model` diinisialisasi menggunakan model ResNet-50 atau Inception V3 yang telah dilatih sebelumnya. Pada penelitian ini, kedua arsitektur tidak akan menyertakan lapisan `fully connected` terakhir (`top layer`) yang dimana nanti akan digantikan lapisan `fully connected` sesuai dengan kebutuhan. Lapisan-lapisan dalam kedua model juga diatur agar tidak dapat berubah bobotnya selama proses pelatihan berikutnya. Selanjutnya, beberapa lapisan tambahan ditambahkan untuk mengadaptasi model sesuai dengan tugas spesifik yang diinginkan. Proses ini dapat dilihat pada Gambar 4.2

```

for layer in pre_trained_model.layers:
    layer.trainable = False

# Add custom layers
x = pre_trained_model.output
x = BatchNormalization()(x)
# Add Pooling layer
x = GlobalAveragePooling2D()(x)
# Add a fully connected layer with 512 nodes and ReLU activation
x = Dense(512, activation="relu")(x)
# Add a dropout with rate 0.5
x = Dropout(0.5)(x)

# Specify final output layer with Softmax activation
predictions = Dense(6, activation="softmax")(x)

```

Gambar 4.2 Custom Layer

Berdasarkan pada Gambar 4.2, terdapat penambahan penggunaan lapisan Batch Normalization untuk mempercepat dan menstabilkan proses pelatihan, serta penambahan lapisan Global Average Pooling untuk meratakan output dari model. Lapisan dense (fully connected) juga ditambahkan untuk memproses fitur-fitur yang telah diekstraksi sebelumnya. Ini termasuk lapisan dengan fungsi aktivasi ReLU, yang membantu memperkenalkan sifat non-linearitas pada model. selanjutnya, lapisan dropout dengan nilai 0,5 diterapkan untuk mencegah *overfitting* dengan cara mengabaikan secara acak sebagian neuron selama proses pelatihan. Terakhir, lapisan output ditambahkan untuk menghasilkan prediksi berdasarkan jumlah kelas yang diinginkan. Fungsi aktivasi softmax digunakan untuk menghasilkan probabilitas dari setiap kelas.

4.4. Skenario Percobaan

Pada penelitian ini terdapat beberapa skenario yang akan dilakukan. Hasil analisis dari beberapa skenario percobaan yang dilakukan akan digunakan sebagai acuan untuk menentukan model klasifikasi terbaik. Detail skenario yang akan dilakukan pada penelitian ini dapat dilihat pada Tabel 4.2

Tabel 4.2 Detail Skenario Percobaan

No	Skenario	Keterangan
1.	S1	Arsitektur
2	S2	Arsitektur + <i>Hyperparameter</i> 1. <i>Optimizer</i> 2. <i>Learning rate</i> 3. <i>Batch size</i>
3	S3	Arsitektur + <i>Hyperparameter</i> + Augmentasi

Tabel 4.2 merupakan skenario yang digunakan pada penelitian ini untuk mencari model klasifikasi terbaik dalam mengklasifikasikan penyakit daun teh. Skenario percobaan ini akan diujikan dengan pembagian data 80% training, 10% validasi, dan 10% testing. Penjelasan detail dari ketiga skenario pada Tabel 4.3 adalah sebagai berikut:

1. Skenario 1 (S1)

Pada skenario pertama, dilakukan analisis terhadap pengaruh penggunaan augmentasi pada data training menggunakan ResNet-50 dan Inception V3 dengan parameter awal menggunakan *optimizer* Adam, *learning rate* 0,001, dan *batch size* 12. Pada skenario ini kedua model akan dibandingkan performanya ketika tidak menggunakan augmentasi dan ketika menggunakan augmentasi. Teknik augmentasi yang digunakan adalah zoom, rotasi, *horizontal flip*, dan *vertical flip*. Proses augmentasi dilakukan pada seluruh data training setiap kelas sehingga akan ada perbedaan yang signifikan pada jumlah data training. Detail perbedaan jumlah data dapat dilihat pada Tabel 4.3

Tabel 4.3 Perbedaan Jumlah Data Training Skenario 1

Skenario	Jumlah Data Training
Tanpa Augmentasi	4693
Augmentasi (<i>zoom</i> , rotasi, <i>horizontal flip</i> , <i>vertical flip</i>)	23465

2. Skenario 2 (S2)

Pada skenario kedua, dilakukan analisis pengaruh penggunaan hyperparameter dalam meningkatkan akurasi pada ResNet-50 dan Inception V3. Skenario ini terbagi lagi menjadi 3 bagian yaitu *optimizer*, *learning rate*, dan *batch size*.

a. *Optimizer*

Pada skenario ini dilakukan analisis pengaruh *optimizer* untuk performa model klasifikasi menggunakan ResNet-50 dan Inception V3. *Optimizer* yang akan digunakan adalah Adam (Adaptive Moment Estimation) dan SGD (Stochastic Gradient Descent). Parameter awal yang digunakan pada skenario ini adalah *learning rate* sebesar 0,001 dan *batch size* sebesar 32, sedangkan teknik augmentasi yang digunakan adalah rotasi, zoom, horizontal flip, dan vertical flip.

b. *Learning rate*

Pada skenario ini dilakukan analisis pengaruh *learning rate* untuk performa model klasifikasi menggunakan ResNet-50 dan Inception V3. Nilai *learning rate* yang akan digunakan adalah 0,001, 0,0005, 0,00025, dan 0,0001. *Batch size* dan teknik augmentasi yang digunakan pada skenario dua ini sama seperti skenario sebelumnya, sedangkan untuk *optimizer* akan menggunakan *optimizer* yang memperoleh hasil optimal dari pengujian skenario pertama.

c. *Batch size*

Pada skenario ini dilakukan analisis pengaruh nilai *batch size* terhadap performa model klasifikasi menggunakan ResNet-50 dan Inception V3. Nilai *batch size* yang digunakan adalah 16, 32, dan 64. Parameter yang digunakan pada skenario ini akan menggunakan parameter yang memperoleh hasil optimal dari pengujian yang dilakukan pada skenario 1 dan 2.

3. Skenario 3 (S3)

Pada skenario keempat dilakukan analisis lebih lanjut terhadap pengaruh penggunaan teknik augmentasi terhadap performa model klasifikasi menggunakan ResNet-50 dan Inception V3. Pada skenario 1, telah diketahui bahwa penggunaan augmentasi mempengaruhi akurasi kedua model, sehingga pada skenario ini akan dilakukan analisis terhadap teknik yang digunakan. Terdapat tiga kombinasi teknik augmentasi yang akan dilakukan pada skenario ini, yaitu zoom, rotasi, horizontal flip, dan vertical flip; zoom, rotasi, dan shear; dan terakhir zoom, rotasi, weight shift, dan height shift. Perbedaan jumlah data yang digunakan dapat dilihat pada Tabel 4.4. Parameter yang digunakan pada skenario ini menggunakan parameter yang memperoleh hasil terbaik dari skenario yang dilakukan sebelumnya.

Tabel 4.4 Perbandingan Jumlah Data Training Skenario 3

Skenario	Jumlah Data Training
Augmentasi 1 (zoom, rotasi, horizontal flip, vertical flip)	23465
Augmentasi 2 (zoom, rotasi, shear)	18772
Augmentasi 3 (zoom, rotasi, weight shift, height shift)	23465

4.5. Hasil dan Evaluasi Model

Proses klasifikasi penyakit pada daun teh dilakukan sesuai dengan skenario yang telah ditetapkan sebelumnya. Hasil percobaan yang dilakukan akan dijelaskan secara rinci dan dilakukan analisis terhadap pengaruh penggunaan parameter, augmentasi, dan image enhanced.

4.5.1 Hasil Pelatihan Model

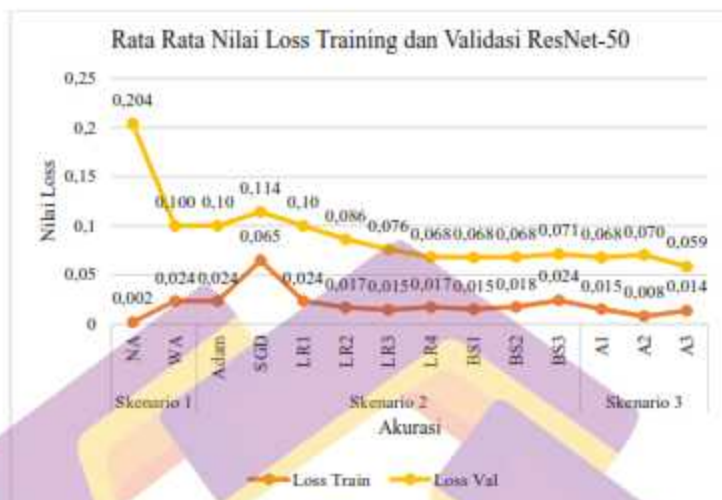
Model yang digunakan untuk klasifikasi penyakit daun teh telah dilatih pada empat skenario yang sudah ditentukan sebelumnya. Dalam penelitian ini, hasil dari setiap pelatihan akan dilakukan analisis. Grafik perbandingan akurasi training dan validasi dari keempat skenario menggunakan ResNet-50 ditampilkan pada Gambar 4.3



Gambar 4.3 Grafik Rata-Rata Akurasi Training dan Validasi ResNet-50

Berdasarkan Gambar 4.3, akurasi pada data training dan validasi dari setiap skenario tidak memiliki perbedaan yang cukup tinggi kecuali pada Skenario 1 dan Skenario 2 *Optimizer*. Hasil akurasi pada skenario pertama tanpa menggunakan augmentasi menghasilkan akurasi validasi yang jauh lebih rendah dibanding yang lain. Sedangkan pada *optimizer*, penggunaan *optimizer* Adam memiliki akurasi training dan validasi yang lebih tinggi daripada *optimizer* SGD dengan selisih akurasi sebesar 0,0119 pada training dan 0,0045 untuk validasi. Pada skenario 2 *learning rate*, terlihat data training memiliki hasil yang tidak jauh berbeda antara *learning rate* satu dengan nilai *learning rate* yang lain. Namun, terjadi perbedaan yang cukup signifikan pada akurasi validasi. Terlihat adanya kenaikan pada akurasi validasi dari *learning rate* satu ke *learning rate* empat dan mendapat nilai akurasi validasi tertinggi pada *learning rate* ke empat yaitu 0,981.

Selanjutnya, pada skenario 2 *batch size*, terlihat akurasi training dan validasi memiliki nilai yang stabil pada penggunaan *batch size* satu dan *batch size* dua namun mengalami penurunan pada penggunaan *batch size* ketiga. Pada skenario 3, terlihat bahwa akurasi training tertinggi dan validasi terendah terjadi ketika menggunakan augmentasi kedua, sedangkan pada augmentasi satu dan augmentasi tiga terlihat stabil. Selanjutnya, grafik perbandingan nilai loss training dan validasi dari keempat skenario menggunakan ResNet-50 dapat dilihat pada Gambar 4.4.



Gambar 4.4 Grafik Rata Rata Nilai Loss Training dan Validasi ResNet-50

Berdasarkan Gambar 4.4, loss pada data training dan validasi dari setiap skenario tidak memiliki perbedaan yang cukup jauh kecuali pada skenario pertama. Dapat dilihat pada grafik bahwa tanpa menggunakan augmentasi diperoleh nilai loss validasi yang sangat tinggi. Selanjutnya penggunaan *optimizer* SGD pada skenario 2 menghasilkan loss validasi kedua yang paling tinggi dibanding dengan yang lain. Pada skenario 2 *learning rate*, nilai loss pada data training terlihat stabil dibanding nilai loss pada data validasi. Nilai loss validasi terlihat mengalami penurunan seiring berkurangnya nilai *learning rate*. Selanjutnya, pada skenario 2 *batch size* terlihat nilai loss pada data training dan validasi mengalami sedikit kenaikan. Seiring bertambahnya nilai *batch size*, keduanya mengalami kenaikan. Terakhir pada skenario 3, terlihat nilai loss data train tertinggi dan loss data validasi terendah berada di penggunaan kombinasi augmentasi yang kedua. Melihat nilai akurasi dan loss pada data training dan data validasi yang cenderung stabil dan tidak

mengalami banyak perubahan (naik dan turun yang ekstrim), ini bisa menjadi indikasi bahwa ResNet-50 dapat menyesuaikan diri dengan baik pada data daun teh yang digunakan beserta parameter yang digunakan. Selanjutnya rata-rata akurasi pada data train dan data validasi pada Inception V3 ditampilkan pada Gambar 4.5.

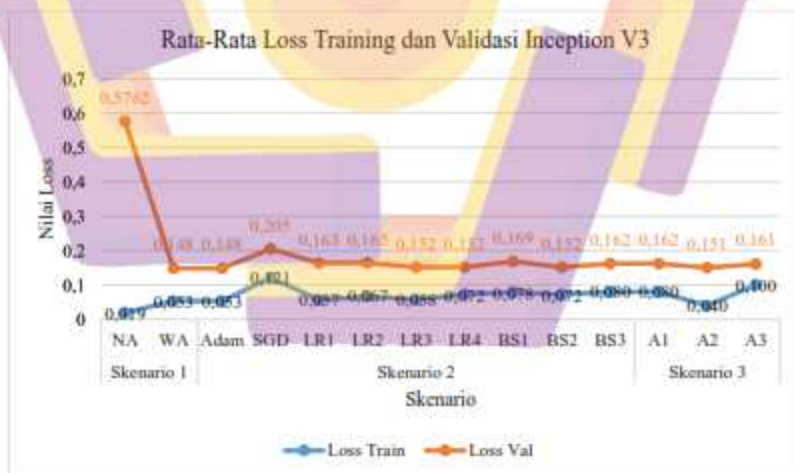


Gambar 4.5 Grafik Rata-Rata Akurasi Training dan Validasi Inception V3

Berdasarkan Gambar 4.5, akurasi pada data training dan validasi dari setiap skenario tidak memiliki perbedaan yang cukup jauh kecuali pada skenario 1. Hasil akurasi validasi pada skenario 1 memiliki nilai yang jauh dari akurasi training. Nilai ini juga memiliki nilai akurasi terendah dibanding dengan skenario yang lain. Hasil akurasi pada skenario 2 bagian *optimizer*, dengan menggunakan *optimizer* Adam dan SGD terlihat bahwa penggunaan *optimizer* Adam memiliki akurasi training dan validasi yang lebih tinggi daripada *optimizer* SGD dengan selisih akurasi sebesar 0,0171 pada training dan 0,0279 untuk validasi. Pada skenario kedua, terlihat

penggunaan *learning rate* tiga memiliki akurasi training yang paling tinggi yaitu 0,981 dan *learning rate* dua memiliki akurasi training yang paing rendah yaitu 0,9761.

Selanjutnya, pada skenario 2 *batch size*, terlihat akurasi training memiliki nilai rata-rata yang sama pada penggunaan *batch size* satu hingga *batch size* tiga namun memiliki nilai akurasi validasi yang berbeda. Akurasi validasi memiliki nilai tertinggi penggunaan *batch size* kedua dan nilai terendah pada penggunaan *batch size* kedua. Pada skenario 3, terlihat bahwa akurasi training tertinggi dan validasi tertinggi terjadi ketika menggunakan augmentasi kedua yaitu 0,9905 pada training dan 0,9684 pada validasi. Selanjutnya, grafik perbandingan nilai loss training dan validasi dari keempat skenario menggunakan Inception V3 dapat dilihat pada Gambar 4.6.



Gambar 4.6 Grafik Rata-Rata Loss Training dan Validasi Inception V3

Berdasarkan Gambar 4.6, rata-rata loss pada data training dan validasi dari setiap skenario memiliki perbedaan cukup jauh pada beberapa skenario. Pada skenario pertama, terlihat penggunaan augmentasi benar-benar mempengaruhi nilai akurasi validasi. Terdapat perbedaan yang cukup jauh pada akurasi validasi ketika tidak menggunakan augmentasi. Pada skenario 2, *optimizer* SGD menghasilkan loss training dan validasi paling tinggi dibanding dengan yang lain bahkan mencapai selisih sebesar 0,6 dengan nilai loss training dan validasi pada penggunaan *optimizer* Adam. Pada penggunaan *learning rate* dan *batch size*, nilai loss pada data training dan validasi terlihat stabil dan tidak mengalami banyak perubahan. Sedangkan pada skenario 3, dapat dilihat bahwa terjadi perubahan yang cukup signifikan pada penggunaan kombinasi augmentasi kedua. Nilai loss training yang dihasilkan mencapai 0,40 dan loss validasi 0,151. Melihat nilai akurasi dan loss pada data training dan data validasi yang cenderung stabil dan tidak mengalami banyak perubahan (naik dan turun yang ekstrim) pada semua skenario, ini bisa menjadi indikasi bahwa Inception V3 dapat menyesuaikan diri dengan baik pada data daun teh yang digunakan beserta parameter yang digunakan. Selanjutnya untuk mengukur kinerja dari ResNet-50 dan Inception V3, akan dilakukan evaluasi pada data testing.

4.5.2 Hasil Pengujian Model

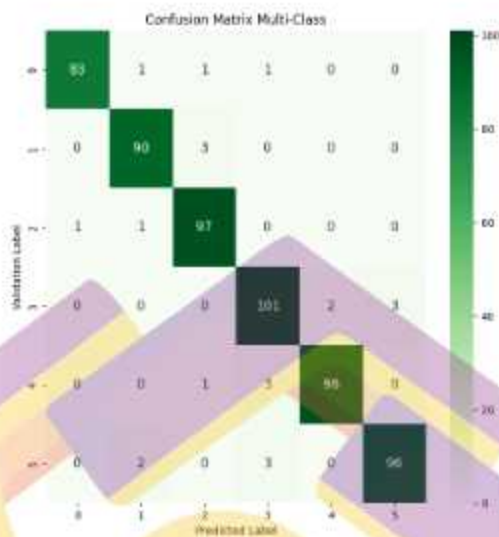
Setelah melakukan proses pelatihan pada setiap skenario, selanjutnya dilakukan proses evaluasi model menggunakan *Confusion Matrix* untuk mengetahui tingkat performa dari setiap arsitektur, dengan menampilkan nilai

accuracy, *precision*, *recall* dan *f1-score* dari setiap model yang digunakan melakukan proses klasifikasi penyakit daun teh.

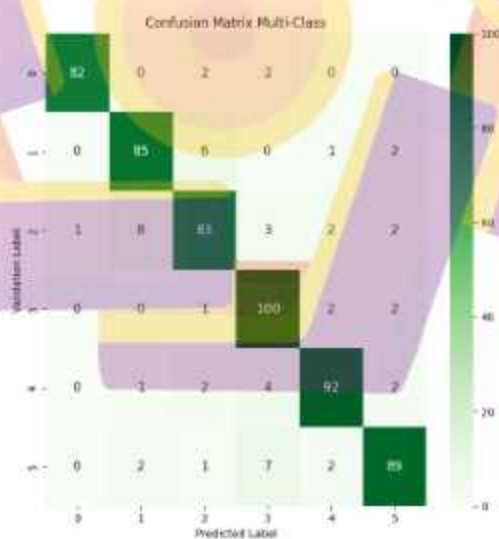
4.5.2.1 Pengujian Skenario 1

Pada skenario pertama, dilakukan percobaan dengan menggunakan mencoba untuk melihat pengaruh penggunaan augmentasi pada data training. Terdapat dua hal yang dilakukan pada skenario ini, yaitu tanpa menggunakan augmentasi dan dengan menggunakan augmentasi. Augmentasi yang digunakan adalah zoom, rotasi, horizontal flip, dan vertical flip. Sehingga akan ada perbedaan jumlah data yang cukup signifikan pada data training yang digunakan. Perbedaan jumlah data dilihat pada Tabel 4.4 Perbandingan Jumlah Data Training Skenario 3 pada sub bab 4.4 Skenario Percobaan.

Parameter yang digunakan pada keduanya sama yaitu *optimizer* Adam, *learning rate* 0,001 dan *batch size* 32. Penggunaan augmentasi dan tanpa augmentasi ini dijalankan berulang sebanyak tiga kali pada model klasifikasi ResNet-50 dan Inception V3. Sehingga hasil yang akan disajikan merupakan nilai rata-rata dari ketiga pengujian yang dilakukan. Percobaan pertama akan dilakukan dengan menjalankan model tanpa menggunakan augmentasi pada data training. Hasil pengujian tanpa augmentasi pada ResNet-50 dan Inception V3 dengan menggunakan *Confusion Matrix* secara berurutan dapat dilihat pada Gambar 4.7 dan Gambar 4.8



Gambar 4.7 *Confusion Matrix* Tanpa Augmentasi pada ResNet-50



Gambar 4.8 *Confusion Matrix* Tanpa Augmentasi pada Inception V3

Dalam *Confusion Matrix* pada Gambar 4.7, 0 mewakili penyakit Alga Spot, 1 mewakili *Brown Blight*, 2 mewakili *Gray Blight*, 3 mewakili Daun Sehat, 4 mewakili *helopeltis*, dan 5 mewakili *red spot*. Pada Gambar 4.7 terlihat terdapat 565 gambar yang diklasifikasikan dengan benar dan 22 gambar salah. Pada kelas *algal spot*, 83 gambar diklasifikasikan dengan benar, dan terdapat 3 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *brown blight*, kelas *gray blight*, dan kelas *healthy*. Pada kelas *brown blight*, 90 gambar diklasifikasikan dengan benar, dan 3 gambar salah diklasifikasikan sebagai kelas *gray blight*. Pada kelas *gray blight*, 97 gambar diklasifikasikan dengan benar, dan ada 2 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *algal spot* dan kelas *brown blight*. Pada kelas *healthy*, 101 gambar diklasifikasikan dengan benar, dan terdapat 5 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *helopeltis* dan 3 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *helopeltis*, 96 gambar diklasifikasikan dengan benar, dan terdapat 5 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *brown blight* dan 3 gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *red spot*, 96 gambar diklasifikasikan dengan benar dan terdapat 5 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *brown blight* dan 3 gambar salah diklasifikasikan sebagai kelas *healthy*.

Dari hasil *Confusion Matrix* pada Gambar 4.7, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada clasification report pada Tabel 4.5

Tabel 4.5 Clasification Report Tanpa Augmentasi ResNet-50

	Precision	Recall	F1-Score	Support
0	0,9843	0,9690	0,9766	86
1	0,9644	0,9610	0,9627	94
2	0,9539	0,9764	0,9651	99
3	0,9294	0,9528	0,9410	106
4	0,9763	0,9505	0,9632	101
5	0,9601	0,9538	0,9570	101
akurasi			0,9602	587
macro avg	0,9614	0,9606	0,9609	587
weighted avg	0,9605	0,9602	0,9603	587

Berdasarkan Tabel 4.5, pengujian menggunakan ResNet-50 ketika tidak menggunakan augmentasi mendapat rata-rata akurasi 96,02%, rata-rata presisi 96,14%, rata-rata recall 96,06%, dan rata-rata F1 Score 96,09%.

Selanjutnya, pada Gambar 4.8 *Confusion Matrix* Tanpa Augmentasi pada Inception V3, terlihat terdapat 531 gambar yang diklasifikasikan dengan benar dan 56 gambar salah. Pada kelas *algal spot*, 82 gambar diklasifikasikan dengan benar, dan ada 4 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *gray blight* dan 2 gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *brown blight*, 85 gambar diklasifikasikan dengan benar, dan ada 9 kesalahan di mana 6 gambar salah diklasifikasikan sebagai kelas *gray blight*, 1 gambar salah diklasifikasikan sebagai kelas *helopeltis*, dan 2 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *gray blight*, 83 gambar diklasifikasikan dengan benar, dan ada 16 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *algal spot*, 8 gambar salah diklasifikasikan sebagai kelas *brown blight*, 3 gambar salah diklasifikasikan sebagai kelas *healthy*, 2 gambar salah diklasifikasikan sebagai kelas *helopeltis*, dan 2 gambar lagi diklasifikasikan sebagai kelas *red spot*.

Pada kelas *healthy*, 100 gambar diklasifikasikan dengan benar, dan ada 5 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *gray blight*, 2 gambar salah diklasifikasikan sebagai kelas *helopeltis*, dan 2 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *helopeltis*, 92 gambar diklasifikasikan dengan benar, dan ada 9 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *brown blight*, 2 gambar diklasifikasikan sebagai kelas *gray blight*, 4 gambar salah diklasifikasikan sebagai kelas *healthy*, dan 2 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *red spot*, 89 gambar diklasifikasikan dengan benar, dan ada 12 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *brown blight*, 1 gambar salah diklasifikasikan sebagai kelas *gray blight*, 7 gambar salah diklasifikasikan sebagai kelas *healthy*, dan 2 gambar salah diklasifikasikan sebagai kelas *helopeltis*.

Dari hasil *Confusion Matrix* pada Gambar 4.8, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada classification report pada Tabel 4.6

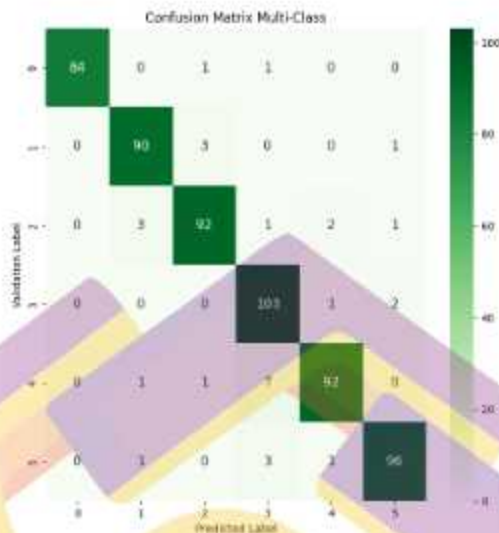
Tabel 4.6 Clasification Report Tanpa Augmentasi Inception V3

	Precision	Recall	F1-Score	Support
0	0,9763	0,9574	0,9667	86
1	0,8793	0,9043	0,8916	94
2	0,8857	0,8350	0,8596	99
3	0,8674	0,9465	0,9053	106
4	0,9233	0,9142	0,9187	101
5	0,9175	0,8812	0,8990	101
akurasi			0,9057	587
macro avg	0,9083	0,9064	0,9068	587
weighted avg	0,9066	0,9057	0,9056	587

Berdasarkan Tabel 4.6, pengujian menggunakan Inception V3 ketika tidak menggunakan augmentasi mendapat rata-rata akurasi 90,57%, rata-rata presisi 90,83%, rata-rata recall 90,64%, dan rata-rata F1 Score 90,68%.

Selanjutnya, pada percobaan kedua akan dilakukan dengan menjalankan model dengan menambahkan augmentasi pada data training. Hasil pengujian menggunakan augmentasi pada ResNet-50 dan Inception V3 dengan menggunakan *Confusion Matrix* secara berurutan dapat dilihat pada Gambar 4.9 dan Gambar 4.10





Gambar 4.10 *Confusion Matrix* dengan Augmentasi pada Inception V3

Dalam *Confusion Matrix* pada Gambar 4.9, 0 mewakili penyakit Alga Spot, 1 mewakili *Brown Blight*, 2 mewakili *Gray Blight*, 3 mewakili Daun Sehat, 4 mewakili *helopeltis*, dan 5 mewakili *red spot*. Pada Gambar 4.7 terlihat terdapat 570 gambar yang diklasifikasikan dengan benar dan 17 gambar salah. Pada kelas *algal spot*, 85 gambar diklasifikasikan dengan benar, dan ada 1 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *brown blight*, 90 gambar diklasifikasikan dengan benar, dan ada 4 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *gray blight*. Pada kelas *gray blight*, 97 gambar diklasifikasikan dengan benar, dan ada 2 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *algal spot*. Pada kelas *healthy*, 103 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *algal spot*, 1 gambar salah diklasifikasikan sebagai

kelas *helopeltis*, dan 1 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *helopeltis*, 98 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *gray blight*, dan 2 gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *red spot*, 96 gambar diklasifikasikan dengan benar, dan ada 4 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *gray blight*, 3 gambar salah diklasifikasikan sebagai kelas *healthy* dan 1 gambar salah diklasifikasikan sebagai kelas *helopeltis*.

Dari hasil *Confusion Matrix* pada Gambar 4.9, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada classification report pada Tabel 4.7.

Tabel 4.7 Classification Report dengan Augmentasi ResNet-50

	Precision	Recall	F1-Score	Support
0	0,9770	0,9884	0,9827	86
1	0,9926	0,9574	0,9747	94
2	0,9419	0,9832	0,9621	99
3	0,9509	0,9748	0,9627	106
4	0,9801	0,9736	0,9768	101
5	0,9897	0,9505	0,9697	101
akurasi			0,9710	587
macro avg	0,9720	0,9713	0,9715	587
weighted avg	0,9716	0,9710	0,9711	587

Berdasarkan Tabel 4.7, penggunaan augmentasi pada model ResNet-50 memperoleh rata-rata akurasi yang lebih tinggi yaitu 97,10%, rata-rata presisi 97,20%, rata-rata recall 97,13%, dan rata-rata F1 Score 97,15%.

Selanjutnya, pada Gambar 4.10 *Confusion Matrix* dengan Augmentasi pada Inception V3, terdapat 558 gambar yang diklasifikasikan dengan benar dan 19 gambar salah. Pada kelas *algae spot*, terdapat 83 gambar diklasifikasikan dengan benar, dan ada 2 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *gray blight* dan 2 gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *brown blight*, 91 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *gray blight* dan 1 gambar salah diklasifikasikan sebagai *red spot*. Pada kelas *gray blight*, 93 gambar diklasifikasikan dengan benar, dan ada 6 kesalahan di mana 3 gambar salah diklasifikasikan sebagai kelas *brown blight*, 1 gambar salah diklasifikasikan sebagai kelas *healthy*, 1 gambar salah diklasifikasikan sebagai kelas *helopeltis*, dan 1 gambar lagi diklasifikasikan sebagai kelas *red spot*. Pada kelas *healthy*, 103 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *gray blight*, 1 gambar salah diklasifikasikan sebagai kelas *helopeltis*, dan 1 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *helopeltis*, 93 gambar diklasifikasikan dengan benar, dan ada 8 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *brown blight*, 1 gambar diklasifikasikan sebagai kelas *gray blight*, 5 gambar salah diklasifikasikan sebagai kelas *healthy*, dan 1 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *red spot*, 95 gambar diklasifikasikan dengan benar, dan ada 6 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *gray blight*, dan 5 gambar salah diklasifikasikan sebagai kelas *healthy*.

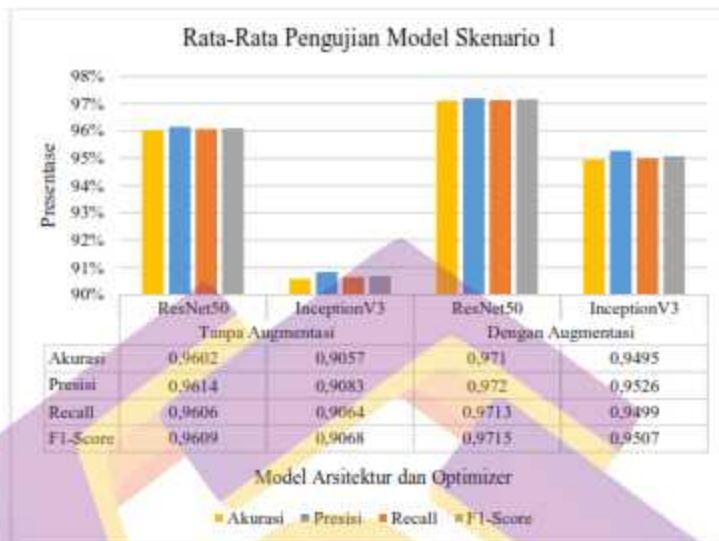
Dari hasil *Confusion Matrix* pada Gambar 4.8, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada classification report pada Tabel 4.8.

Tabel 4.8 Classification Report dengan Augmentasi Inception V3

	Precision	Recall	F1-Score	Support
0	1,0000	0,9651	0,9822	86
1	0,9512	0,9681	0,9596	94
2	0,9490	0,9394	0,9442	99
3	0,8960	0,9748	0,9337	106
4	0,9721	0,9208	0,9458	101
5	0,9564	0,9406	0,9484	101
akurasi			0,9512	587
macro avg	0,9541	0,9515	0,9523	587
weighted avg	0,9525	0,9512	0,9513	587

Berdasarkan Tabel 4.8, penggunaan augmentasi pada model Inception V3 memperoleh akurasi yang lebih tinggi yaitu rata-rata akurasi 95,12%, rata-rata presisi 95,41%, rata-rata recall 95,15%, dan rata-rata F1 Score 95,23%.

Untuk melihat secara detail pengujian pada skenario 1 ini, hasil perbandingan rata-rata dapat dilihat pada Gambar 4.11.

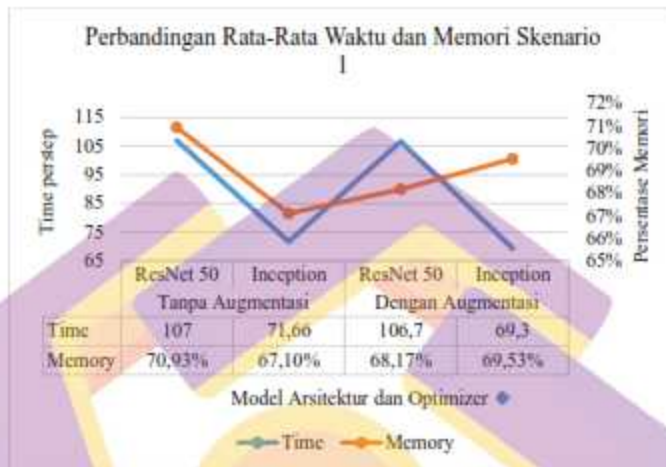


Gambar 4.11 Rata Rata Pengujian Model Skenario 1

Dapat dilihat pada Gambar 4.11, kedua model memiliki hasil yang optimal ketika menggunakan augmentasi. Pada ResNet-50, diperoleh hasil optimal dengan rata-rata 97,10%, rata-rata presisi 97,20%, rata-rata recall 97,13%, dan rata-rata F1 Score 97,15%. Sedangkan pada Inception V3 diperoleh rata-rata akurasi 95,12%, rata-rata presisi 95,41%, rata-rata recall 95,15%, dan rata-rata F1 Score 95,23%. Augmentasi yang dilakukan pada data training membantu untuk meningkatkan variasi data yang digunakan ketika proses pelatihan sehingga membantu model untuk lebih baik mengidentifikasi pola-pola yang mendasari setiap kelas. Penggunaan augmentasi ini juga mengurangi kemungkinan terjadinya *overfitting*.

Selain melakukan perbandingan dengan menggunakan akurasi, presisi, recall, dan F1-Score, penelitian ini juga memperhatikan *time consumption per step*

ketika proses pengujian dan memori yang digunakan secara keseluruhan. Gambar 4.10 merupakan perbandingan *time consumption perstep* dan penggunaan memori.



Gambar 4.12 Perbandingan Rata-Rata Waktu dan Memori Skenario 1

Dapat dilihat pada Gambar 4.12, ResNet-50 memiliki *time consumption perstep* yang lebih tinggi dibanding Inception V3. Selain itu, ketika model dijalankan tanpa menggunakan augmentasi ternyata membutuhkan lebih banyak *time consumption perstep* dibanding dengan menggunakan augmentasi. Tanpa penggunaan augmentasi, dibutuhkan rata-rata 107ms/step untuk ResNet-50 dan 71,66ms/step untuk Inception V3 sedangkan dengan penggunaan augmentasi, ResNet-50 dan Inception V3 membutuhkan rata-rata 106,7ms/step dan 69,3ms/step.

Pada perbandingan memori yang diperlukan, juga terjadi sedikit perbedaan. Tanpa penggunaan augmentasi, terlihat ResNet-50 membutuhkan 70,93% dari memori yang tersedia, sedangkan Inception V3 membutuhkan 67,10% dari jumlah memori yang tersedia. Sedangkan ketika menggunakan augmentasi, ResNet-50

membutuhkan 68,17% dari memori yang tersedia, sedangkan Inception V3 membutuhkan 69,53% dari jumlah memori yang tersedia. Perbedaan penggunaan memori ini dipengaruhi oleh arsitektur yang digunakan karena adanya perbedaan jumlah layer. Selain itu, proses iterasi juga memiliki pengaruh terhadap perbedaan penggunaan memori karena pada proses training digunakan fungsi 'early_stop' untuk menghindari terjadinya *overfitting*. Sehingga ketika proses training berhenti sebelum mencapai epoch ke 100, prosesnya akan membutuhkan memori yang lebih sedikit dibanding dengan model yang mencapai epoch ke 100. Dengan mempertimbangkan dan memahami waktu dan memori yang digunakan ini, peneliti dapat memilih pendekatan yang sesuai dengan kebutuhan untuk menyelesaikan tugas komputasi.

4.5.2.2 Pengujian Skenario 2

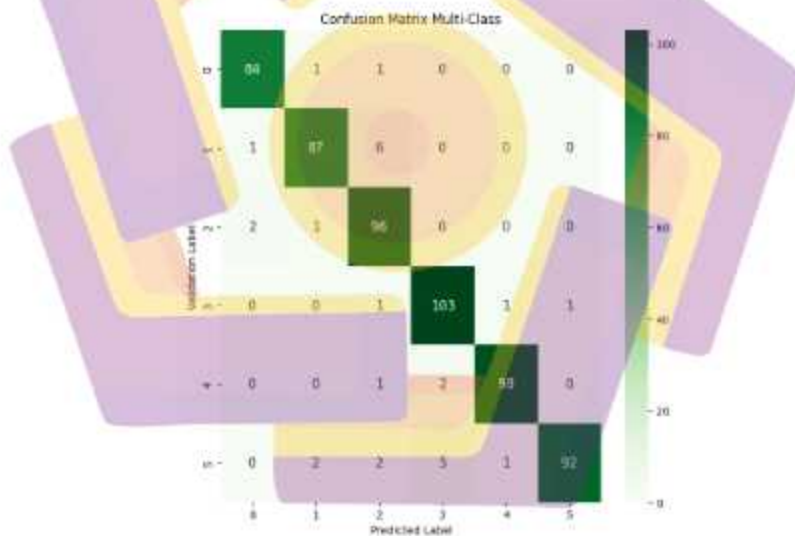
Pada skenario kedua, dilakukan percobaan dengan menggunakan mencoba untuk melihat pengaruh penggunaan *optimizer*, *learning rate*, dan *batch size*.

4.5.2.2.1 *Optimizer*

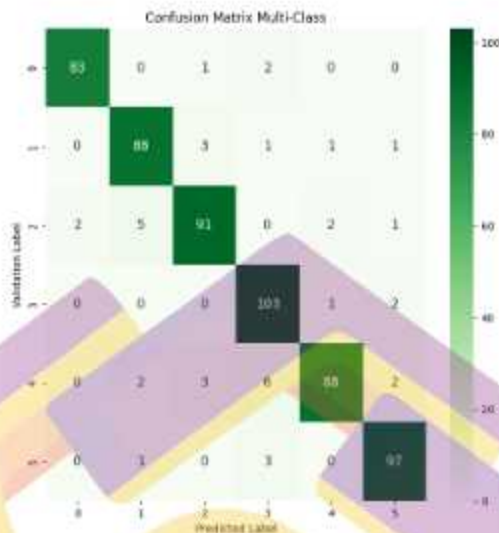
Pada skenario *optimizer*, dilakukan percobaan dengan menggunakan dua jenis *optimizer* yang berbeda untuk setiap model, yaitu SGD dan Adam. Parameter yang digunakan pada keduanya sama yaitu *learning rate* 0,001 dan *batch size* 32. Skenario ini juga menambahkan augmentasi pada data training karena telah terbukti pada skenario 1 bahwa augmentasi mempengaruhi akurasi. Pengujian pada skenario ini dilakukan sebanyak tiga kali pada kedua model. Sehingga hasil yang akan disajikan merupakan nilai rata-rata dari ketiga pengujian yang dilakukan. Percobaan pertama akan dilakukan dengan menjalankan model dengan *Optimizer*

Adam. Karena parameter yang digunakan memiliki kesamaan pada skenario 1, hasil pengujian menggunakan *optimizer* Adam pada ResNet-50 dan Inception V3 dengan menggunakan *Confusion Matrix* secara berurutan dapat dilihat pada Gambar 4.9 dan Gambar 4.10 sedangkan untuk hasil clasification report dapat dilihat pada Tabel 4.7 dan Tabel 4.8

Selanjutnya, pada percobaan kedua akan dilakukan dengan menjalankan model dengan menggunakan *optimizer* SGD. Hasil pengujian menggunakan *optimizer* SGD pada ResNet-50 dan Inception V3 dengan menggunakan *Confusion Matrix* secara berurutan dapat dilihat pada Gambar 4.13 dan Gambar 4.14



Gambar 4.13 *Confusion Matrix Optimizer* SGD pada ResNet-50



Gambar 4.14 *Confusion Matrix Optimizer SGD* pada Inception V3

Dalam *Confusion Matrix* pada Gambar 4.13, 0 mewakili penyakit Alga Spot, 1 mewakili *Brown Blight*, 2 mewakili *Gray Blight*, 3 mewakili Daun Sehat, 4 mewakili *helopeltis*, dan 5 mewakili *red spot*. Pada Gambar 4.13 terlihat terdapat 560 gambar yang diklasifikasikan dengan benar dan 27 gambar salah. Pada kelas *algal spot*, 84 gambar diklasifikasikan dengan benar, dan ada 2 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *brown blight* dan *gray blight*. Pada kelas *brown blight*, 87 gambar diklasifikasikan dengan benar, dan ada 7 kesalahan di mana 1 gambar salah diklasifikasikan sebagai *algal spot* dan 6 gambar salah diklasifikasikan sebagai kelas *gray blight*. Pada kelas *gray blight*, 96 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *algal spot* dan 1 gambar salah diklasifikasikan sebagai *brown blight*. Pada kelas *healthy*, 103 gambar diklasifikasikan dengan

benar, dan ada 3 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *gray blight*, *helopeltis*, dan kelas *red spot*. Pada kelas *helopeltis*, 98 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *gray blight*, dan 2 gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *red spot*, 92 gambar diklasifikasikan dengan benar, dan ada 10 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *algal spot*, 2 gambar salah diklasifikasikan sebagai kelas *gray blight*, 5 gambar salah diklasifikasikan sebagai kelas *healthy* dan 1 gambar salah diklasifikasikan sebagai kelas *helopeltis*.

Dari hasil *Confusion Matrix* pada Gambar 4.13, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada classification report pada Tabel 4.9.

Tabel 4.9 Classification Report *Optimizer* SGD pada ResNet-50

	Precision	Recall	F1-Score	Support
0	0,9692	0,9767	0,9730	86
1	0,9632	0,9291	0,9458	94
2	0,8972	0,9697	0,9320	99
3	0,9364	0,9717	0,9537	106
4	0,9799	0,9670	0,9734	101
5	0,9857	0,9076	0,9450	101
akurasi			0,9534	587
macro avg	0,9553	0,9536	0,9538	587
weighted avg	0,9549	0,9534	0,9535	587

Berdasarkan Tabel 4.9, penggunaan *optimizer* SGD pada model ResNet-50 memperoleh rata-rata akurasi 95,34%, rata-rata presisi 95,33%, rata-rata recall 95,36%, dan rata-rata F1 Score 95,38%.

Selanjutnya, pada Gambar 4.14 *Confusion Matrix Optimizer* SGD pada Inception V3, terdapat 550 gambar yang diklasifikasikan dengan benar dan 39 gambar salah. Pada kelas *algal spot*, 83 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *gray blight* dan 2 gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *brown blight*, 88 gambar diklasifikasikan dengan benar, dan ada 6 kesalahan di mana 3 gambar salah diklasifikasikan sebagai kelas *gray blight*, 1 gambar salah diklasifikasikan sebagai kelas *healthy*, 1 gambar salah diklasifikasikan sebagai kelas *helopeltis* dan 1 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *gray blight*, 91 gambar diklasifikasikan dengan benar, dan ada 10 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *algal spot*, 5 gambar salah diklasifikasikan sebagai kelas *brown blight*, 2 gambar salah diklasifikasikan sebagai kelas *helopeltis*, dan 1 gambar lagi diklasifikasikan sebagai kelas *red spot*. Pada kelas *healthy*, 103 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *helopeltis* dan 2 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *helopeltis*, 88 gambar diklasifikasikan dengan benar, dan ada 13 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *brown blight*, 3 gambar diklasifikasikan sebagai kelas *gray blight*, 6 gambar salah diklasifikasikan sebagai kelas *healthy*, dan 2 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *red spot*, 97 gambar

diklasifikasikan dengan benar, dan ada 4 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *brown blight* dan 3 gambar salah diklasifikasikan sebagai kelas *healthy*.

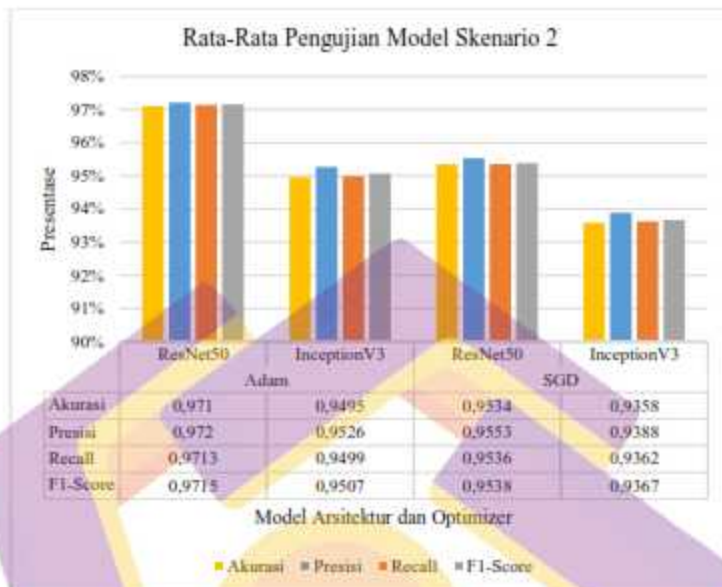
Dari hasil *Confusion Matrix* pada Gambar 4.14 Gambar 4.8, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada clasification report pada Tabel 4.10

Tabel 4.10 Clasification Report *Optimizer* SGD Inception V3

	Precision	Recall	F1-Score	Support
0	0,9920	0,9612	0,9704	86
1	0,9201	0,9397	0,9298	94
2	0,9220	0,9158	0,9189	99
3	0,8928	0,9686	0,9291	106
4	0,9670	0,8713	0,9167	101
5	0,9387	0,9604	0,9494	101
akurasi			0,9358	587
macro avg	0,9388	0,9362	0,9367	587
weighted avg	0,9373	0,9358	0,9358	587

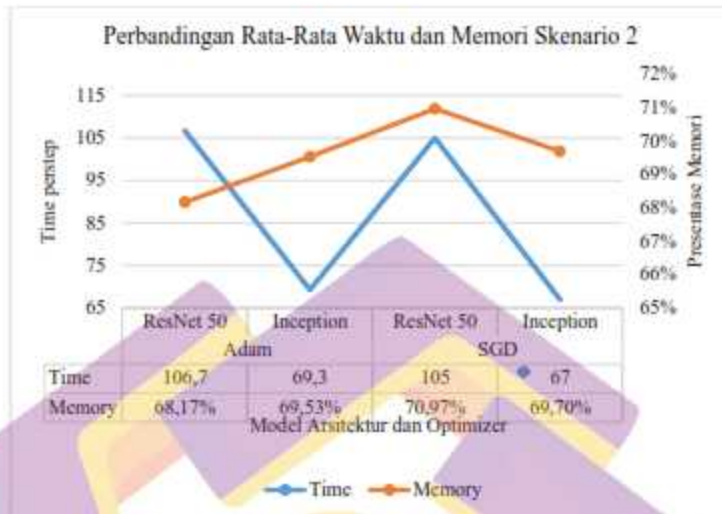
Berdasarkan Tabel 4.10, penggunaan *optimizer* SGD pada model Inception V3 memperoleh rata-rata akurasi 93,58%, rata-rata presisi 93,88%, rata-rata recall 93,62%, dan rata-rata F1 Score 93,67%.

Pada percobaan yang dilakukan, diperoleh rata-rata akurasi tertinggi ketika menggunakan *optimizer* Adam pada kedua model. Hasil perbandingan rata-rata pengujian pada penggunaan *optimizer* dapat dilihat pada Gambar 4.15.



Gambar 4.15 Rata Rata Pengujian Model Skenario 2 *Optimizer*

Dapat dilihat pada Gambar 4.15, kedua model memiliki hasil yang optimal ketika menggunakan *optimizer* Adam. Pada ResNet-50, diperoleh hasil optimal dengan rata-rata akurasi 97,10%, rata-rata presisi 97,20%, rata-rata recall 97,15%, dan rata-rata F1-Score 97,14%. Sedangkan pada Inception V3 diperoleh rata-rata akurasi 94,95, rata-rata presisi 95,26%, rata-rata recall 94,99%, dan rata-rata F1-Score 95,07%. Selain melakukan perbandingan dengan menggunakan akurasi, presisi, recall, dan F1-Score, penelitian ini juga memperhatikan *time consumption per step* ketika proses pengujian dan memori yang digunakan secara keseluruhan. Gambar 4.16 merupakan perbandingan *time consumption perstep* dan penggunaan memori.



Gambar 4.16 Perbandingan Rata-Rata Waktu dan Memori Skenario 2 *Optimizer*

Dapat dilihat pada Gambar 4.16, ResNet-50 memiliki time consumption perstep yang lebih tinggi dibanding Inception V3. Selain itu, dengan selisih yang sedikit, penggunaan Adam membutuhkan lebih banyak time consumption perstep dibanding pada *optimizer* SGD. Secara berturut-turut, pada penggunaan *optimizer* Adam, ResNet-50 dan Inception V3 membutuhkan rata-rata 106,7ms/step dan 69,3ms/step sedangkan pada penggunaan *optimizer* SGD, dibutuhkan rata-rata 105ms/step untuk ResNet-50 dan 67ms/step untuk Inception V3.

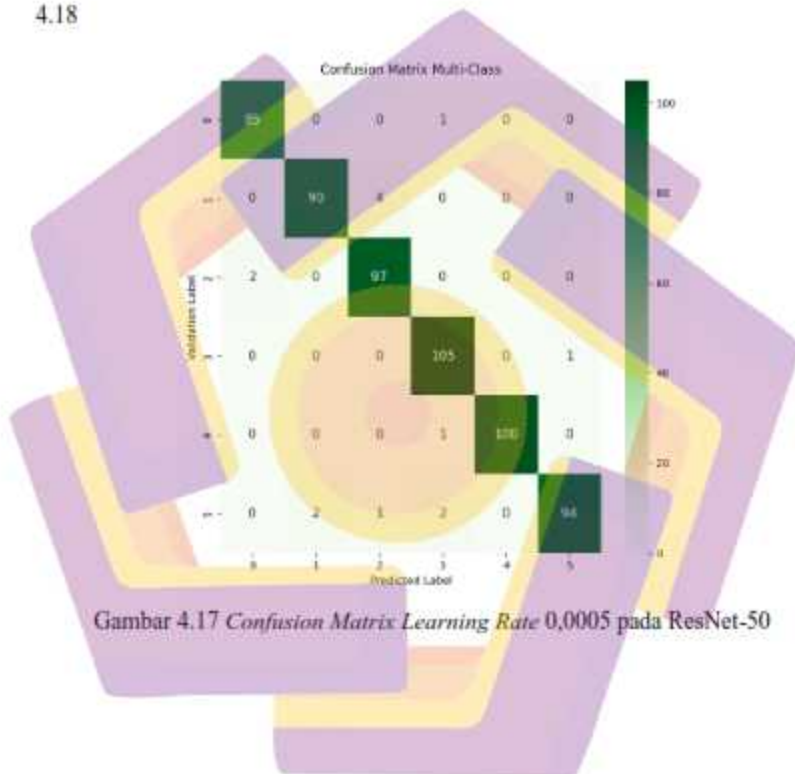
Pada perbandingan memori yang diperlukan, terjadi sedikit perbedaan antara penggunaan Adam dan SGD. Pada penggunaan Adam, terlihat ResNet-50 membutuhkan 68,17% dari memori yang tersedia, sedangkan Inception V3 membutuhkan 1,36% memori lebih banyak yaitu 69,53% dari jumlah memori yang tersedia. Sedangkan pada SGD, ResNet-50 membutuhkan 70,97% sedangkan Inception V3 membutuhkan 69,70%. Perbedaan penggunaan memori ini

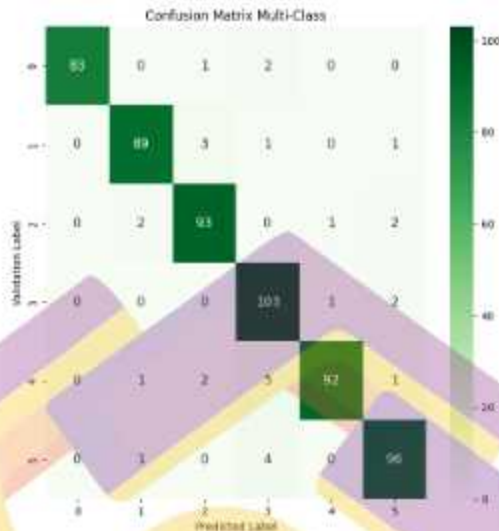
dipengaruhi oleh arsitektur yang digunakan karena adanya perbedaan jumlah layer. Selain itu, proses iterasi juga memiliki pengaruh terhadap perbedaan penggunaan memori karena pada proses training digunakan fungsi 'early_stop' untuk menghindari terjadinya *overfitting*. Sehingga ketika proses training berhenti sebelum mencapai epoch ke 100, prosesnya akan membutuhkan memori yang lebih sedikit dibanding dengan model yang mencapai epoch ke 100. Dengan mempertimbangkan dan memahami waktu dan memori yang digunakan ini, peneliti dapat memilih pendekatan yang sesuai dengan kebutuhan untuk menyelesaikan tugas komputasi.

4.5.2.2.2 *Learning Rate*

Pada skenario *learning rate*, dilakukan percobaan dengan menggunakan empat nilai *learning rate* yang berbeda untuk setiap model. *Optimizer* yang digunakan adalah Adam karena terbukti *optimizer* Adam memiliki akurasi lebih tinggi dibanding SGD dan *batch size* 32. Pengujian pada skenario ini dilakukan sebanyak tiga kali pada kedua model. Sehingga hasil yang akan disajikan merupakan nilai rata-rata dari ketiga pengujian yang dilakukan. Percobaan pertama akan dilakukan dengan menjalankan model dengan *learning rate* 0,001. Karena parameter yang digunakan memiliki kesamaan pada skenario 1 dan skenario 2 pada *optimizer*, hasil pengujian menggunakan *learning rate* 0,001 pada ResNet-50 dan Inception V3 dengan menggunakan *Confusion Matrix* secara berurutan dapat dilihat pada Gambar 4.9 dan Gambar 4.10 sedangkan untuk hasil clasification report dapat dilihat pada Tabel 4.7 dan Tabel 4.8

Selanjutnya, pada percobaan kedua akan dilakukan dengan menjalankan model dengan menggunakan *learning rate* 0,0005. Hasil pengujian menggunakan *learning rate* 0,0005 pada ResNet-50 dan Inception V3 dengan menggunakan *Confusion Matrix* secara berurutan dapat dilihat pada Gambar 4.17 dan Gambar 4.18





Gambar 4.18 *Confusion Matrix Learning Rate 0,0005 pada Inception V3*

Dalam *Confusion Matrix* pada Gambar 4.17 dan Gambar 4.18, 0 mewakili penyakit *Alga Spot*, 1 mewakili *Brown Blight*, 2 mewakili *Gray Blight*, 3 mewakili Daun Sehat, 4 mewakili *helopeltis*, dan 5 mewakili *red spot*. Pada Gambar 4.17 terlihat terdapat 571 gambar yang diklasifikasikan dengan benar dan 16 gambar salah. Pada kelas *algal spot*, 85 gambar diklasifikasikan dengan benar dan ada 1 kesalahan dimana gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *brown blight*, 90 gambar diklasifikasikan dengan benar, dan ada 4 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *gray blight*. Pada kelas *gray blight*, 97 gambar diklasifikasikan dengan benar, dan ada 2 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *algal spot*. Pada kelas *healthy*, 105 gambar diklasifikasikan dengan benar dan ada 1 kesalahan dimana gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *helopeltis*, 100 gambar

diklasifikasikan dengan benar, dan ada 1 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *red spot*, 94 gambar diklasifikasikan dengan benar, dan ada 5 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *brown blight*, 1 gambar salah diklasifikasikan sebagai kelas *gray blight*, dan 2 gambar salah diklasifikasikan sebagai kelas *healthy*.

Dari hasil *Confusion Matrix* pada Gambar 4.17, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada classification report pada Tabel 4.11.

Tabel 4.11 Classification Report *Learning Rate* 0,0005 pada ResNet-50

	Precision	Recall	F1-Score	Support
0	0,9769	0,9845	0,9807	86
1	0,9676	0,9539	0,9607	94
2	0,9387	0,9798	0,9588	99
3	0,9602	0,9874	0,9736	106
4	0,9934	0,9868	0,9901	101
5	0,9930	0,9340	0,9626	101
akurasi			0,9710	587
macro avg	0,9716	0,9711	0,9711	587
weighted avg	0,9716	0,9710	0,9710	587

Berdasarkan Tabel 4.11, penggunaan *learning rate* 0,0005 pada ResNet-50 memperoleh rata-rata akurasi 97,10%, rata-rata presisi 97,16%, rata-rata recall 97,11%, dan rata-rata F1 Score 97,11%.

Selanjutnya, pada Gambar 4.18 *Confusion Matrix Learning Rate* 0,0005 pada Inception V3, terdapat 556 gambar yang diklasifikasikan dengan benar dan 31

gambar salah. Pada kelas *algal spot*, 83 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *gray blight* dan 2 gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *brown blight*, 89 gambar diklasifikasikan dengan benar, dan ada 5 kesalahan di mana 3 gambar salah diklasifikasikan sebagai kelas *gray blight*, 1 gambar salah diklasifikasikan sebagai kelas *healthy*, dan 1 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *gray blight*, 93 gambar diklasifikasikan dengan benar, dan ada 5 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *brown blight*, 1 gambar salah diklasifikasikan sebagai kelas *helopeltis*, dan 2 gambar lagi diklasifikasikan sebagai kelas *red spot*. Pada kelas *healthy*, 103 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *helopeltis* dan 2 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *helopeltis*, 92 gambar diklasifikasikan dengan benar, dan ada 9 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *brown blight*, 2 gambar diklasifikasikan sebagai kelas *gray blight*, 5 gambar salah diklasifikasikan sebagai kelas *healthy*, dan 1 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *red spot*, 96 gambar diklasifikasikan dengan benar, dan ada 5 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *brown blight* dan 4 gambar salah diklasifikasikan sebagai kelas *healthy*.

Dari hasil *Confusion Matrix* pada Gambar 4.18 Gambar 4.8, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3),

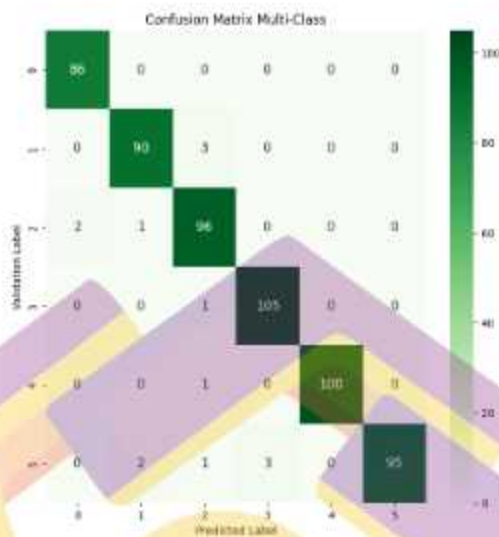
dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada classification report pada Tabel 4.12

Tabel 4.12 Classification Report *Learning Rate* 0,0005 Inception V3

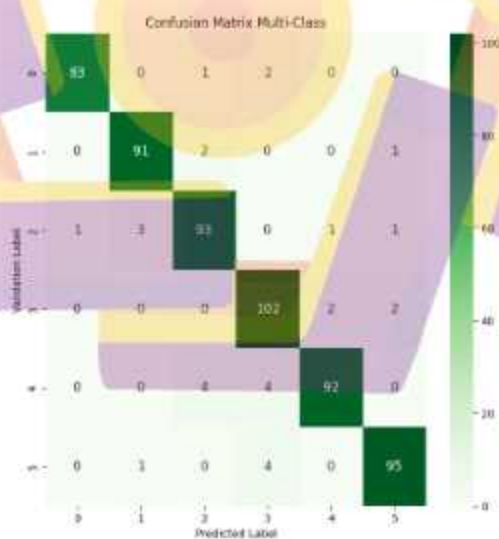
	Precision	Recall	F1-Score	Support
0	0,9960	0,9651	0,9803	86
1	0,9570	0,9468	0,9519	94
2	0,9333	0,9428	0,9380	99
3	0,8934	0,9748	0,9323	106
4	0,9788	0,9142	0,9454	101
5	0,9503	0,9472	0,9488	101
akurasi			0,9483	587
macro avg	0,9515	0,9485	0,9494	587
weighted avg	0,9498	0,9483	0,9485	587

Berdasarkan Tabel 4.12, penggunaan *learning rate* 0,0005 pada model Inception V3 memperoleh rata-rata akurasi 94,83%, rata-rata presisi 95,15%, rata-rata recall 94,85%, dan rata-rata F1 Score 94,94%.

Selanjutnya, pada percobaan ketiga akan dilakukan dengan menjalankan model dengan menggunakan *learning rate* 0,00075. Hasil pengujian menggunakan *learning rate* 0,00075 pada ResNet-50 dan Inception V3 dengan menggunakan *Confusion Matrix* secara berurutan dapat dilihat pada Gambar 4.19 dan Gambar 4.20



Gambar 4.19 *Confusion Matrix Learning Rate 0,00075 pada ResNet-50*



Gambar 4.20 *Confusion Matrix Learning Rate 0,00075 pada Inception V3*

Dalam *Confusion Matrix* pada Gambar 4.19 dan Gambar 4.20, 0 mewakili penyakit Alga Spot, 1 mewakili *Brown Blight*, 2 mewakili *Gray Blight*, 3 mewakili Daun Sehat, 4 mewakili *helopeltis*, dan 5 mewakili *red spot*. Pada Gambar 4.19, terlihat terdapat 572 gambar yang diklasifikasikan dengan benar dan 15 gambar salah. Pada kelas *algal spot*, 86 gambar diklasifikasikan dengan benar. Pada kelas *brown blight*, 90 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 3 gambar salah diklasifikasikan sebagai kelas *gray blight*. Pada kelas *gray blight*, 96 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *algal spot* dan 1 gambar salah diklasifikasikan sebagai *brown blight*. Pada kelas *healthy*, 105 gambar diklasifikasikan dengan benar, dan ada 1 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *gray blight*. Pada kelas *helopeltis*, 100 gambar diklasifikasikan dengan benar, dan ada 1 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *gray blight*. Pada kelas *red spot*, 95 gambar diklasifikasikan dengan benar, dan ada 6 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *brown blight*, 1 gambar salah diklasifikasikan sebagai kelas *gray blight*, dan 3 gambar salah diklasifikasikan sebagai kelas *healthy*.

Dari hasil *Confusion Matrix* pada Gambar 4.19, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada classification report pada Tabel 4.13.

Tabel 4.13 Clasification Report *Learning Rate* 0,00075 pada ResNet-50

	Precision	Recall	F1-Score	Support
0	0,9735	0,9961	0,9847	86
1	0,9713	0,9610	0,9661	94
2	0,9412	0,9697	0,9552	99
3	0,9723	0,9937	0,9829	106
4	1,0000	0,9934	0,9967	101
5	1,0000	0,9439	0,9711	101
akurasi			0,9761	587
macro avg	0,9764	0,9763	0,9761	587
weighted avg	0,9766	0,9761	0,9762	587

Berdasarkan Tabel 4.13, penggunaan learning rate 0,00075 pada model ResNet-50 memperoleh rata-rata akurasi 97,61%, rata-rata presisi 97,64%, rata-rata recall 97,63%, dan rata-rata F1 Score 97,61%.

Selanjutnya, pada Gambar 4.20 *Confusion Matrix Learning Rate* 0,00075 pada Inception V3, terdapat 556 gambar yang diklasifikasikan dengan benar dan 31 gambar salah. Pada kelas *algal spot*, 83 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *gray blight* dan 2 gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *brown blight*, 91 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *gray blight* dan 1 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *gray blight*, 93 gambar diklasifikasikan dengan benar, dan ada 6 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *algal spot*, 3 gambar salah diklasifikasikan sebagai kelas *brown blight*, 1 gambar salah diklasifikasikan sebagai kelas *helopeltis*, dan 1 gambar lagi diklasifikasikan sebagai kelas *red spot*. Pada kelas *healthy*, 102 gambar diklasifikasikan dengan benar, dan ada 4 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas

helopeltis dan 2 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *helopeltis*, 92 gambar diklasifikasikan dengan benar, dan ada 8 kesalahan di mana 4 gambar salah diklasifikasikan sebagai kelas *gray blight* dan 4 gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *red spot*, 95 gambar diklasifikasikan dengan benar, dan ada 5 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *brown blight* dan 4 gambar salah diklasifikasikan sebagai kelas *healthy*.

Dari hasil *Confusion Matrix* pada Gambar 4.20 Gambar 4.8, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada classification report pada Tabel 4.14.

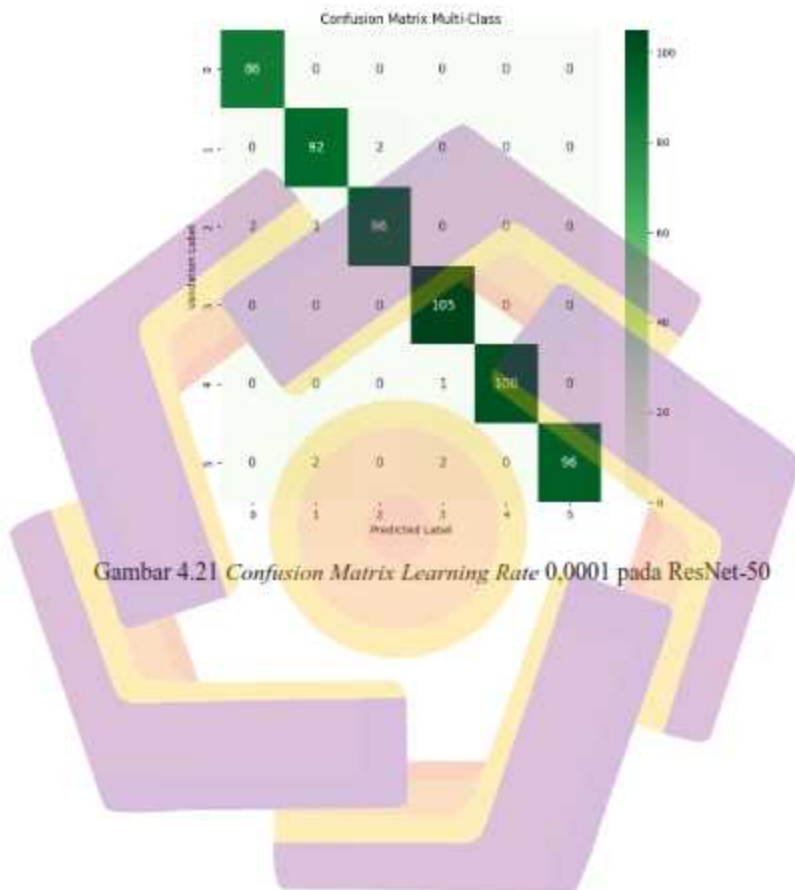
Tabel 4.14 Classification Report *Learning Rate* 0,00075 Inception V3

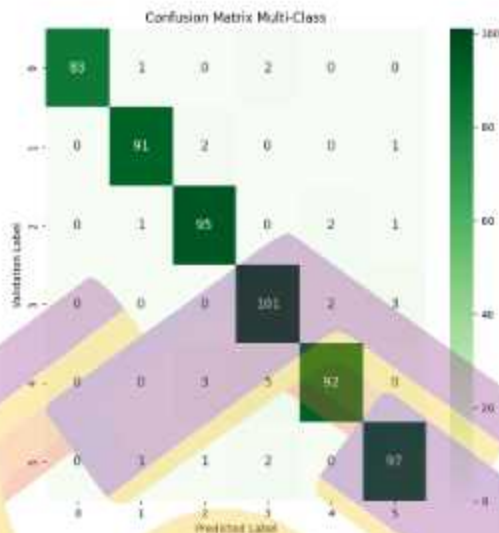
	Precision	Recall	F1-Score	Support
0	0,9843	0,9690	0,9766	86
1	0,9545	0,9681	0,9613	94
2	0,9300	0,9394	0,9347	99
3	0,9110	0,9654	0,9374	106
4	0,9652	0,9142	0,9390	101
5	0,9630	0,9439	0,9533	101
akurasi			0,9495	587
macro avg	0,9513	0,9500	0,9504	587
weighted avg	0,9502	0,9495	0,9495	587

Berdasarkan Tabel 4.14, penggunaan *learning rate* 0,00075 pada model Inception V3 memperoleh rata-rata akurasi 94,95%, rata-rata presisi 95,13%, rata-rata recall 95,00%, dan rata-rata F1 Score 95,04%.

Selanjutnya, pada percobaan keempat akan dilakukan dengan menjalankan model menggunakan *learning rate* 0,0001. Hasil pengujian menggunakan *learning*

rate 0,0001 pada ResNet-50 dan Inception V3 dengan menggunakan *Confusion Matrix* secara berurutan dapat dilihat pada Gambar 4.21 dan Gambar 4.22.





Gambar 4.22 *Confusion Matrix Learning Rate 0,0001 pada Inception V3*

Dalam *Confusion Matrix* pada Gambar 4.21 dan Gambar 4.22, 0 mewakili penyakit *Alga Spot*, 1 mewakili *Brown Blight*, 2 mewakili *Gray Blight*, 3 mewakili Daun Sehat, 4 mewakili *helopeltis*, dan 5 mewakili *red spot*. Pada Gambar 4.21 terlihat terdapat 574 gambar yang diklasifikasikan dengan benar dan 13 gambar salah. Pada kelas *algal spot* semua gambar sejumlah 86 diklasifikasikan dengan benar. Pada kelas *brown blight*, terdapat 92 gambar diklasifikasikan dengan benar, dan ada 2 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *gray blight*. Pada kelas *gray blight*, terdapat 96 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *algal spot* dan 1 gambar salah diklasifikasikan sebagai kelas *brown blight*. Pada kelas *healthy*, 105 gambar diklasifikasikan dengan benar. Pada kelas *helopeltis*, 100 gambar diklasifikasikan dengan benar, dan ada 1 kesalahan di mana gambar salah

diklasifikasikan sebagai kelas *healthy*. Pada kelas *red spot*, 96 gambar diklasifikasikan dengan benar, dan ada 4 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *brown blight* dan 2 gambar salah diklasifikasikan sebagai kelas *healthy*.

Dari hasil *Confusion Matrix* pada Gambar 4.21, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada clasification report pada Tabel 4.15.

Tabel 4.15 Clasification Report *Learning Rate* 0,0001 ResNet-50

	Precision	Recall	F1-Score	Support
0	0,9809	0,9961	0,9885	86
1	0,9615	0,9752	0,9683	94
2	0,9731	0,9731	0,9731	99
3	0,9664	0,9937	0,9798	106
4	0,9934	0,9868	0,9901	101
5	0,9965	0,9472	0,9712	101
akurasi			0,9784	587
macro avg	0,9786	0,9787	0,9785	587
weighted avg	0,9787	0,9784	0,9784	587

Berdasarkan Tabel 4.15., penggunaan *learning rate* 0,0001 pada model ResNet-50 memperoleh rata-rata akurasi 97,84%, rata-rata presisi 97,86%, rata-rata recall 97,87%, dan rata-rata F1 Score 97,85%.

Selanjutnya pada Gambar 4.22 *Confusion Matrix Learning Rate* 0,0001 pada Inception V3 terlihat terdapat 559 gambar yang diklasifikasikan dengan benar dan 28 gambar salah. Pada kelas *algal spot*, 83 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas

gray blight dan 2 gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *brown blight*, 91 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *gray blight* dan 1 gambar salah diklasifikasikan sebagai *red spot*. Pada kelas *gray blight*, 95 gambar diklasifikasikan dengan benar, dan ada 4 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *brown blight*, 2 gambar salah diklasifikasikan sebagai kelas *helopeltis*, dan 1 gambar lagi diklasifikasikan sebagai kelas *red spot*. Pada kelas *healthy*, 101 gambar diklasifikasikan dengan benar, dan ada 5 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *helopeltis* dan 3 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *helopeltis*, 92 gambar diklasifikasikan dengan benar, dan ada 9 kesalahan di mana 3 gambar salah diklasifikasikan sebagai kelas *gray blight*, 5 gambar salah diklasifikasikan sebagai kelas *healthy*, dan 1 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *red spot*, 97 gambar diklasifikasikan dengan benar, dan ada 4 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *brown blight*, 1 gambar salah diklasifikasikan sebagai kelas *gray blight*, dan 2 gambar salah diklasifikasikan sebagai kelas *healthy*.

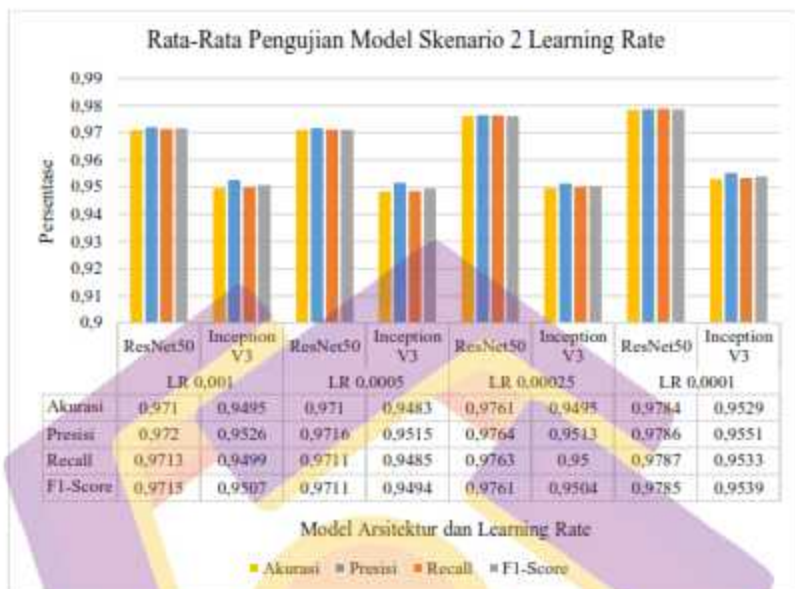
Dari hasil *Confusion Matrix* pada Gambar 4.22, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada clasification report pada Tabel 4.16.

Tabel 4.16 Clasification Report *Learning Rate* 0,0001 Inception V3

	Precision	Recall	F1-Score	Support
0	1,0000	0,9651	0,9822	86
1	0,9680	0,9645	0,9663	94
2	0,9375	0,9596	0,9484	99
3	0,9157	0,9560	0,9354	106
4	0,9583	0,9109	0,9340	101
5	0,9511	0,9637	0,9574	101
akurasi			0,9529	587
macro avg	0,9551	0,9533	0,9539	587
weighted avg	0,9535	0,9529	0,9529	587

Berdasarkan Tabel 4.16, penggunaan *learning rate* 0,0001 pada model Inception V3 memperoleh rata-rata akurasi 95,29%, rata-rata presisi 95,51%, rata-rata recall 95,33%, dan rata-rata F1 Score 95,39%.

Pada percobaan yang dilakukan, diperoleh rata-rata akurasi tertinggi ketika menggunakan *learning rate* 0,0001 pada kedua model. Hasil perbandingan rata-rata pengujian pada penggunaan *optimizer* dapat dilihat pada Gambar 4.23.



Gambar 4.23 Rata Rata Pengujian Model Skenario 2 *Learning Rate*

Dapat dilihat pada Gambar 4.23, kedua model memiliki hasil yang optimal ketika menggunakan *learning rate* 0,0001. Akurasi yang diperoleh mengalami kenaikan seiring dengan berkurangnya nilai *learning rate* walaupun pada nilai 0,001 dan 0,0005 memiliki rerata akurasi yang sama. Pada ResNet-50, diperoleh hasil optimal dengan rata-rata akurasi 97,84%, rata-rata presisi 97,86%, rata-rata recall 97,87%, dan rata-rata F1-Score 97,85%. Sedangkan pada Inception V3 diperoleh rata-rata akurasi 95,29%, rata-rata presisi 95,51%, rata-rata recall 95,33%, dan rata-rata F1-Score 95,39%.

Selain melakukan perbandingan dengan menggunakan akurasi, presisi, recall, dan F1-Score, penelitian ini juga memperhatikan *time consumption per step*

ketika proses pengujian dan memori yang digunakan secara keseluruhan. Gambar 4.14 merupakan perbandingan *time consumption perstep* dan penggunaan memori.



Gambar 4.24 Perbandingan Rata-Rata Waktu dan Memori Skenario 2 *Learning Rate*

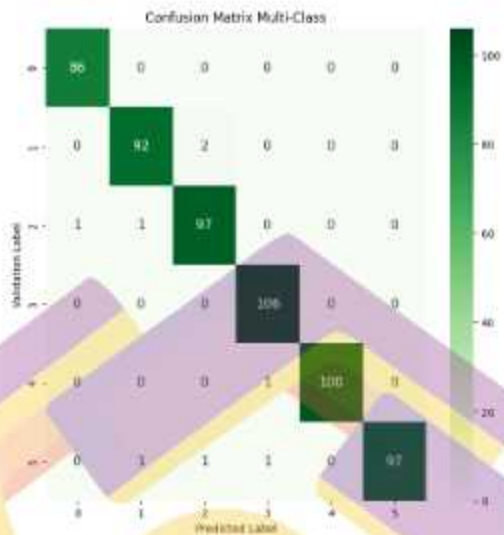
Dapat dilihat pada gambar Gambar 4.24, ResNet-50 memiliki *time consumption perstep* yang lebih tinggi dibanding Inception V3. Pada skenario dua ini, *time consumption perstep* tidak mengalami banyak perubahan untuk kedua arsitektur yang digunakan. Secara berturut-turut, pada penggunaan *learning rate* 0,001, ResNet-50 dan Inception V3 membutuhkan rata-rata 106ms/step dan 69,3ms/ste, pada *learning rate* 0,0005 ResNet-50 dan Inception V3 membutuhkan rata-rata 106,3ms/step dan 70,7ms/step, pada *learning rate* 0,00025 ResNet-50 dan Inception V3 membutuhkan rata-rata 106,7ms/step dan 71,7ms/step, dan terakhir pada *learning rate* 0,0001, membutuhkan rata-rata 107,3ms/step untuk ResNet-50 dan 73ms/step untuk Inception V3. Dibandingkan dengan *learning rate* 0,001,

learning rate 0,0001 membutuhkan waktu yang lebih lama. Hal ini terjadi karena nilai *learning rate* yang lebih tinggi akan berdampak pada proses pembelajaran model yang lebih cepat sehingga memiliki tingkat ketelitian yang cukup rendah. Sebaliknya, *learning rate* yang lebih rendah akan memiliki tingkat ketelitian yang lebih tinggi, namun dalam prosesnya akan memerlukan lebih banyak waktu.

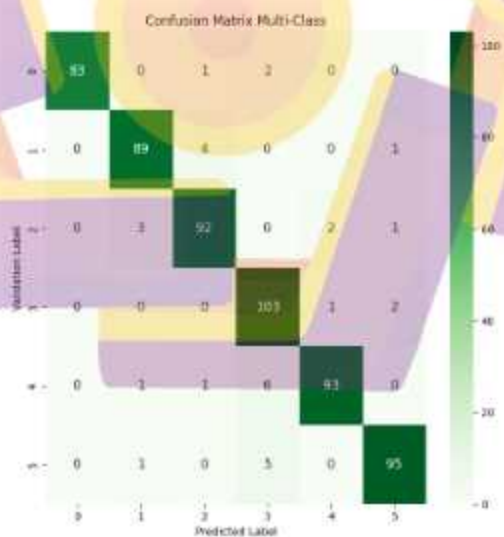
Pada perbandingan memori yang diperlukan, terlihat bahwa pada semua percobaan *learning rate*, ResNet-50 menghabiskan memori yang lebih sedikit dibanding dengan Inception V3. Perbedaan penggunaan memori ini dipengaruhi oleh arsitektur yang digunakan karena adanya perbedaan jumlah layer. Dengan mempertimbangkan dan memahami waktu dan memori yang digunakan ini, peneliti dapat memilih pendekatan yang paling efisien dan relevan dengan kebutuhan untuk menyelesaikan tugas komputasi dengan efisien.

4.5.2.2.3 *Batch Size*

Pada skenario *batch size*, dilakukan percobaan dengan menggunakan tiga nilai *batch size* yang berbeda untuk setiap model. *Optimizer* yang digunakan adalah Adam karena terbukti *optimizer* Adam memiliki akurasi lebih tinggi dibanding SGD dan *learning rate* 0,0001 karena terbukti memiliki akurasi lebih tinggi dibanding dengan nilai *learning rate* yang lain. Pengujian pada skenario ini dilakukan sebanyak tiga kali pada kedua model. Sehingga hasil yang akan disajikan merupakan nilai rata-rata dari ketiga pengujian yang dilakukan. Percobaan pertama akan dilakukan dengan menjalankan model dengan *batch size* 16. Hasil pengujian *batch size* 16 pada ResNet-50 dan Inception V3 dengan menggunakan *Confusion Matrix* secara berurutan dapat dilihat pada Gambar 4.25 dan Gambar 4.26



Gambar 4.25 Confusion Matrix Batch Size 16 pada ResNet-50



Gambar 4.26 Confusion Matrix Batch Size 16 pada Inception V3

Dalam *Confusion Matrix* pada Gambar 4.25 dan Gambar 4.26, 0 mewakili penyakit Alga Spot, 1 mewakili *Brown Blight*, 2 mewakili *Gray Blight*, 3 mewakili Daun Sehat, 4 mewakili *helopeltis*, dan 5 mewakili *red spot*. Pada Gambar 4.17 terlihat terdapat 578 gambar yang diklasifikasikan dengan benar dan 9 gambar salah. Pada kelas *algal spot*, 86 gambar diklasifikasikan dengan benar. Pada kelas *brown blight*, 92 gambar diklasifikasikan dengan benar, dan ada 2 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *gray blight*. Pada kelas *gray blight*, 97 gambar diklasifikasikan dengan benar, dan ada 2 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *algal spot* dan *brown blight*. Pada kelas *healthy*, 106 gambar diklasifikasikan dengan benar. Pada kelas *helopeltis*, 100 gambar diklasifikasikan dengan benar, dan ada 1 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *red spot*, 97 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *brown blight*, kelas *gray blight*, dan kelas *healthy*. Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada classification report pada Tabel 4.17.

Tabel 4.17 Classification Report *Batch Size* 16 pada ResNet-50

	Precision	Recall	F1-Score	Support
0	0,9809	0,9961	0,9885	86
1	0,9752	0,9752	0,9752	94
2	0,9699	0,9764	0,9732	99
3	0,9784	0,9969	0,9875	106
4	0,9967	0,9934	0,9950	101
5	0,9966	0,9604	0,9782	101
akurasi			0,9830	587
macro avg	0,9829	0,9831	0,9829	587
weighted avg	0,9831	0,9830	0,9829	587

Berdasarkan Tabel 4.17, penggunaan *batch size* 16 pada ResNet-50 memperoleh rata-rata akurasi 98,30%, rata-rata presisi 98,29%, rata-rata recall 98,31%, dan rata-rata F1 Score 98,29%.

Selanjutnya, pada Gambar 4.26 *Confusion Matrix Batch Size* 16 pada Inception V3, terdapat 555 gambar yang diklasifikasikan dengan benar dan 32 gambar salah. Pada kelas *algal spot*, 83 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *gray blight* dan 2 gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *brown blight*, 89 gambar diklasifikasikan dengan benar, dan ada 5 kesalahan di mana 4 gambar salah diklasifikasikan sebagai kelas *gray blight* dan 1 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *gray blight*, 92 gambar diklasifikasikan dengan benar, dan ada 6 kesalahan di mana 3 gambar salah diklasifikasikan sebagai kelas *brown blight*, 2 gambar salah diklasifikasikan sebagai kelas *helopeltis*, dan 1 gambar lagi diklasifikasikan sebagai kelas *red spot*. Pada kelas *healthy*, 103 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *helopeltis* dan 2 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *helopeltis*, 93 gambar diklasifikasikan dengan benar, dan ada 8 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *brown blight*, 1 gambar diklasifikasikan sebagai kelas *gray blight*, dan 6 gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *red spot*, 95 gambar diklasifikasikan dengan benar, dan ada 6 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *brown blight* dan 5 gambar salah diklasifikasikan sebagai kelas *healthy*.

Dari hasil *Confusion Matrix* pada Gambar 4.26, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada clasification report pada Tabel 4.18.

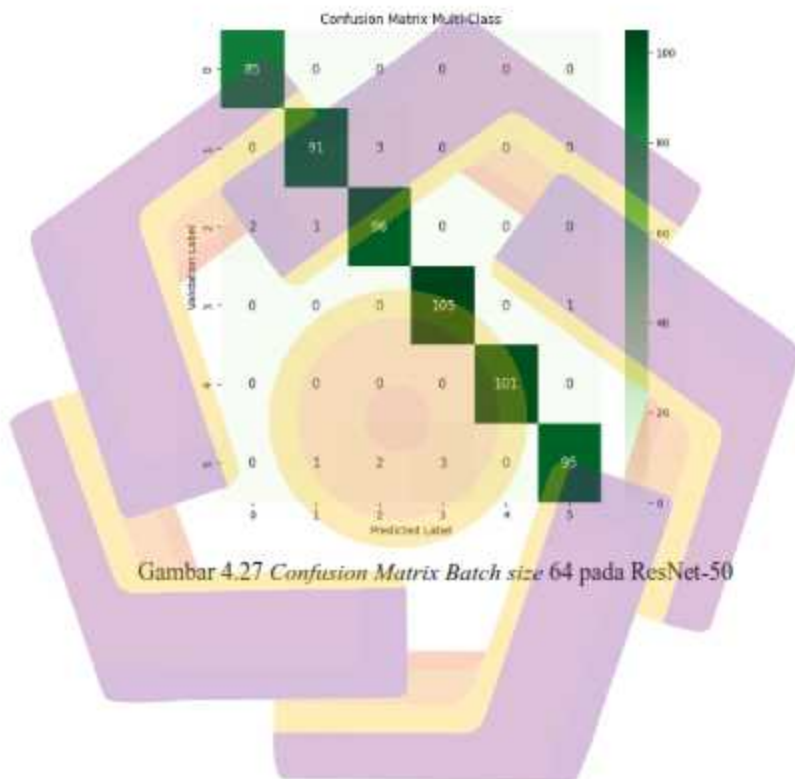
Tabel 4.18 Clasification Report *Batch size* 16 Inception V3

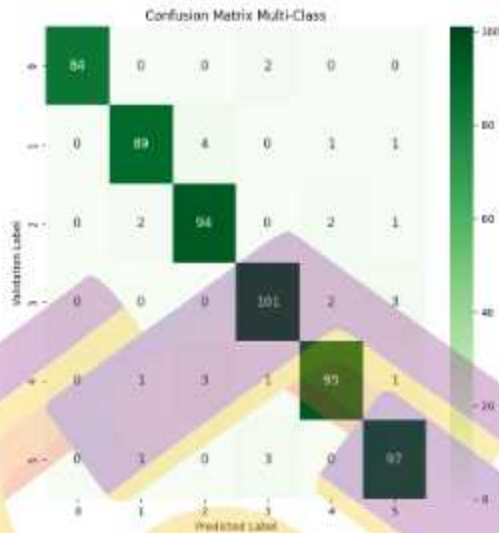
	Precision	Recall	F1-Score	Support
0	0,9960	0,9612	0,9783	86
1	0,9468	0,9468	0,9468	94
2	0,9322	0,9259	0,9291	99
3	0,8879	0,9717	0,9279	106
4	0,9654	0,9208	0,9426	101
5	0,9564	0,9406	0,9484	101
akurasi			0,9443	587
macro avg	0,9475	0,9445	0,9455	587
weighted avg	0,9458	0,9443	0,9446	587

Berdasarkan Tabel 4.18, penggunaan *batch size* 16 pada model Inception V3 memperoleh rata-rata akurasi 94,43%, rata-rata presisi 94,75%, rata-rata recall 94,45%, dan rata-rata F1 Score 94,55%.

Selanjutnya pada percobaan kedua akan dilakukan dengan menjalankan model dengan *batch size* 32. Karena parameter yang digunakan memiliki kesamaan pada skenario 2 pada *learning rate* percobaan keempat, hasil pengujian menggunakan *batch size* 32 pada ResNet-50 dan Inception V3 dengan menggunakan *Confusion Matrix* secara berurutan dapat dilihat pada Gambar 4.21 dan Gambar 4.22 sedangkan untuk hasil clasification report dapat dilihat pada Tabel 4.15 dan Tabel 4.16

Selanjutnya, pada percobaan ketiga akan dilakukan dengan menjalankan model dengan menggunakan *batch size* 64. Hasil pengujian menggunakan *batch size* 64 pada ResNet-50 dan Inception V3 dengan menggunakan *Confusion Matrix* secara berurutan dapat dilihat pada Gambar 4.27 dan Gambar 4.28.





Gambar 4.28 *Confusion Matrix* Batch size 64 pada Inception V3

Dalam *Confusion Matrix* pada Gambar 4.27 dan Gambar 4.28, 0 mewakili penyakit Alga Spot, 1 mewakili *Brown Blight*, 2 mewakili *Gray Blight*, 3 mewakili Daun Sehat, 4 mewakili *helopeltis*, dan 5 mewakili *red spot*. Pada Gambar 4.19, terlihat terdapat 573 gambar yang diklasifikasikan dengan benar dan 14 gambar salah. Pada kelas *algal spot*, 85 gambar diklasifikasikan dengan benar. Pada kelas *brown blight*, 91 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *gray blight*. Pada kelas *gray blight*, 96 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *algal spot* dan 1 gambar salah diklasifikasikan sebagai *brown blight*. Pada kelas *healthy*, 105 gambar diklasifikasikan dengan benar, dan ada 1 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *helopeltis*, 101 gambar

diklasifikasikan dengan benar. Pada kelas *red spot*, 95 gambar diklasifikasikan dengan benar, dan ada 6 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *brown blight*, 2 gambar salah diklasifikasikan sebagai kelas *gray blight*, dan 3 gambar salah diklasifikasikan sebagai kelas *healthy*.

Dari hasil *Confusion Matrix* pada Gambar 4.27, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada clasification report pada Tabel 4.19

Tabel 4.19 Classification Report *Batch size* 64 pada ResNet-50

	Precision	Recall	F1-Score	Support
0	0,9734	0,9922	0,9827	86
1	0,9749	0,9645	0,9697	94
2	0,9441	0,9663	0,9551	99
3	0,9692	0,9906	0,9798	106
4	0,9967	1,0000	0,9984	101
5	0,9930	0,9373	0,9643	101
akurasi			0,9750	587
macro avg	0,9752	0,9752	0,9750	587
weighted avg	0,9753	0,9750	0,9750	587

Berdasarkan Tabel 4.19, penggunaan *batch size* 64 pada model ResNet-50 memperoleh rata-rata akurasi 97,50%, rata-rata presisi 97,52%, rata-rata recall 97,52%, dan rata-rata F1 Score 97,50%.

Selanjutnya, pada Gambar 4.28 *Confusion Matrix Batch size* 64 pada Inception V3, 560 gambar yang diklasifikasikan dengan benar dan 27 gambar salah. Pada kelas *algae spot*, terdapat 84 gambar diklasifikasikan dengan benar, dan ada 2 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas

brown blight, terdapat 89 gambar diklasifikasikan dengan benar, dan ada 6 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *gray blight*, 2 gambar salah diklasifikasikan sebagai kelas *helopeltis*, dan 1 gambar salah diklasifikasikan sebagai *red spot*. Pada kelas *gray blight*, terdapat 94 gambar diklasifikasikan dengan benar, dan ada 5 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *brown blight*, 2 gambar salah diklasifikasikan sebagai kelas *helopeltis*, dan 1 gambar lagi diklasifikasikan sebagai kelas *red spot*. Pada kelas *healthy*, 101 gambar diklasifikasikan dengan benar, dan ada 5 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *helopeltis* dan 3 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *helopeltis*, 95 gambar diklasifikasikan dengan benar, dan ada 5 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *brown blight*, 3 gambar salah diklasifikasikan sebagai kelas *gray blight*, 1 gambar salah diklasifikasikan sebagai kelas *healthy* dan 1 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *red spot*, 97 gambar diklasifikasikan dengan benar, dan ada 4 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *brown blight* dan 3 gambar salah diklasifikasikan sebagai kelas *healthy*.

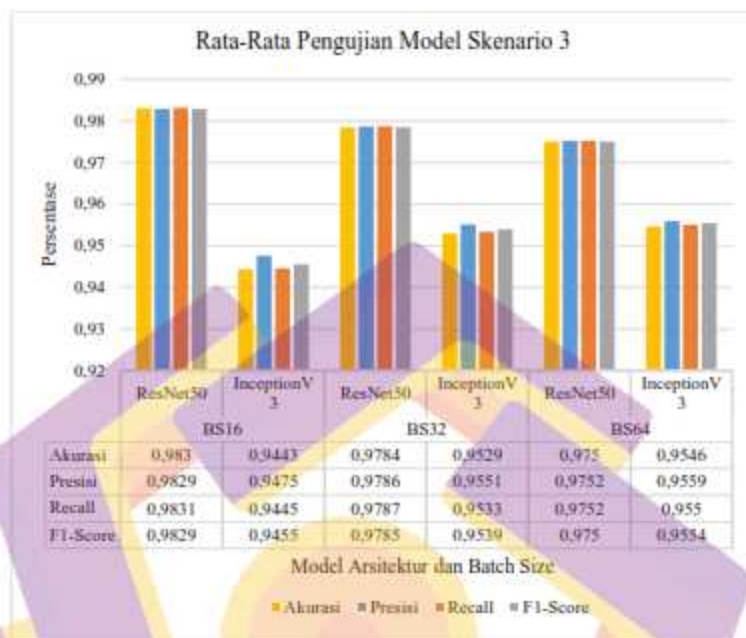
Dari hasil *Confusion Matrix* pada Gambar 4.28, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada classification report pada Tabel 4.20.

Tabel 4.20 Clasification Report *Batch size* 64 Inception V3

	Precision	Recall	F1-Score	Support
0	0,9960	0,9767	0,9863	86
1	0,9603	0,9433	0,9517	94
2	0,9309	0,9529	0,9418	99
3	0,9441	0,9560	0,9500	106
4	0,9564	0,9406	0,9484	101
5	0,9479	0,9604	0,9541	101
akurasi			0,9546	587
macro avg	0,9559	0,9550	0,9554	587
weighted avg	0,9548	0,9546	0,9546	587

Berdasarkan Tabel 4.20, penggunaan *batch size* 64 pada Inception V3 memperoleh rata-rata akurasi 95,46%, rata-rata presisi 95,59%, rata-rata recall 95,50%, dan rata-rata F1 Score 95,54%.

Pada percobaan yang dilakukan, diperoleh rata-rata akurasi tertinggi ketika menggunakan *batch size* 16 pada model ResNet-50 dan *batch size* 64 pada model Inception V3. Hasil perbandingan rata-rata pengujian pada penggunaan *optimizer* dapat dilihat pada Gambar 4.29.



Gambar 4.29 Rata Rata Pengujian Model Skenario 2 *Batch size*

Dapat dilihat pada Gambar 4.29 bahwa kedua model memiliki hasil optimal dengan nilai *batch size* yang berbeda. ResNet-50 memperoleh hasil optimal menggunakan nilai *batch size* 16 dengan rata-rata akurasi 98,30%, rata-rata presisi 98,29%, rata-rata recall 98,31%, dan rata-rata F1-Score 98,29%. Sedangkan pada Inception V3 diperoleh nilai optimal pada *batch size* 64 dengan rata-rata akurasi 95,46%, rata-rata presisi 95,59%, rata-rata recall 95,50%, dan rata-rata F1-Score 95,54%. Perbedaan hasil yang optimal pada penggunaan *batch size* ini terjadi karena adanya perbedaan kompleksitas yang cukup signifikan dari arsitektur yang digunakan. Pada ResNet-50, arsitekturnya cenderung lebih dalam dan memiliki lebih banyak parameter dibandingkan dengan Inception V3 yang memiliki modul

Inception untuk mempercepat dan meningkatkan kinerja jaringan. Oleh karena itu, ResNet-50 mungkin membutuhkan *batch size* yang lebih kecil untuk mengelola kompleksitas modelnya dibanding dengan Inception V3.

Selain melakukan perbandingan dengan menggunakan akurasi, presisi, recall, dan F1-Score, penelitian ini juga memperhatikan *time consumption per step* ketika proses pengujian dan memori yang digunakan secara keseluruhan. Gambar 4.30 merupakan perbandingan *time consumption perstep* dan penggunaan memori.



Gambar 4.30 Perbandingan Rata-Rata Waktu dan Memori Skenario 2 *Batch size*

Dapat dilihat pada Gambar 4.30, ResNet-50 memiliki *time consumption* perstep yang lebih tinggi dibanding Inception V3. Pada skenario tiga ini, *time consumption* mengalami kenaikan ketika menggunakan nilai *batch size* yang semakin tinggi. Hal ini dikarenakan ukuran batch yang lebih besar cenderung membutuhkan lebih banyak waktu komputasi. Pada perbandingan memori yang

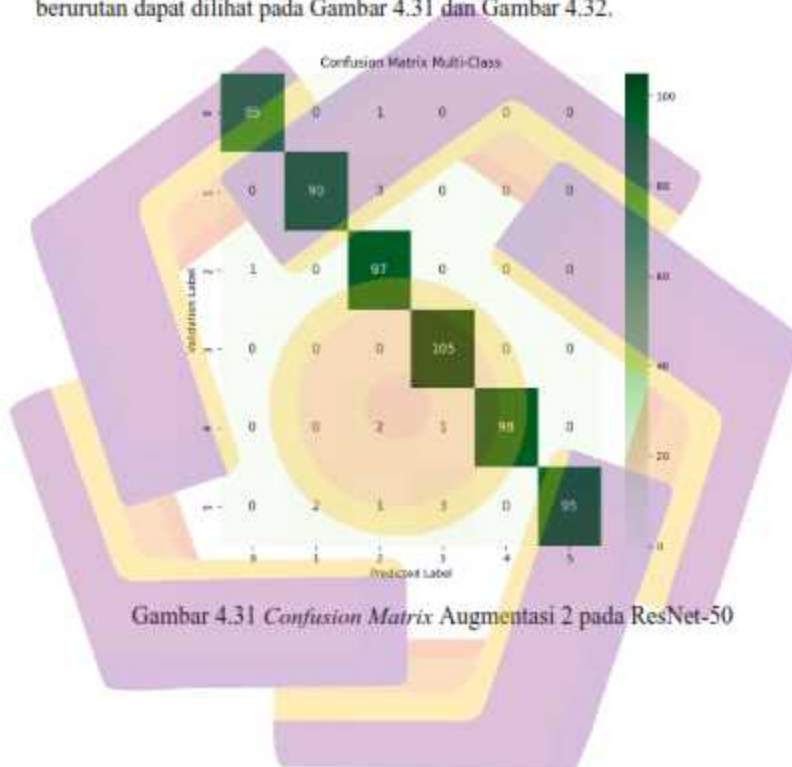
diperlukan, terlihat bahwa pada semua percobaan *batch size* ResNet-50 menghabiskan memori yang lebih sedikit dibanding dengan Inception V3. Perbedaan penggunaan memori ini dipengaruhi oleh arsitektur yang digunakan karena adanya perbedaan jumlah layer. Dengan mempertimbangkan dan memahami waktu dan memori yang digunakan ini, peneliti dapat memilih pendekatan yang paling efisien dan relevan dengan kebutuhan untuk menyelesaikan tugas komputasi dengan efisien.

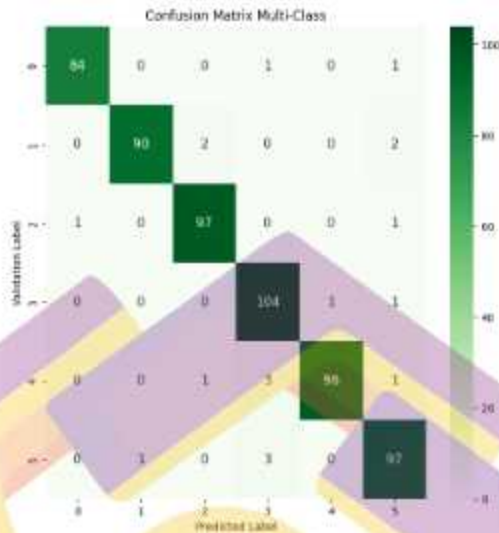
4.5.2.3 Pengujian Skenario 3

Pada skenario ketiga, dilakukan percobaan dengan menggunakan tiga skenario augmentasi seperti yang sudah dijelaskan pada sub bab 4.4 untuk setiap model. Pada pengujian ini, parameter yang digunakan pada kedua model arsitektur mengalami perbedaan pada *batch size* karena hasil pengujian sebelumnya diperoleh rata-rata akurasi optimal pada kondisi yang berbeda. Pada skenario 3 ini, ResNet-50 akan menggunakan *batch size* 16 dan Inception V3 akan menggunakan *batch size* 64.

Percobaan pertama akan dilakukan dengan menjalankan model dengan augmentasi zoom, rotasi, horizontal flip, dan vertical flip pada kedua model. Karena parameter yang digunakan memiliki kesamaan pada skenario 2 *batch size*, hasil pengujian menggunakan teknik augmentasi yang pertama pada ResNet-50 *batch size* 16 dan Inception V3 *batch size* 64 dengan menggunakan *Confusion Matrix* secara berurutan dapat dilihat pada Gambar 4.25 dan Gambar 4.28 sedangkan untuk hasil clasification report dapat dilihat pada Tabel 4.17 dan Tabel 4.20

Selanjutnya, pada percobaan kedua akan dilakukan dengan menjalankan model dengan menggunakan kombinasi teknik augmentasi yang kedua, yaitu zoom, rotasi, dan shear. Hasil pengujian menggunakan kombinasi augmentasi yang kedua pada ResNet-50 dan Inception V3 dengan menggunakan *Confusion Matrix* secara berurutan dapat dilihat pada Gambar 4.31 dan Gambar 4.32.





Gambar 4.32 *Confusion Matrix* Augmentasi 2 pada Inception V3

Dalam *Confusion Matrix* pada Gambar 4.31 dan Gambar 4.32, 0 mewakili penyakit *Alga Spot*, 1 mewakili *Brown Blight*, 2 mewakili *Gray Blight*, 3 mewakili Daun Sehat, 4 mewakili *helopeltis*, dan 5 mewakili *red spot*. Pada Gambar 4.31 terlihat terdapat 570 gambar yang diklasifikasikan dengan benar dan 17 gambar salah. Pada kelas *algal spot*, 85 gambar diklasifikasikan dengan benar dan ada 1 kesalahan dimana gambar salah diklasifikasikan sebagai kelas *gray blight*. Pada kelas *brown blight*, 90 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *gray blight*. Pada kelas *gray blight*, 97 gambar diklasifikasikan dengan benar, dan ada 1 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *algal spot*. Pada kelas *healthy*, 105 gambar diklasifikasikan dengan benar. Pada kelas *helopeltis*, 98 gambar diklasifikasikan dengan benar, dan ada 3 kesalahan di mana 2 gambar salah

diklasifikasikan sebagai kelas *gray blight* dan 1 gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *red spot*, 95 gambar diklasifikasikan dengan benar, dan ada 6 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *brown blight*, 1 gambar salah diklasifikasikan sebagai kelas *gray blight*, dan 3 gambar salah diklasifikasikan sebagai kelas *healthy*.

Dari hasil *Confusion Matrix* pada Gambar 4.31 Gambar 4.17, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada *classification report* pada Tabel 4.21.

Tabel 4.21 Classification Report Augmentasi 2 pada ResNet-50

	Precision	Recall	F1-Score	Support
0	0,9808	0,9922	0,9865	86
1	0,9748	0,9610	0,9679	94
2	0,9389	0,9832	0,9605	99
3	0,9605	0,9937	0,9768	106
4	0,9966	0,9703	0,9833	101
5	0,9965	0,9439	0,9695	101
akurasi			0,9739	587
macro avg	0,9747	0,9741	0,9741	587
weighted avg	0,9745	0,9739	0,9739	587

Berdasarkan Tabel 4.21, penggunaan kombinasi augmentasi 2 pada ResNet-50 memperoleh rata-rata akurasi 97,39%, rata-rata presisi 97,47%, rata-rata recall 97,41%, dan rata-rata F1 Score 97,41%.

Selanjutnya, pada Gambar 4.32 *Confusion Matrix* Augmentasi 2 pada Inception V3, terdapat 568 gambar yang diklasifikasikan dengan benar dan 19 gambar salah. Pada kelas *algal spot*, terdapat 84 gambar diklasifikasikan dengan

benar, dan ada 2 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *healthy* dan 1 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *brown blight*, terdapat 90 gambar diklasifikasikan dengan benar, dan ada 4 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *gray blight* dan 2 gambar salah diklasifikasikan sebagai *red spot*. Pada kelas *gray blight*, terdapat 97 gambar diklasifikasikan dengan benar, dan ada 2 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *algae spot* dan 1 gambar lagi diklasifikasikan sebagai kelas *red spot*. Pada kelas *healthy*, 104 gambar diklasifikasikan dengan benar, dan ada 2 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *helopeltis* dan 1 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *helopeltis*, 96 gambar diklasifikasikan dengan benar, dan ada 4 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *gray blight*, 3 gambar salah diklasifikasikan sebagai kelas *healthy*, dan 1 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *red spot*, 97 gambar diklasifikasikan dengan benar, dan ada 4 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *brown blight*, dan 3 gambar salah diklasifikasikan sebagai kelas *healthy*.

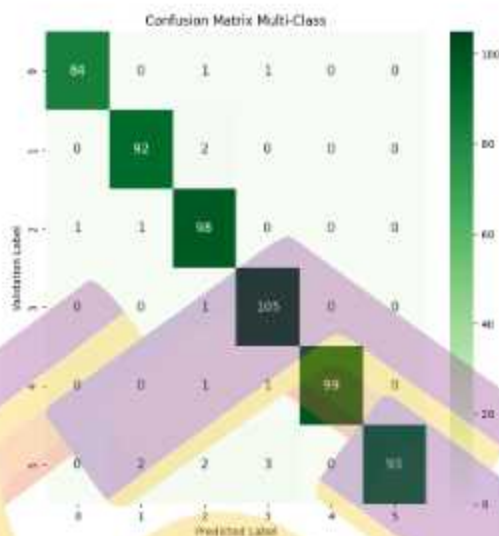
Dari hasil *Confusion Matrix* pada Gambar 4.32 Gambar 4.8, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada classification report pada Tabel 4.22

Tabel 4.22 Classification Report Augmentasi 2 Inception V3

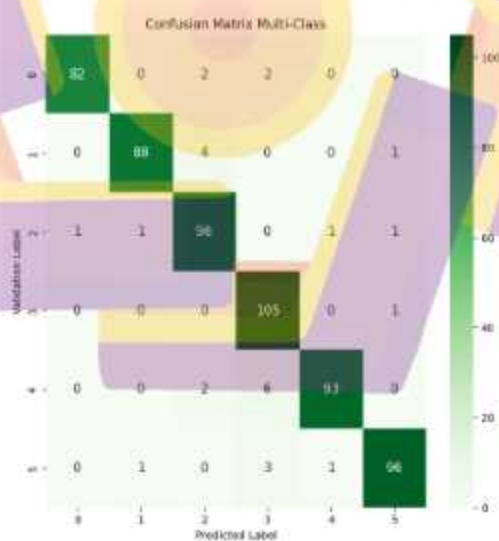
	Precision	Recall	F1-Score	Support
0	0,9921	0,9767	0,9844	86
1	0,9890	0,9574	0,9730	94
2	0,9636	0,9798	0,9716	99
3	0,9341	0,9811	0,9571	106
4	0,9863	0,9505	0,9681	101
5	0,9477	0,9571	0,9524	101
akurasi			0,9671	587
macro avg	0,9688	0,9671	0,9677	587
weighted avg	0,9677	0,9671	0,9672	587

Berdasarkan Tabel 4.22, penggunaan kombinasi augmentasi kedua pada model Inception V3 memperoleh rata-rata akurasi 96,71%, rata-rata presisi 96,88%, rata-rata recall 96,71%, dan rata-rata F1 Score 96,77%.

Selanjutnya, pada percobaan ketiga akan dilakukan dengan menjalankan model dengan menggunakan kombinasi augmentasi 3 yaitu zoom, rotasi, widht shift, dan weight shift. Hasil pengujian menggunakan kombinasi augmentasi 2 pada ResNet-50 dan Inception V3 dengan menggunakan *Confusion Matrix* secara berurutan dapat dilihat pada Gambar 4.33 dan Gambar 4.34.



Gambar 4.33 *Confusion Matrix* Augmentasi 3 pada ResNet-50



Gambar 4.34 *Confusion Matrix* Augmentasi 3 pada Inception V3

Dalam *Confusion Matrix* pada Gambar 4.33 dan Gambar 4.34, 0 mewakili penyakit Alga Spot, 1 mewakili *Brown Blight*, 2 mewakili *Gray Blight*, 3 mewakili Daun Sehat, 4 mewakili *helopeltis*, dan 5 mewakili *red spot*. Pada Gambar 4.19, terlihat terdapat 571 gambar yang diklasifikasikan dengan benar dan 16 gambar salah. Pada kelas *algal spot*, 84 gambar diklasifikasikan dengan benar dan terdapat 2 kesalahan dimana gambar salah diklasifikasikan sebagai kelas *gray blight* dan kelas *healthy*. Pada kelas *brown blight*, 92 gambar diklasifikasikan dengan benar, dan ada 2 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *gray blight*. Pada kelas *gray blight*, 98 gambar diklasifikasikan dengan benar, dan ada 2 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *algal spot* dan kelas *brown blight*. Pada kelas *healthy*, 105 gambar diklasifikasikan dengan benar, dan ada 1 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *gray blight*. Pada kelas *helopeltis*, 99 gambar diklasifikasikan dengan benar, dan ada 2 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *gray blight* dan kelas *healthy*. Pada kelas *red spot*, 93 gambar diklasifikasikan dengan benar, dan ada 7 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *brown blight*, 2 gambar salah diklasifikasikan sebagai kelas *gray blight*, dan 3 gambar salah diklasifikasikan sebagai kelas *healthy*.

Dari hasil *Confusion Matrix* pada Gambar 4.33, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada classification report pada Tabel 4.23.

Tabel 4.23 Classification Report Augmentasi 3 pada ResNet-50

	Precision	Recall	F1-Score	Support
0	0,9883	0,9806	0,9844	86
1	0,9717	0,9752	0,9735	94
2	0,9272	0,9865	0,9560	99
3	0,9573	0,9874	0,9721	106
4	0,9966	0,9802	0,9884	101
5	1,0000	0,9241	0,9605	101
akurasi			0,9722	587
macro avg	0,9735	0,9723	0,9725	587
weighted avg	0,9732	0,9722	0,9722	587

Berdasarkan Tabel 4.23, penggunaan kombinasi augmentasi 3 pada model ResNet-50 memperoleh rata-rata akurasi 97,22%, rata-rata presisi 97,35%, rata-rata recall 97,23%, dan rata-rata F1 Score 97,25%.

Selanjutnya, pada Gambar 4.34 *Confusion Matrix* Augmentasi 3 pada Inception V3, terdapat 560 gambar yang diklasifikasikan dengan benar dan 27 gambar salah. Pada kelas *algal spot*, 82 gambar diklasifikasikan dengan benar, dan ada 4 kesalahan di mana 2 gambar salah diklasifikasikan sebagai kelas *gray blight* dan 2 gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *brown blight*, 88 gambar diklasifikasikan dengan benar, dan ada 5 kesalahan di mana 4 gambar salah diklasifikasikan sebagai kelas *gray blight* dan 1 gambar salah diklasifikasikan sebagai kelas *red spot*. Pada kelas *gray blight*, 96 gambar diklasifikasikan dengan benar, dan ada 4 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *algal spot*, *brown blight*, kelas *helopeltis*, dan kelas *red spot*. Pada kelas *healthy*, 105 gambar diklasifikasikan dengan benar, dan ada 1 kesalahan di mana gambar salah diklasifikasikan sebagai kelas *helopeltis*. Pada kelas *helopeltis*, 93 gambar diklasifikasikan dengan benar, dan ada 8 kesalahan di mana 2 gambar salah

diklasifikasikan sebagai kelas *gray blight* dan 6 gambar salah diklasifikasikan sebagai kelas *healthy*. Pada kelas *red spot*, 96 gambar diklasifikasikan dengan benar, dan ada 5 kesalahan di mana 1 gambar salah diklasifikasikan sebagai kelas *brown blight*, 3 gambar salah diklasifikasikan sebagai kelas *healthy*, dan 1 gambar salah diklasifikasikan sebagai kelas *helopeltis*.

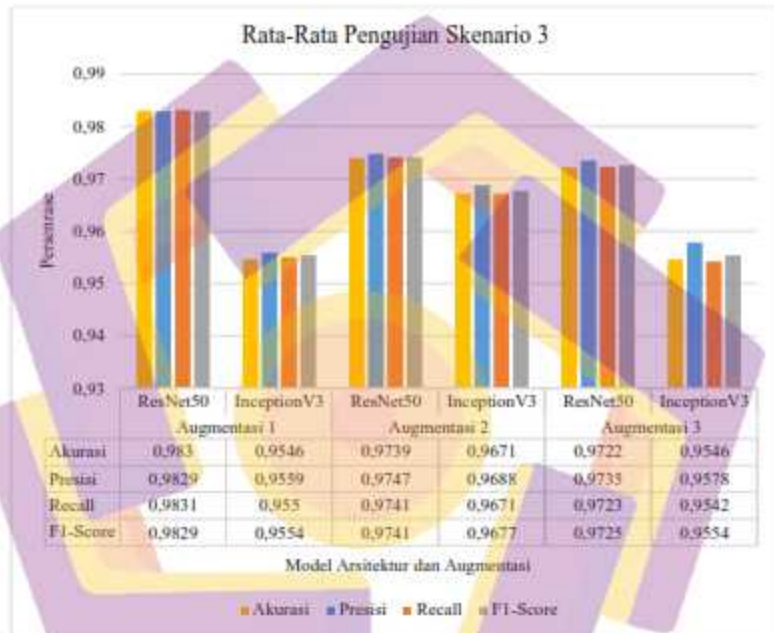
Dari hasil *Confusion Matrix* pada Gambar 4.34, selanjutnya dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, recall, dan F1-Score. Perhitungan dapat dilakukan dengan menggunakan persamaan (1), (2), (3), dan (4). Detail nilai akurasi, presisi, recall, dan f1-score hasil perhitungan *Confusion Matrix* untuk tiap kelas dapat dilihat pada classification report pada Tabel 4.24.

Tabel 4.24 Classification Report Augmentasi 3 Inception V3

	Precision	Recall	F1-Score	Support
0	0,9920	0,9574	0,9744	86
1	0,9815	0,9397	0,9601	94
2	0,9258	0,9663	0,9456	99
3	0,9023	0,9874	0,9429	106
4	0,9790	0,9241	0,9508	101
5	0,9664	0,9505	0,9584	101
akurasi			0,9546	587
macro avg	0,9578	0,9542	0,9554	587
weighted avg	0,9563	0,9546	0,9548	587

Berdasarkan Tabel 4.24, penggunaan kombinasi teknik augmentasi 3 pada model Inception V3 memperoleh rata-rata akurasi 95,46%, rata-rata presisi 95,78%, rata-rata recall 95,42%, dan rata-rata F1 Score 95,54%.

Pada percobaan yang dilakukan, diperoleh rata-rata akurasi tertinggi ketika menggunakan augmentasi 1 pada model ResNet-50 dan augmentasi 2 pada model Inception V3. Hasil perbandingan rata-rata pengujian pada penggunaan *optimizer* dapat dilihat pada Gambar 4.35.



Gambar 4.35 Rata-Rata Akurasi Model Skenario 3

Dapat dilihat pada Gambar 4.35 bahwa kedua model memiliki hasil optimal dengan teknik augmentasi yang berbeda. ResNet-50 memperoleh hasil optimal menggunakan augmentasi berupa rotasi, zoom, vertical flip dan horizontal flip sedangkan Inception V3 memperoleh hasil optimal dengan teknik augmentasi berupa rotasi, zoom, dan shear. Hasil optimal ResNet-50 diperoleh dengan rata-rata akurasi 98,30%, rata-rata presisi 98,29%, rata-rata recall 98,31%, dan rata-rata F1-

Score 98,29%. Sedangkan pada Inception V3 diperoleh rata-rata akurasi 96,71%, rata-rata presisi 96,88%, rata-rata recall 96,71%, dan rata-rata F1-Score 96,77%. Selain melakukan perbandingan dengan menggunakan akurasi, presisi, recall, dan F1-Score, penelitian ini juga memperhatikan *time consumption per step* ketika proses pengujian dan memori yang digunakan secara keseluruhan. Gambar 4.36 merupakan perbandingan *time consumption perstep* dan penggunaan memori.



Gambar 4.36 Perbandingan Rata-Rata Waktu dan Memori Skenario 3

Dapat dilihat pada Gambar 4.36, ResNet-50 memiliki *time consumption* perstep yang lebih tinggi dibanding Inception V3. Pada skenario empat ini, waktu komputasi perstep tidak mengalami banyak perubahan untuk kedua arsitektur yang digunakan. Secara berturut-turut, pada kombinasi augmentasi yang pertama, ResNet-50 dan Inception V3 membutuhkan rata-rata 62ms/step dan 127,3ms/step, pada kombinasi augmentasi yang kedua ResNet-50 dan Inception V3 membutuhkan waktu komputasi dengan rata-rata 60,7ms/step dan 130ms/step, dan terakhir pada

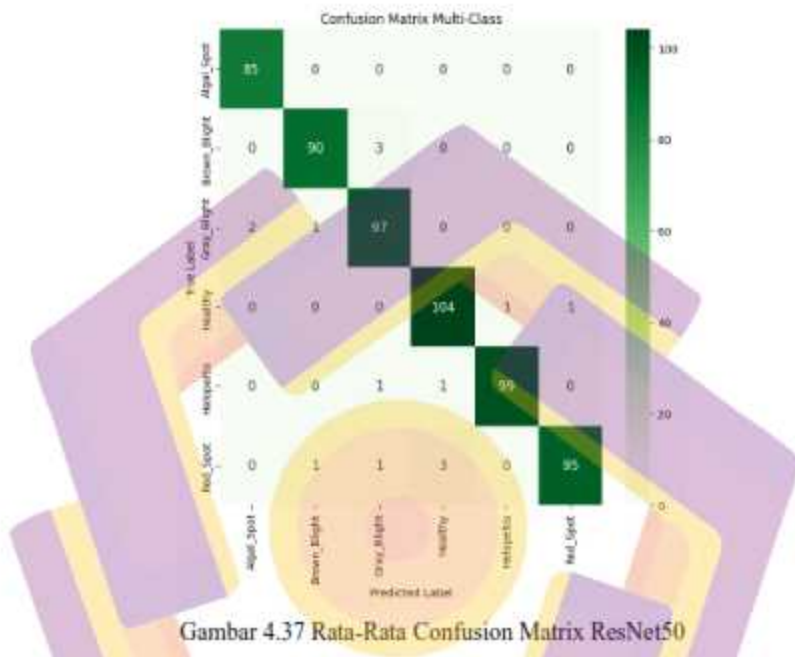
penggunaan kombinasi augmentasi yang ketiga, dibutuhkan waktu komputasi dengan rata-rata 59ms/step untuk ResNet-50 dan 125,3ms/step untuk Inception V3. Pada skenario empat ini, terlihat perbedaan waktu komputasi yang sangat jelas. Perbedaan ini dikarenakan penggunaan nilai *batch size* yang tidak sama antara ResNet-50 dan Inception V3 karena hasil dari pengujian skenario sebelumnya.

Pada perbandingan memori yang diperlukan, terlihat bahwa pada semua percobaan augmentasi, ResNet-50 menghabiskan memori yang lebih sedikit dibanding dengan Inception V3 karena penggunaan *batch size* pada Inception V3 yang lebih besar. Selain itu, perbedaan penggunaan memori ini juga dipengaruhi oleh arsitektur yang digunakan karena adanya perbedaan jumlah layer antara dua model yang diusulkan. Dengan mempertimbangkan dan memahami waktu dan memori yang digunakan ini, peneliti dapat memilih pendekatan yang paling efisien dan relevan dengan kebutuhan untuk menyelesaikan tugas komputasi dengan efisien.

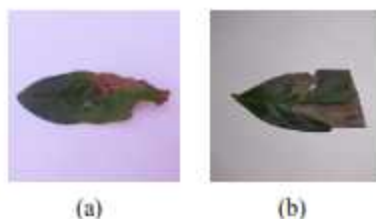
4.5.3 Analisis Kesalahan Hasil *Confusion Matrix*

Setelah melakukan evaluasi kedua model menggunakan *Confusion Matrix*, dapat dilihat berapa banyak gambar yang diklasifikasikan secara benar dan berapa banyak yang diklasifikasikan salah. Berdasarkan Tabel 3.1 Detail Dataset, terlihat bahwa setiap kelas memiliki beberapa kemiripan karakteristik dari segi warna bercak. Sehingga, kesalahan klasifikasi memiliki peluang yang cukup besar karena adanya kemiripan itu. Untuk melihat detail kelas apa saja yang sering salah diklasifikasikan model, dilakukan perhitungan rata-rata hasil confusion matrix dari

skenario 1 hingga skenario 3 agar penilaian menjadi lebih objektif. Gambar 4.37 merupakan rata-rata *confusion matrix* dari ResNet-50.

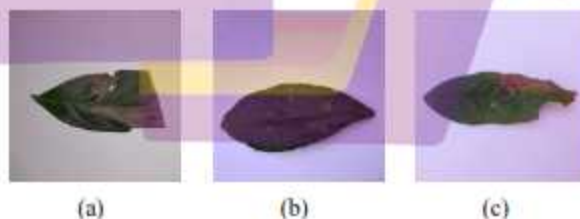


Berdasarkan Gambar 4.37, ResNet-50 memiliki rata-rata dapat mendeteksi kelas *algal spot* dengan benar. Sedangkan pada kelas *brown blight*, terdapat rata-rata 3 gambar yang salah diklasifikasikan sebagai kelas *gray blight*. Hal ini dipengaruhi karena kedua penyakit ini sama-sama memiliki bercak dan warna bercak yang cukup mirip. Untuk melihat lebih detail kemiripan keduanya, Gambar 4.38 merupakan gambar penyakit *brown blight*(a) dan penyakit *gray blight*(b)



Gambar 4.38 Perbedaan Penyakit *Brown Blight*(a) dengan Penyakit *Gray Blight*(b)

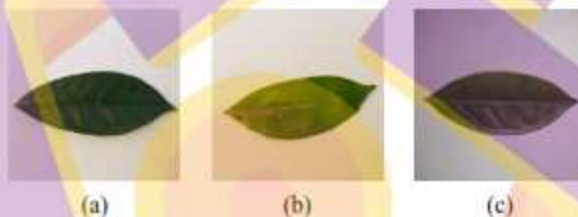
Pada Gambar 4.38, dapat dilihat bahwa keduanya memang sama-sama memiliki bercak yang lebar dan terdapat kemiripan warna. Selanjutnya, pada kelas *gray blight* terdapat 2 gambar salah diklasifikasikan sebagai *algal spot* dan 1 gambar salah diklasifikasikan sebagai *brown blight*. Sama seperti sebelumnya, ketiga penyakit ini memiliki kemiripan pada bercak dan warna pada bercak itu sendiri dimana *gray blight* dengan bercak berwarna keabuan, coklat, merah bahkan hitam sedangkan *algal spot* memiliki bercak berwarna merah kecoklatan, dan *brown blight* memiliki bercak berwarna kecoklatan bahkan hitam. Untuk melihat lebih detail kemiripan dari ketiganya, Gambar 4.39 merupakan gambar penyakit *gray blight*(a), penyakit *algal spot*(b) dan penyakit *brown blight*(c).



Gambar 4.39 Perbedaan Penyakit Gray Blight(a), Algal Spot(b) dan Brown Blight(c)

Pada Gambar 4.39, dapat dilihat bahwa ketiga memang sama-sama memiliki bercak dan terdapat kemiripan warna. Namun, ukuran bercak pada *algal*

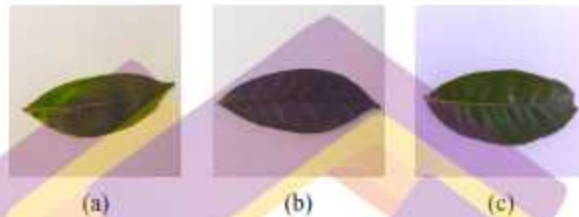
spot cenderung lebih kecil jika dibandingkan dengan *gray blight* dan *brown blight*. Selanjutnya, pada kelas *healthy* terdapat dua gambar yang salah diklasifikasikan sebagai kelas *helopeltis* dan *red spot*. Karakter dari ketiga penyakit ini sebenarnya tidak terlalu mirip. Namun, ketika daun baru terserang penyakit dan belum banyak penyebaran penyakit ke bagian daun yang lain, penyakit *helopeltis* dan *red spot* akan memiliki detail yang cukup kecil sehingga membuat daun akan terlihat seperti daun yang sehat. Untuk melihat lebih detail kemiripan dari ketiganya, Gambar 4.40 merupakan gambar daun *healthy*(a), penyakit *helopeltis*(b) dan penyakit *redspot*(c).



Gambar 4.40 Perbedaan Penyakit *Healthy*(a), *Helopeltis*(b) dan *Red Spot*(c)

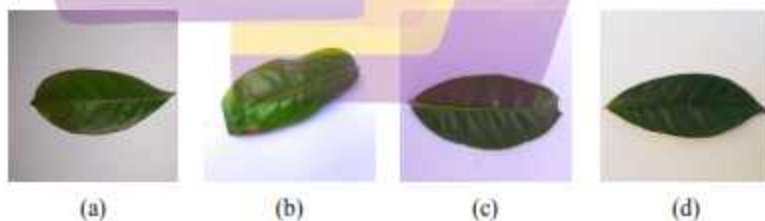
Pada Gambar 4.40, dapat dilihat bahwa ketiga gambar memiliki cukup banyak kemiripan. Pada kelas *helopeltis*, detail yang mengindikasikan daun tersebut berpenyakit berada di area tulang daun sehingga terlihat seperti bukan indikasi penyakit namun bagian dari tulang daun. Pada kelas *red spot*, warna merah pada daun kurang tegas sehingga terlihat seperti daun hijau yang sehat. Selanjutnya, pada kelas *helopeltis* terdapat dua gambar yang salah diklasifikasikan sebagai kelas *healthy* dan *gray blight*. Karakter dari ketiga penyakit ini sebenarnya tidak terlalu mirip. Namun, pada penyakit *gray blight* ketika daun baru terserang penyakit dan belum banyak penyebaran penyakit ke bagian daun yang lain, penyakit ini akan memiliki detail dengan bercak yang lebih kecil sehingga membuat daun akan

terlihat seperti daun yang akan terserang penyakit *helopeltis*. Sedangkan untuk daun *healthy*, tidak ditemukan adanya kemiripan sehingga perlu dilakukan analisis lebih lanjut. Untuk melihat lebih detail dari ketiganya, Gambar 4.41 merupakan gambar daun berpenyakit *helopeltis*(a), daun *healthy*(b) dan penyakit *gray blight*(c).



Gambar 4.41 Perbedaan Penyakit *Helopeltis*(a), *Healthy*(b) dan *Gray Blight*(c)

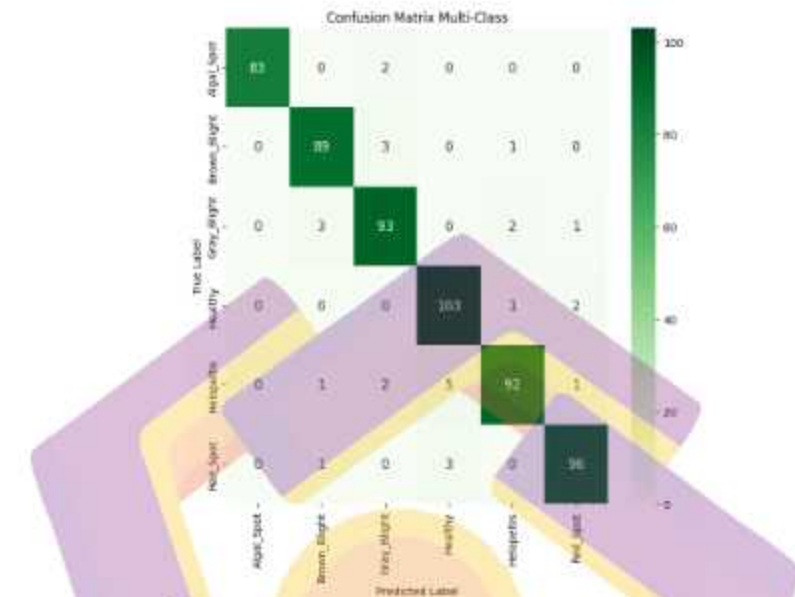
Pada Gambar 4.41, dapat dilihat bahwa *helopeltis* dan *gray blight* memiliki kemiripan, sedangkan pada kelas *healthy* tidak ditemukan sebuah kemiripan. Selanjutnya pada kelas terakhir yaitu *red spot* terdapat lima gambar yang salah diklasifikasikan dengan satu gambar salah diklasifikasikan sebagai *brown blight*, satu gambar salah diklasifikasikan sebagai *gray blight*, dan tiga gambar salah diklasifikasikan sebagai kelas *healthy*. Detail dari keempat kelas tersebut dapat dilihat pada Gambar 4.42.



Gambar 4.42 Perbedaan Penyakit *Red Spot*(a), *Brown Blight*(b), *Gray Blight*(c) dan *Healthy*(d)

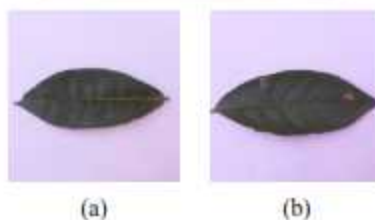
Pada Gambar 4.42, dapat dilihat bahwa memang terdapat kemiripan antara kelas *red spot* dengan *brown blight* dan *gray blight*. Pada sampel yang digunakan, kemungkinan besar daun yang berpenyakit *brown blight* dan *gray blight* merupakan daun yang baru terserang penyakit dan belum banyak penyebaran penyakit ke bagian daun yang lain sehingga terlihat ada kemiripan detail dengan bercak yang lebih kecil sehingga membuat daun akan terlihat seperti daun yang akan terserang penyakit *red spot*. Selain itu, penyakit *red spot* yang baru terserang juga akan memiliki warna merah yang tidak mencolok dan samar, sehingga akan terlihat seperti daun yang sehat.

Setelah melakukan analisis berdasarkan karakteristiknya, ResNet-50 membuat kesalahan paling banyak dalam mengklasifikasikan gambar-gambar pada kelas *red spot*, yaitu sebanyak 5 gambar. Selain itu, terdapat kesalahan sebanyak 3 gambar pada kelas *brown blight* dan *gray blight*, dan 2 gambar pada kelas *healthy* dan *helopeltis*. Kelas *red spot* memiliki 5 kesalahan dalam klasifikasi dikarenakan terdapat 3 gambar yang salah diklasifikasikan sebagai *healthy* karena ketika baru mulai terpapar, daunnya terlihat seperti daun muda. Selanjutnya akan ditampilkan rata-rata hasil *confusion matrix* dari Inception V3. Gambar 4.43 merupakan rata-rata *confusion matrix* dari Inception-V3.



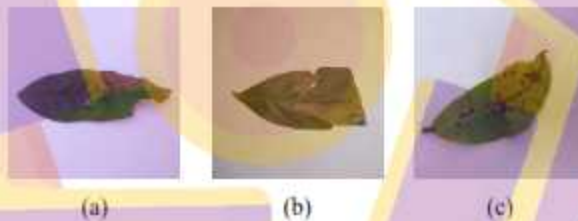
Gambar 4.43 Rata-Rata Confusion Matrix ResNet50

Berdasarkan Gambar 4.43, Inception V3 tidak memiliki kelas yang dideteksi dengan benar secara keseluruhan. Pada kelas *algal spot*, terdapat 2 gambar yang salah diklasifikasikan yaitu gambar salah diklasifikasikan sebagai *gray blight*. Karakter dari penyakit *algal spot* dan *gray blight* memang memiliki sedikit kemiripan dari bercak yang ada pada daun. Gambar 4.44 merupakan detail gambar penyakit *algal spot*(a), dan penyakit *gray blight*(b)



Gambar 4.44 Perbedaan Penyakit *Algal Spot*(a), dan penyakit *Gray Blight*(b)

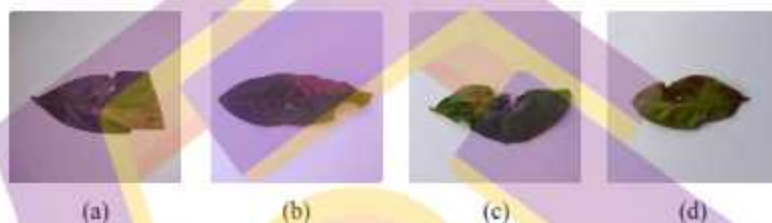
Pada Gambar 4.44, terlihat bahwa *algal spot* dan daun *healthy* ternyata terdapat kemiripan. Daun berpenyakit *algal spot* terlihat seperti daun sehat karena penyakit yang menyerang belum menyebar ke area daun yang lain dan baru terlihat detail yang sangat kecil. Sedangkan untuk *gray blight*, keduanya memang memiliki karakteristik bercak yang mirip yaitu berwarna kecoklatan. Selanjutnya pada kelas *brown blight*, terdapat rata-rata 4 gambar yang salah diklasifikasikan dimana terdapat 3 gambar salah diklasifikasikan sebagai kelas *gray blight* dan 1 gambar salah diklasifikasikan sebagai kelas *helopeltis*. Karakter dari ketiga penyakit tersebut memang memiliki bercak dan warna bercak yang cukup mirip. Untuk melihat lebih detail kemiripan ketiganya, Gambar 4.45 merupakan gambar penyakit *brown blight*(a), *gray blight*(b), dan *helopeltis*(c).



Gambar 4.45 Perbedaan Penyakit *Brown Blight*(a), *Gray Blight*(b), dan *Helopeltis*(c)

Pada Gambar 4.45, dapat dilihat bahwa ketiganya memang sama-sama memiliki bercak dan terdapat kemiripan warna. Selanjutnya, pada kelas *gray blight* terdapat 6 gambar salah diklasifikasikan dimana 3 gambar salah diklasifikasikan sebagai *brown blight*, 2 gambar salah diklasifikasikan sebagai *helopeltis* dan 1 gambar salah diklasifikasikan sebagai *red spot*. Sama seperti sebelumnya, ketiga penyakit ini memiliki kemiripan pada bercak dan warna pada bercak itu sendiri

dimana *gray blight* dengan bercak berwarna keabuan, coklat, merah bahkan hitam sedangkan *brown blight* memiliki bercak berwarna kecoklatan bahkan hitam, *helopeltis* memiliki bercak berwarna kemerahan atau kecoklatan, dan *red spot* memiliki bercak yang cenderung berwarna merah kecokelatan. Untuk melihat lebih detail kemiripan dari keempatnya, Gambar 4.46 merupakan gambar penyakit *gray blight*(a), penyakit *brown blight* (b), *helopeltis*(c) dan *red spot*(d).

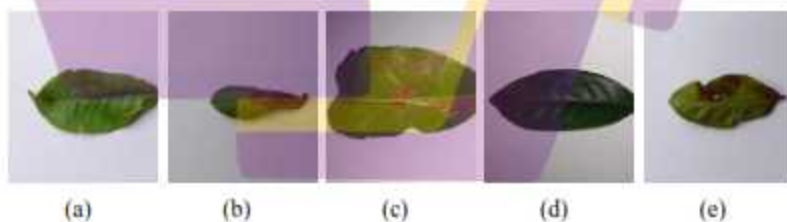


Gambar 4.46 Perbedaan Penyakit *Gray Blight*(a), *Algal Spot*(b) dan *Brown Blight*(c)

Pada Gambar 4.46, dapat dilihat bahwa keempatnya memang memiliki bercak dan terdapat kemiripan warna. Selanjutnya, pada kelas healthy terdapat tiga gambar yang salah diklasifikasikan dimana terdapat 2 gambar yang salah diklasifikasikan sebagai kelas *helopeltis* dan 1 gambar salah diklasifikasikan sebagai kelas *red spot*. Karakter dari ketiga penyakit ini sebenarnya tidak terlalu mirip. Namun, ketika daun baru terserang penyakit dan belum banyak penyebaran penyakit ke bagian daun yang lain, penyakit *helopeltis* dan *red spot* akan memiliki detail yang cukup kecil sehingga membuat daun akan terlihat seperti daun yang sehat. Untuk melihat lebih detail kemiripan dari ketiganya, Gambar 4.40 merupakan gambar daun *healthy*(a), penyakit *helopeltis*(b) dan penyakit *redpot*(c). Pada Gambar 4.40, dapat dilihat bahwa ketiga gambar memiliki cukup banyak

kemiripan. Pada kelas *helopeltis*, detail yang mengindikasikan daun tersebut berpenyakit berada di area tulang daun sehingga terlihat seperti bukan indikasi penyakit namun bagian dari tulang daun. Pada kelas *red spot*, warna merah pada daun kurang tegas sehingga terlihat seperti daun hijau yang sehat.

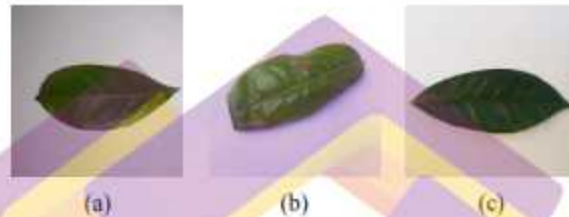
Selanjutnya, pada kelas *helopeltis* terdapat sembilan gambar yang salah diklasifikasikan dimana satu gambar salah diklasifikasikan sebagai kelas *brown blight*, dua gambar salah diklasifikasikan sebagai *gray blight*, lima gambar salah diklasifikasikan sebagai kelas *healthy* dan 1 gambar salah diklasifikasikan sebagai *red spot*. Karakter dari *helopeltis* dengan *brown blight*, *gray blight*, dan *red spot* ini memang ada kemiripan dari karakternya yaitu adanya bercak. Namun, pada daun *healthy* tidak ditemukan sebuah kemiripan. Sedangkan untuk daun *healthy*, tidak ditemukan adanya kemiripan sehingga perlu dilakukan analisis lebih lanjut. Untuk melihat lebih detail dari kelimanya, Gambar 4.47 merupakan gambar daun *helopeltis*(a), penyakit *brown blight*(b), penyakit *gray blight*(c), daun *healthy*(d) dan penyakit *red spot*(e).



Gambar 4.47 Perbedaan Penyakit *Helopeltis*(a), *Healthy*(b) dan *Gray Blight*(c)

Pada Gambar 4.47, dapat dilihat bahwa *helopeltis* dengan *brown blight*, *gray blight* dan *red spot* memiliki kemiripan, sedangkan pada kelas *healthy* tidak ditemukan sebuah kemiripan. Selanjutnya pada kelas terakhir yaitu *red spot*

terdapat empat gambar yang salah diklasifikasikan dengan satu gambar salah diklasifikasikan sebagai *brown blight*, dan tiga gambar salah diklasifikasikan sebagai kelas *healthy*. Detail dari keempat kelas tersebut dapat dilihat pada Gambar 4.42.



Gambar 4.48 Perbedaan Penyakit *Red Spot*(a), *Brown Blight*(b), dan *Healthy*(c)

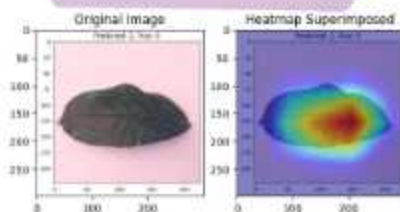
Pada Gambar 4.42, dapat dilihat bahwa memang terdapat kemiripan antara kelas *red spot* dengan *brown blight*. Pada sampel yang digunakan, kemungkinan besar daun yang berpenyakit *brown blight* merupakan daun yang baru terserang penyakit dan belum banyak penyebaran penyakit ke bagian daun yang lain sehingga terlihat ada kemiripan detail dengan bercak yang lebih kecil sehingga membuat daun akan terlihat seperti daun yang akan terserang penyakit *red spot*. Selain itu, penyakit *red spot* yang baru terserang juga akan memiliki warna merah yang tidak mencolok dan samar, sehingga akan terlihat seperti daun yang sehat.

Setelah melakukan analisis berdasarkan karakteristiknya, Inception V3 membuat kesalahan paling banyak dalam mengklasifikasikan gambar-gambar pada kelas *helopeltis*, yaitu sebanyak 9 gambar. Setelah itu, terdapat kesalahan sebanyak 6 gambar pada *gray blight*, 4 gambar pada kelas *brown blight* dan *red spot*, dan terakhir 3 gambar pada kelas *algal spot* dan *healthy*. Helopeltis yang memiliki kesalahan hingga 9 gambar disebabkan oleh didominasinya 5 gambar daun *healthy*

yang terdeteksi sebagai *helopeltis*. Hal ini dikarenakan daun yang terserang *helopeltis* diawal, memiliki bentuk daun yang terlihat masih sehat walaupun sebenarnya sudah terinfeksi.

Hasil analisis dari confusion matrix tersebut hanya berupa perkiraan berdasarkan dari adanya kemiripan pada karakteristik. Selain menganalisis berdasarkan karakter dari setiap penyakit yang menyerang daun teh, penggunaan Grad-CAM dapat menjadi salah satu cara yang tepat untuk menganalisis hasil klasifikasi yang salah. Melalui visualisasi yang dihasilkan, akan lebih mudah memahami faktor apa yang mempengaruhi model dalam menghasilkan prediksi yang salah selain dari adanya kemiripan dari karakteristik gambar itu sendiri.

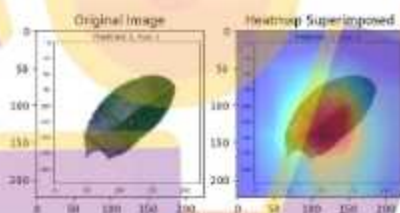
Berdasarkan hasil *Confusion Matrix* pada percobaan yang telah dilakukan, kelas *algal spot* pada ResNet50 secara garis besar dapat mendeteksi secara benar, sedangkan pada Inception V3, algal spot sering kali salah diklasifikasikan sebagai kelas *gray blight* dan *healthy*. Kemiripan ini sudah dibahas sebelumnya pada Gambar 4.44. Namun untuk memberikan wawasan yang lebih mendalam tentang hasil klasifikasi yang salah pada Inception V3, dilakukan analisis menggunakan teknik Grad-CAM. Gambar 4.49 merupakan implementasi Grad-CAM pada Inception V3.



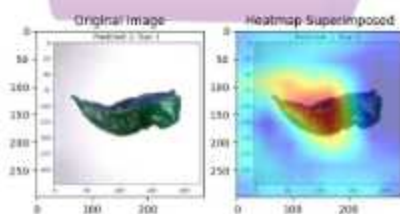
Gambar 4.49 Kesalahan Inception V3 Memprediksi Kelas *Algal spot* Sebagai *Gray blight*

Visualisasi hasil Grad-CAM pada Gambar 4.49 secara jelas *heatmap* menyoroti area yang mempengaruhi keputusan model, khususnya pada detail gambar yang mencerminkan jenis penyakit tersebut. Sehingga, dapat dikatakan bahwa kesalahan klasifikasi yang terjadi memang disebabkan oleh kemiripan warna bercak yang dimiliki antara kedua kelas.

Selanjutnya pada kelas *brown blight*. Berdasarkan hasil *confusion matrix* pada percobaan yang telah dilakukan, sering kali kelas *brown blight* salah diklasifikasikan sebagai kelas *gray blight*. Seperti yang sudah dijelaskan sebelumnya pada Gambar 4.38 atau Gambar 4.45, penyakit *brown blight* dan *gray blight* memang memiliki kemiripan. Gambar 4.50 merupakan implementasi Grad-CAM pada model ResNet-50, sedangkan Gambar 4.51 merupakan implementasi Grad-CAM pada model Inception V3.



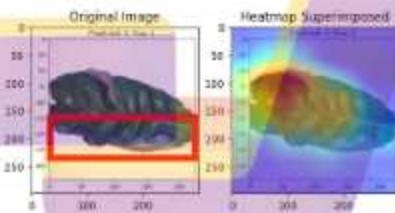
Gambar 4.50 Kesalahan ResNet-50 Memprediksi *Brown blight* Sebagai *Gray blight*



Gambar 4.51 Kesalahan Inception V3 Memprediksi *Brown blight* Sebagai *Gray blight*

Visualisasi hasil Grad-CAM pada Gambar 4.50 dan Gambar 4.51 secara jelas menyoroti area yang mempengaruhi keputusan model, khususnya pada detail gambar yang mencerminkan jenis penyakit tersebut. Selain melihat visualisasi hasil Grad-CAM, perlu disoroti pula gambar asli yang digunakan. Sama seperti sebelumnya, kelas *brown blight* dan *gray blight* memiliki perbedaan yang signifikan pada warna bercak pada daun. Gambar 4.50 dan Gambar 4.51, khususnya pada *original image* yang digunakan, terlihat memang bercak daun lebih berwarna keabuan dari pada kecoklatan. Sehingga, dapat dikatakan bahwa kesalahan klasifikasi yang terjadi disebabkan oleh kemiripan warna bercak yang dimiliki antara kedua kelas.

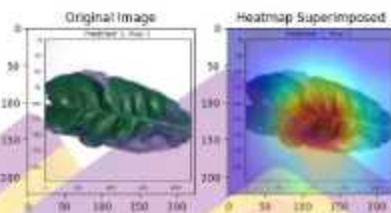
Selain salah diklasifikasikan sebagai kelas *gray blight*, pada Inception V3 kelas *brown blight* juga salah diklasifikasikan sebagai *helopeltis*. Padahal kedua kelas ini tidak memiliki persamaan yang signifikan. Gambar 4.52 merupakan implementasi Grad-CAM pada model Inception V3.



Gambar 4.52 Kesalahan Inception V3 Memprediksi Kelas *Brown blight* Sebagai *Helopeltis*

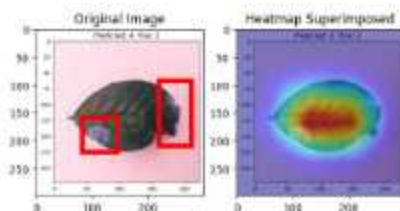
Berdasarkan hasil visualisasi Grad-CAM pada Gambar 4.52, terlihat bahwa model tidak menyoroti area yang seharusnya yang ditandai warna merah, dimana bagian tersebut memperlihatkan bercak coklat yang cukup besar. Namun, model justru menyoroti bagian lain yang ternyata dapat mengindikasikan gambar tersebut

masuk ke dalam kelas *helopeltis*. Selanjutnya, dilakukan perbandingan dengan menggunakan ResNet50 karena ResNet-50 melakukan klasifikasi dengan benar. Gambar 4.53 merupakan implementasi Grad-CAM pada model Inception V3.



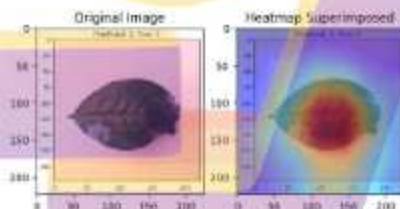
Gambar 4.53 ResNet-50 Memprediksi Kelas *Brown blight* dengan Benar

Berdasarkan hasil visualisasi Grad-CAM pada Gambar 4.53, terlihat bahwa model ResNet-50 menyoroti di area yang berbeda dengan Inception V3. Area yang disoroti oleh ResNet-50 menunjukkan detail dari penyakit brown blight. Selanjutnya pada kelas *gray blight*, berdasarkan hasil *Confusion Matrix* pada percobaan yang telah dilakukan, sering kali kelas *gray blight* salah diklasifikasikan sebagai sebagai kelas *brown blight*. Hal ini memiliki pernyataan yang sama seperti sebelumnya pada kelas *algae spot*. Selain salah diklasifikasikan sebagai kedua kelas tersebut, pada Inception V3 kelas *gray blight* juga salah diklasifikasikan sebagai *helopeltis*. Gambar 4.54 merupakan implementasi Grad-CAM pada model Inception V3.



Gambar 4.54 Kesalahan ResNet-50 Memprediksi Kelas *Gray blight* Sebagai *Helopeltis*

Berdasarkan hasil visualisasi Grad-CAM pada Gambar 4.54, terlihat bahwa model tidak menyoroti area yang seharusnya menjadi detail penyakit seperti pada bagian yang ditandai warna merah. Bahkan warna kuning pada heatmap hanya sedikit mencapai di area yang ditandai warna merah. Model menyoroti bagian lain yang ternyata dapat mengindikasikan gambar tersebut masuk ke dalam kelas *helopeltis*. Selanjutnya, dilakukan perbandingan dengan menggunakan ResNet50 karena ResNet-50 melakukan klasifikasi dengan benar pada gambar tersebut. Gambar 4.55 merupakan implementasi Grad-CAM pada model ResNet-50.

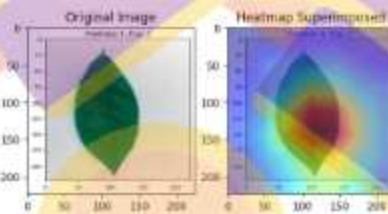


Gambar 4.55 ResNet-50 Memprediksi Kelas *Gray blight* dengan Benar

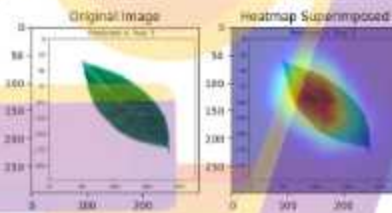
Berdasarkan hasil visualisasi Grad-CAM pada Gambar 4.55, terlihat bahwa model ResNet-50 menyoroti di area yang berbeda dengan Inception V3. Area yang disoroti oleh ResNet-50 sebenarnya juga tidak menunjuk detail dari penyakit brown

blight. Namun, warna kuning pada *heatmap* menjangkau dari detail penyakit pada daun, sehingga tidak terjadi kesalahan dalam melakukan klasifikasi.

Selanjutnya pada kelas *healthy*, berdasarkan hasil *confusion matrix* pada percobaan yang telah dilakukan, sering kali kelas *healthy* pada kedua model salah diklasifikasikan sebagai kelas *helopeltis*. Gambar 4.56 merupakan implementasi Grad-CAM pada model ResNet-50, sedangkan Gambar 4.57 merupakan implementasi Grad-CAM pada model Inception V3.



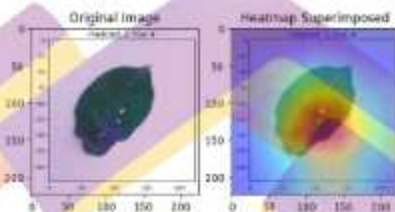
Gambar 4.56 Kesalahan ResNet-50 Memprediksi Kelas *Healthy* Sebagai *Helopeltis*



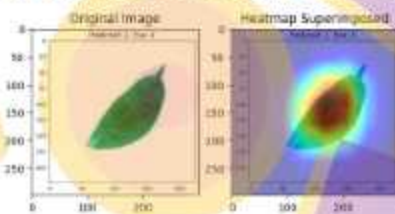
Gambar 4.57 Kesalahan Inception V3 Memprediksi Kelas *Healthy* Sebagai *Helopeltis*

Hasil visualisasi Grad-CAM pada Gambar 4.56 dan Gambar 4.57 terlihat bahwa model sebenarnya telah menyoroti area yang mempengaruhi keputusan model secara jelas. Gambar asli pada dataset juga terlihat tidak memiliki masalah, sehingga kesalahan ini mungkin terjadi ada detail gambar yang terlihat pada model namun tidak terlihat jika dilihat secara langsung oleh mata.

Selanjutnya pada kelas *helopeltis*. Berdasarkan hasil *Confusion Matrix* pada percobaan yang telah dilakukan, sering kali kelas *helopeltis* pada kedua model salah diklasifikasikan sebagai kelas *gray blight*. Gambar 4.58 merupakan implementasi Grad-CAM pada model ResNet-50 sedangkan Gambar 4.59 merupakan implementasi Grad-CAM pada model Inception V3.



Gambar 4.58 Kesalahan ResNet-50 Memprediksi *Helopeltis* Sebagai *Gray blight*

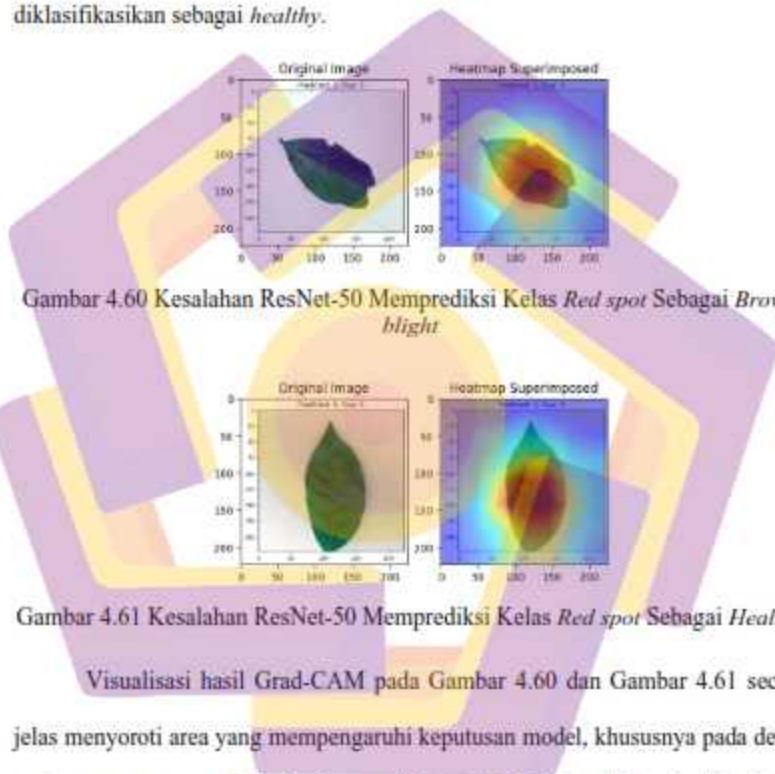


Gambar 4.59 Kesalahan Inception V3 Memprediksi *Helopeltis* Sebagai *Gray blight*

Hasil visualisasi Grad-CAM pada Gambar 4.58 dan Gambar 4.59 terlihat bahwa model sebenarnya telah menyoroti area yang mempengaruhi keputusan model secara jelas. Namun, gambar asli pada dataset memang terlihat memiliki warna abu-abu yang gelap pada bagian yang disoroti. Sehingga, dapat dikatakan bahwa kesalahan klasifikasi yang terjadi disebabkan oleh kemiripan warna bercak yang dimiliki antara kedua kelas pada gambar masukan.

Terakhir pada kelas *red spot*. Berdasarkan hasil *confusion matrix* pada percobaan yang telah dilakukan, sering kali kelas *red spot* pada model ResNet-50

salah diklasifikasikan sebagai kelas *brown blight* dan kelas *healthy*. Gambar 4.60 merupakan implementasi Grad-CAM pada model ResNet-50 kelas *red spot* yang salah diklasifikasikan sebagai *brown blight*, sedangkan Gambar 4.61 merupakan implementasi Grad-CAM pada model ResNet-50 kelas *red spot* yang salah diklasifikasikan sebagai *healthy*.



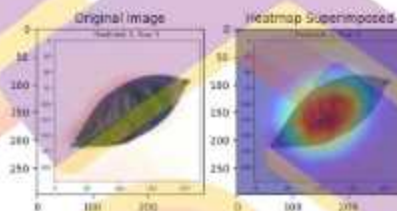
Gambar 4.60 Kesalahan ResNet-50 Memprediksi Kelas *Red spot* Sebagai *Brown blight*

Gambar 4.61 Kesalahan ResNet-50 Memprediksi Kelas *Red spot* Sebagai *Healthy*

Visualisasi hasil Grad-CAM pada Gambar 4.60 dan Gambar 4.61 secara jelas menyoroti area yang mempengaruhi keputusan model, khususnya pada detail gambar yang mencerminkan jenis penyakit tersebut. Selain melihat visualisasi hasil Grad-CAM, perlu disoroti pula gambar asli yang digunakan. Pada Gambar 4.60 khususnya pada *original image* yang digunakan, terlihat memang daun lebih berwarna kecoklatan dibanding berwarna merah. Sehingga, dapat dikatakan bahwa kesalahan klasifikasi yang terjadi disebabkan karena adanya kemiripan antara warna

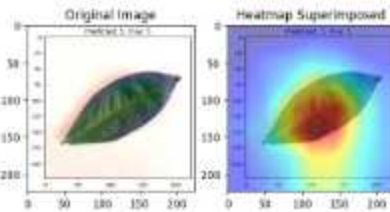
merah dan cokelat ketika gambar diproses pada model. Selanjutnya pada Gambar 4.61, khususnya daun pada *original image* yang digunakan tidak tampak adanya ciri khas dari penyakit *red spot*, sehingga lebih terlihat seperti daun yang sehat.

Selanjutnya, Inception V3 juga seringkali salah mengklasifikasikan kelas *red spot* sebagai kelas *gray blight*. Gambar 4.62 merupakan implementasi Grad-CAM pada model Inception V3.



Gambar 4.62 Kesalahan Inception V3 Memprediksi *Red spot* Sebagai *Gray blight*

Berdasarkan visualisasi hasil Grad-CAM pada Gambar 4.62, model sebenarnya tidak menyoroti area yang mempengaruhi keputusan model secara jelas karena detail penyakit berada di tepi daun namun model menyoroti dibagian tengah namun warna kuning pada heatmap masih menjangkau di bagian tepi daun. Selain melihat visualisasi hasil Grad-CAM, perlu disoroti pula gambar asli yang digunakan. Terlihat pada *original image* yang digunakan memang terlihat lebih berwarna keabu-abuan dibanding berwarna merah. Sehingga, dapat dikatakan bahwa kesalahan klasifikasi yang terjadi disebabkan karena adanya kemiripan antara warna merah dan abu-abu ketika gambar diproses pada model. Selanjutnya, dilakukan perbandingan dengan menggunakan ResNet50 karena ResNet-50 melakukan klasifikasi dengan benar pada gambar tersebut. Gambar 4.63 merupakan implementasi Grad-CAM pada model ResNet-50.



Gambar 4.63 ResNet-50 Memprediksi Kelas *Red Spot* dengan Benar

Berdasarkan hasil visualisasi Grad-CAM pada Gambar 4.63, terlihat bahwa model ResNet-50 menyoroti di area yang berbeda dengan Inception V3. Area yang disoroti oleh ResNet-50 lebih mendekati tepi bagian bawah daun dimana bagian tersebut memang terdapat detail karakteristik dari penyakit tersebut.

Melihat hasil analisis diatas, penyebab kesalahan klasifikasi yang terjadi pada model ResNet-50 dan Inception V3 dipengaruhi oleh tiga hal utama, yaitu kemiripan antara satu kelas dengan kelas yang lain, model yang digunakan tidak menyoroti di area yang seharusnya menjadi detail penyakit pada daun, serta kualitas dari dataset itu sendiri.

4.5.4 Faktor Yang Mempengaruhi Akurasi

Berdasarkan perbandingan hasil dan visualisasi kedua model arsitektur, terdapat beberapa faktor memberikan pengaruh terhadap performa terbaik dari kedua model dalam melakukan klasifikasi penyakit daun teh. Selanjutnya, dilakukan analisis dari akurasi yang diperoleh melalui tiga kali percobaan dari setiap skenario dengan menggunakan nilai mean, standar deviasi, dan p-value.

Nilai mean digunakan untuk memberikan gambaran tentang performa rata-rata kedua model yang digunakan pada setiap skenario yang digunakan, standar deviasi mengukur seberapa stabil atau bervariasinya hasil uji coba yang sudah

dilakukan, dan p-value memberikan indikasi tentang apakah ada perbedaan signifikan di antara hasil-hasil tersebut. Perhitungan mean, standar deviasi, dan p-value dilakukan menggunakan persamaan yang sama pada sub bab 2.3.16. Hasil perhitungan dari nilai mean, standar deviasi, dan p-value dapat dilihat pada Tabel 4.25

Tabel 4.25 Mean, Standar Deviasi dan P-Value

Skenario	Arsitektur	Keterangan	Mean	Standar Deviasi	P-Value
S1	ResNet-50	Augmentasi	0,9710	0,00640	0,0458
		Tanpa Augmentasi	0,9580	0,00080	
	Inception V3	Augmentasi	0,9495	0,00080	0,0009
		Tanpa Augmentasi	0,9057	0,00698	
S2	ResNet-50	Optimizer Adam	0,9710	0,00640	0,0317
		Optimizer SGD	0,9534	0,00424	
	Inception V3	Optimizer Adam	0,9495	0,00080	0,0132
		Optimizer SGD	0,9359	0,00446	
	ResNet-50	LR 0,001	0,9710	0,00640	0,3701
		LR 0,0005	0,9710	0,00640	
		LR 0,00025	0,9761	0,00245	
		LR 0,0001	0,9784	0,00216	
	Inception V3	LR 0,001	0,9495	0,00080	0,1310
		LR 0,0005	0,9483	0,00080	
		LR 0,00025	0,9495	0,00289	
		LR 0,0001	0,9529	0,00160	
	ResNet-50	BS16	0,9830	0,00138	0,0326
		BS32	0,9784	0,00215	
		BS64	0,9750	0,00292	
	Inception V3	BS16	0,9444	0,00424	0,0217
BS32		0,9529	0,00160		
BS64		0,9546	0,00212		
S3	ResNet-50	Augmentasi 1	0,9830	0,00139	0,0157
		Augmentasi 2	0,9739	0,00212	
		Augmentasi 3	0,9722	0,00160	
	Inception V3	Augmentasi 1	0,9546	0,00212	0,0128
		Augmentasi 2	0,9671	0,00212	
		Augmentasi 3	0,9546	0,00525	

Berdasarkan Tabel 4.25, dapat dilihat bahwa pada skenario satu hingga empat, diperoleh nilai standar deviasi yang rendah. Nilai standar deviasi yang rendah ini menunjukkan bahwa model konsisten dalam memprediksi di setiap skenario dengan variasi yang sangat kecil di sekitar rata-rata. Berbeda dengan

standar deviasi, pada perhitungan p-value diperoleh hasil yang berbeda. Pada skenario pertama, model ResNet-50 memperoleh nilai p-value 0,0458. Hal ini menunjukkan bahwa perbedaan antara tanpa penggunaan augmentasi dan penggunaan augmentasi cukup signifikan secara statistik. Namun, karena angka yang dihasilkan sangat mendekati nilai alpha yang ditentukan, sebaiknya perlu dilakukan pengujian lebih banyak. Selanjutnya, pada model Inception-V3, diperoleh nilai p-value 0,0009, hal ini menunjukkan bahwa perbedaan antara tanpa penggunaan augmentasi dan penggunaan augmentasi sangat signifikan secara statistik karena nilai p-value sangat kecil.

Pada skenario kedua bagian optimizer, ResNet-50 memperoleh nilai p-value sebesar 0,0317. Hal ini menunjukkan bahwa perbedaan antara penggunaan *optimizer* Adam dan SGD cukup signifikan secara statistik. Selanjutnya pada model Inception V3, diperoleh nilai p-value 0,0119. Hal ini menunjukkan bahwa penggunaan *optimizer* Adam dan SGD pada Inception V3 juga memberikan perbedaan yang cukup signifikan secara statistik. Hal ini dapat diindikasikan bahwa perbedaan akurasi yang diperoleh pada penggunaan *optimizer* Adam dan SGD pada kedua model terjadi bukan secara kebetulan.

Pada skenario kedua bagian learning rate, kedua model menunjukkan nilai p-value yang tinggi, yaitu 0,3701 pada ResNet-50 dan 0,1986 pada Inception V3. Hal ini menunjukkan bahwa perbedaan pada penggunaan *learning rate* pada ResNet-50 dan Inception V3 tidak cukup signifikan secara statistik. Ini mengindikasikan bahwa akurasi yang diperoleh pada percobaan dengan nilai *learning rate* yang berbeda ini memiliki nilai yang hampir serupa dan perbedaan

yang diamati mungkin terjadi secara kebetulan. Sehingga perlu dilakukan percobaan lebih banyak untuk bisa memastikan perbedaan performa pada kedua *optimizer* tersebut pada model ResNet-50 dan Inception V3.

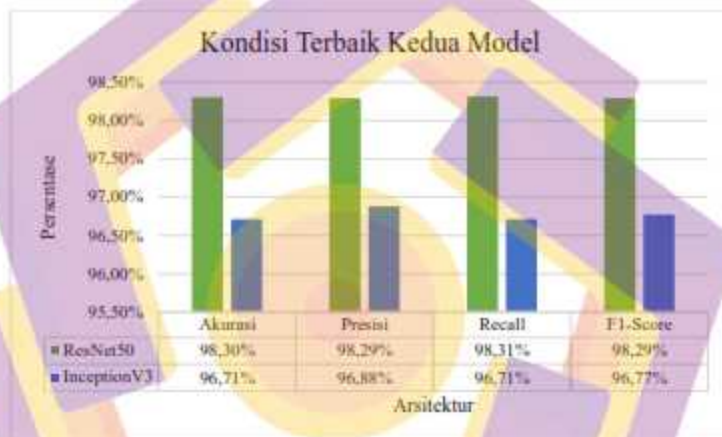
Pada skenario kedua bagian *batch size*, pada model ResNet-50, diperoleh nilai *p-value* sebesar 0,0326 dan pada Inception V3 diperoleh nilai *p-value* sebesar 0,04. Hal ini menunjukkan bahwa penggunaan nilai *batch size* yang berbeda menunjukkan perbedaan yang cukup signifikan secara statistik pada kedua model. Namun, karena nilai *p-value* yang terlalu mendekati nilai 0,05, perlu dilakukan pengujian yang lebih banyak untuk memastikan bahwa akurasi yang diperoleh pada penggunaan nilai *batch size* yang berbeda ini memiliki nilai yang benar-benar berbeda dan bukan terjadi secara kebetulan.

Terakhir pada skenario ketiga, kedua model menunjukkan nilai *p-value* yang rendah, yaitu 0,0157 pada ResNet-50 dan 0,0208 pada Inception V3. Hal ini menunjukkan bahwa penggunaan teknik augmentasi yang berbeda pada skenario ini menunjukkan perbedaan yang cukup signifikan secara statistik. Ini mengindikasikan bahwa akurasi yang dihasilkan pada kombinasi teknik augmentasi yang digunakan tidak terjadi secara kebetulan.

4.6. Kondisi Terbaik Kedua Model

Setelah melakukan semua pengujian skenario 1 hingga skenario 4, diperoleh akurasi terbaik dengan kondisi yang cukup berbeda untuk kedua model yang diusulkan. ResNet-50 memiliki akurasi terbaik sebesar 98,30%, presisi 98,29%, recall 98,31%, dan F1-score 98,29% dengan *optimizer* Adam, *learning rate* 0,0001, *batch size* 16, dan teknik augmentasi yang meliputi zoom, rotasi, horizontal flip dan

vertical flip. Sedangkan Inception V3 memiliki akurasi terbaik sebesar 96,71%, presisi 96,88%, recall 96,71%, dan F1 Score 96,77% dengan *optimizer* Adam, *learning rate* 0,0001, *batch size* 64, dan teknik augmentasi yang digunakan zoom, rotasi, dan shear. Perbedaan pada kedua model dalam memperoleh akurasi terbaik terletak pada penggunaan nilai *batch size* dan teknik augmentasi yang digunakan. Gambar 4.51 merupakan perbandingan pada kondisi terbaik dari kedua model.



Gambar 4.64 Kondisi Terbak Kedua Model

Perbedaan hasil yang didapatkan dapat dikarenakan faktor dari kompleksitas dari masing-masing model, serta pengaturan parameter pada kedua arsitektur model seperti perbedaan penggunaan *optimizer*, *learning rate*, *batch size*, dan teknik augmentasi yang dapat memengaruhi hasil yang didapatkan antara kedua model yang diusulkan.

4.7. Perbandingan dengan Penelitian Sebelumnya

Berdasarkan hasil analisis yang telah dilakukan pada arsitektur ResNet-50 dan Inception V3, selanjutnya akan ditampilkan rangkuman terkait studi literatur

yang ada yang menjadi landasan pada penelitian ini. Rangkuman ini bertujuan untuk membandingkan hasil penelitian ini dengan hasil studi literatur yang digunakan. Perbandingan hasil terhadap studi literatur dapat dilihat pada Tabel 4.26.

Tabel 4.26 Perbandingan Hasil Penelitian

Penulis	Metode	Akurasi	Presisi	Recall	F1-Score
(Hardi, 2022)	NasNetMobile	88%	-	-	-
	MobileNet	95%	-	-	-
(Datta & Gupta, 2023)	CNN	96,56%	96,63%	96,49%	96,50%
Metode yang diusulkan	Inception V3	96,71%	96,88%	96,71%	96,77%
	ResNet-50	98,30%	98,30%	98,31%	98,29%

Berdasarkan hasil perbandingan pada Tabel 4.26, terlihat bahwa setelah dilakukan pengujian, ResNet-50 memiliki nilai rata-rata akurasi, presisi, recall, dan F1-Score yang lebih tinggi dibanding dengan penelitian yang sudah dilakukan sebelumnya oleh (Datta & Gupta, 2023) dan (Hardi, 2022) yaitu 98,30%, 98,30%, 98,31% dan 98,29%.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan dari hasil dan analisis penelitian yang telah dilakukan, dapat disimpulkan bahwa:

1. ResNet-50 dan Inception V3 memiliki performa yang baik dalam melakukan klasifikasi penyakit daun teh. ResNet-50 memperoleh rata-rata akurasi 98,30% dengan menggunakan *optimizer* Adam, *learning rate* 0,0001, *batch size* 16 dan augmentasi berupa rotasi, zoom, dan flip. Sedangkan Inception V3 memperoleh rata-rata akurasi 96,71% dengan menggunakan *optimizer* Adam, *learning rate* 0,0001, *batch size* 64, dan augmentasi berupa rotasi, zoom, dan shear. Sehingga, pada penelitian ini ResNet-50 memiliki akurasi yang lebih tinggi dibanding Inception V3.
2. ResNet-50 memiliki waktu komputasi yang lebih tinggi dibanding Inception V3, namun membutuhkan kapasitas memori yang lebih sedikit dibanding dengan Inception V3 kecuali pada skenario ke tiga karena adanya penggunaan *batch size* yang berbeda.
3. Hasil pengujian pada penelitian ini menemukan bahwa empat faktor yang diujikan dapat mempengaruhi model terhadap nilai akurasi, precision, recall, f1-score. Namun, berdasarkan hasil perhitungan p-value, nilai akurasi pada ResNet-50 dan Inception V3 secara signifikan dipengaruhi oleh *optimizer*, *batch size*, dan teknik augmentasi yang digunakan.

4. Berdasarkan hasil yang diperoleh menggunakan Grad-CAM, kesalahan klasifikasi yang terjadi pada model ResNet-50 dan Inception V3 dipengaruhi oleh tiga hal utama, yaitu kemiripan antara satu kelas dengan kelas yang lain, model yang digunakan tidak menyoroti di area yang seharusnya menjadi detail penyakit pada daun, dan kualitas dari dataset itu sendiri.

5.2. Saran

Berdasarkan analisis hasil percobaan yang dilakukan, terdapat beberapa saran yang dapat digunakan untuk penelitian selanjutnya, yaitu:

1. Menambahkan jumlah dan varian dataset lain seperti penggunaan dataset daun teh yang masih berada di perkebunan perlu dipertimbangkan untuk mengoptimalkan keakuratan dan keandalan model dalam mendeteksi penyakit daun teh di lapangan.
2. Melakukan eksplorasi lebih lanjut dalam melakukan optimasi pada model menggunakan hyperparameter tuning yang lain dan mencoba menerapkan preproesing lain seperti green channel, median filter, dan lain sebagainya.

DAFTAR PUSTAKA

PUSTAKA BUKU

- Elgendy, M. (2020). *Deep Learning for Vision Systems*. Manning Publication.
- Howard, J., & Gugger, S. (2020). *Deep Learning for Coders with Fastai and PyTorch*. O'Reilly Media.
- Navlani, A., Fandango, A., & Idris, I. (2021). *Python Data Analysis : Perform Data Collection, Data Processing, Wrangling, Visualization, and Model Building Using Python*.
- Rahman, S., Ramli, M., Arnia, F., Muharrar, R., Zen, M., & Ikhwan, M. (2021). *Convolutional Neural Networks Untuk Visi Komputer Jaringan Saraf Konvolusional untuk Visi Komputer (Arsitektur Baru, Transfer Learning, Fine Tuning, dan Pruning)*. Deepublish.
- Setiawan, W. (2020a). *Deep Learning Menggunakan Convolutional Neural Network: Teori dan Aplikasi*. Media Nusa Creative.
- Setiawan, W. (2020b). *Topik Khusus Kecerdasan Komputasional: Deep Learning untuk Image dan Speech Recognition*. Media Nusa Creative.
- Shanmugaman, R. (2018). *Deep Learning for Computer Vision*. Packt Publishing.

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

- Ahmed, J. U., Mozahid, M. N., Dhar, A. R., Alamgir, M. S., Jannat, A., & Islam, M. M. (2021). Food Security and Dietary Diversity of Tea Workers of Two Tea Gardens in Greater Sylhet District of Bangladesh. *GeoJournal*, 86(2), 1015–1027. <https://doi.org/10.1007/s10708-019-10108-z>
- Al-gaashani, M. S. A. M., Shang, F., Muthana, M. S. A., Khayyat, M., & El Latif, A. A. A. (2022). Tomato leaf disease classification by exploiting transfer learning and feature concatenation. *ET Image Process.*, 16, 913–925. <https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/ipr2.12397>
- Bao, W., Fan, T., Hu, G., Liang, D., & Li, H. (2022). Detection and identification of tea leaf diseases based on AX-RetinaNet. *Scientific Reports*, 12(1), 1–16. <https://doi.org/10.1038/s41598-022-06181-z>

- Bera, S., & Shrivastava, V. K. (2020). Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification. *International Journal of Remote Sensing*, 41(7), 2664–2683. <https://doi.org/10.1080/01431161.2019.1694725>
- Borman, R. I., Ahmad, I., & Rahmanto, Y. (2022). Klasifikasi Citra Tanaman Perdu Liar Berkhasiat Obat Menggunakan Jaringan Syaraf Tiruan Radial Basis Function. *Bulletin of Informatics and Data Science*, 1(1), 6–13.
- ChandraKarmokar, B., Samawat Ullah, M., Kibria Siddiquee, M., & Md. Rokibul Alam, K. (2015). Tea Leaf Diseases Recognition using Neural Network Ensemble. *International Journal of Computer Applications*, 114(17), 27–30. <https://doi.org/10.5120/20071-1993>
- Chen, J., Liu, Q., & Gao, L. (2019). Visual Tea Leaf Disease Recognition Using A Convolutional Neural Network Model. *Symmetry*, 11(3). <https://doi.org/10.3390/sym11030343>
- Datta, S., & Gupta, N. (2023). A Novel Approach For the Detection of Tea Leaf Disease Using Deep Neural Network. *Procedia Computer Science*, 218, 2273–2286. <https://doi.org/10.1016/j.procs.2023.01.203>
- Dong, N., Zhao, L., Wu, C. H., & Chang, J. F. (2020). Inception v3 based cervical cell classification combined with artificially extracted features. *Applied Soft Computing*, 93, 106311. <https://doi.org/10.1016/j.asoc.2020.106311>
- Fadillah, R. Z., Irawan, A., Susanty, M., & Artikel, I. (2021). Data Augmentasi Untuk Mengatasi Keterbatasan Data Pada Model Penerjemah Bahasa Isyarat Indonesia (BISINDO). *Jurnal Informatika*, 8(2), 208–214. <https://ejournal.bsi.ac.id/ejurnal/index.php/ji/article/view/10768>
- Gahagho, Y. D., Rotinsulu, T. O., & Mandei, D. (2021). Pengaruh Literasi Keuangan Sikap Keuangan Dan Sumber Pendapatan Terhadap Perilaku Pengelolaan Keuangan Mahasiswa Fakultas Ekonomi Dan Bisnis Unsrat Dengan Niat Sebagai Variabel Intervening. *Jurnal EMBA: Jurnal Riset Ekonomi, Manajemen, Bisnis Dan Akuntansi*, 9(1), 543–555. <https://ejournal.unsrat.ac.id/index.php/emba/article/view/32337>
- Gayathri, S., Wise, D. C. J. W., Shamini, P. B., & Muthukumar, N. (2020). Image Analysis and Detection of Tea Leaf Disease using Deep Learning. *Proceedings of the International Conference on Electronics and Sustainable Communication Systems, ICESC 2020, Icesc*, 398–403. <https://doi.org/10.1109/ICESC48915.2020.9155850>
- Hardi, N. (2022). Komparasi Algoritma MobileNet Dan Nasnet Mobile Pada Klasifikasi Penyakit Daun Teh. *Reputasi: Jurnal Rekayasa Perangkat Lunak*, 3(1), 50–55. <https://doi.org/10.31294/reputasi.v3i1.1313>

- Hendrian, S. (2018). Algoritma Klasifikasi Data Mining Untuk Memprediksi Siswa Dalam Memperoleh Bantuan Dana Pendidikan. *Faktor Exacta*, 11(3), 266–274. <https://doi.org/10.30998/faktorexacta.v1i13.2777>
- Hu, G., Yang, X., Zhang, Y., & Wan, M. (2019). Identification of Tea Leaf Diseases by Using an Improved Deep Convolutional Neural Network. *Sustainable Computing: Informatics and Systems*, 24, 100353. <https://doi.org/10.1016/j.suscom.2019.100353>
- Jahagirdar, P., & Budihal, S. V. (2020). Framework to Detect NPK Deficien in Maize Plant Using CNN. *Proceedings of ICACIE*, 2(2).
- Janakiramalah, B., & Kalyani, G. (2021). Dementia Detection Using the Deep Convolutional Neural Network Method. *Elsevier Science*, 157–178. <https://doi.org/https://doi.org/10.1016/B978-0-12-822226-3.00007-6>.
- Julianto, A., Sunyoto, A., & Wibowo, F. W. (2022). Optimasi Hyperparameter Convolutional Neural Network Untuk Klasifikasi Penyakit Tanaman Padi. *TEKNIMEDIA: Teknologi Informasi Dan Multimedia*, 3(2), 98–105. <https://doi.org/10.46764/teknimedia.v3i2.77>
- Kandel, I., & Castelli, M. (2020a). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express*, 6(4), 312–315. <https://doi.org/10.1016/j.ict.2020.04.010>
- Kandel, I., & Castelli, M. (2020b). Transfer learning with convolutional neural networks for diabetic retinopathy image classification. A review. *Applied Sciences (Switzerland)*, 10(6). <https://doi.org/10.3390/app10062021>
- Kim, T. K. (2017). Understanding one-way ANOVA Using Conceptual Figures. *Korean Journal of Anesthesiology*, 70(1), 22. <https://doi.org/10.4097/kjae.2017.70.1.22>
- Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. *ACM Computing Surveys*, 50(6). <https://doi.org/10.1145/3136625>
- Marifatul Azizah, L., Fadillah Umayah, S., & Fajar, F. (2018). Deteksi Kecacatan Permukaan Buah Manggis Menggunakan Metode Deep Learning dengan Konvolusi Multilayer. *Semesta Teknika*, 21(2), 230–236. <https://doi.org/10.18196/st.212229>
- Mathew, M. P., & Mahesh, T. Y. (2022). Leaf-based disease detection in bell pepper plant using YOLO v5. *Signal, Image and Video Processing*, 16(3), 841–847. <https://doi.org/10.1007/s11760-021-02024-y>

- Mayatopani, H., Borman, R. I., Atmojo, W. T., & Arisantoso, A. (2021). Classification of Vehicle Types Using Backpropagation Neural Networks With Metric and Eccentricity Parameters. *Jurnal Riset Informatika*, 4(1), 65–70. <https://doi.org/10.34288/jri.v4i1.293>
- Moujahid, H., Cherradi, B., Al-Sarem, M., Bahatti, L., Eljjaly, A. B. A. M. Y., Alsaeedi, A., & Saeed, F. (2022). Combining CNN and Grad-Cam For Covid-19 Disease Prediction And Visual Explanation. *Intelligent Automation and Soft Computing*, 32(2), 723–745. <https://doi.org/10.32604/iasc.2022.022179>
- Pandian, J. A., Kanchanadevi, K., Kumar, V. D., Jasińska, E., Goño, R., Leonowicz, Z., & Jasiński, M. (2022). A Five Convolutional Layer Deep Convolutional Neural Network for Plant Leaf Disease Detection. *Electronics (Switzerland)*, 11(8). <https://doi.org/10.3390/electronics11081266>
- Pandian, J. A., Nisha, S. N., Kanchanadevi, K., Pandey, A. K., & Rima, S. K. (2023). Grey Blight Disease Detection on Tea Leaves Using Improved Deep Convolutional Neural Network. *Computational Intelligence and Neuroscience*, 2023, 1–11. <https://doi.org/10.1155/2023/7876302>
- Putry, N. M. (2022). Komparasi Algoritma KNN dan Naïve Bayes Untuk Klasifikasi Diagnosis Penyakit Diabetes Mellitus. *EVOLUSI: Jurnal Sains Dan Manajemen*, 10(1). <https://doi.org/10.31294/evolusi.v10i1.12514>
- Ramdan, A., Heryana, A., Arisal, A., Kusumo, R. B. S., & Pardede, H. F. (2020). Transfer Learning and Fine-Tuning for Deep Learning-Based Tea Diseases Detection on Small Datasets. *Proceeding - 2020 International Conference on Radar, Antenna, Microwave, Electronics and Telecommunications, ICRAMET 2020*, 206–211. <https://doi.org/10.1109/ICRAMET51080.2020.9298575>
- Ramdan, A., Suryawati, E., Kusumo, R. B. S., Pardede, H. F., Mahendra, O., Dahlan, R., Fauziah, F., & Syahrian, H. (2019). Deep CNN Based Detection for Tea Clone Identification. *Jurnal Elektronika Dan Telekomunikasi*, 19(2), 45. <https://doi.org/10.14203/jet.v19.45-50>
- Rohim, A., Sari, Y. A., & Tibyani. (2019). Convolution neural network (cnn) untuk pengklasifikasian citra makanan tradisional. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(7), 7038–7042. <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/5851/2789>
- Sanlier, N., Gokcen, B. B., & Altuğ, M. (2018). Tea consumption and disease correlations. *Trends in Food Science and Technology*, 78(April), 95–106. <https://doi.org/10.1016/j.tifs.2018.05.026>
- Sarvamangala, D. R., & Kulkarni, R. V. (2022). Convolutional Neural Networks In Medical Image Understanding: A Survey. *Evolutionary Intelligence*, 15(1), 1–22. <https://doi.org/10.1007/s12065-020-00540-3>

- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision*, 128(2), 336–359. <https://doi.org/10.1007/s11263-019-01228-7>
- Soeb, M. J. A., Jubayer, M. F., Tarin, T. A., Al Mamun, M. R., Ruhad, F. M., Parven, A., Mubarak, N. M., Karri, S. L., & Meftaul, I. M. (2023). Tea Leaf Disease Detection and Identification Based on YOLOV7 (YOLO-T). *Scientific Reports*, 13(1), 6078. <https://doi.org/10.1038/s41598-023-33270-4>
- Stevani, E., Setyaningsih, Y., & Harfiani, E. (2021). Uji Efektivitas Ekstrak Daun Kenikir (*Cosmos caudatus* Kunth) Terhadap Penghambatan Pertumbuhan Jamur *Malassezia furfur*. *Seminar Nasional Riset Kedokteran*, 2, 202–213.
- Suttapakti, U., & Bunpeng, A. (2019). Potato Leaf Disease Classification Based on Distinct Color and Texture Feature Extraction. *Proceedings - 2019 19th International Symposium on Communications and Information Technologies, ISCIT 2019, Mcd*, 82–85. <https://doi.org/10.1109/ISCIT.2019.8905128>
- Wu, B., Liu, Z., Yuan, Z., Sun, G., & Wu, C. (2017). Reducing Overfitting In Deep Convolutional Neural Networks Using Redundancy Regularizer. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10614 LNCS, 49–55. https://doi.org/10.1007/978-3-319-68612-7_6
- Yudono, M. A. S., Hamidi, E. A. Z., Kuspranoto, A. H., & Sidik, A. D. W. M. (2022). Jaringan Syaraf Tiruan Perambatan Balik Untuk Klasifikasi Covid-19 Berbasis Tekstur Menggunakan Orde Pertama Berdasarkan Citra Chest X-Ray. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK)*, 9(4), 799–808. <https://doi.org/10.25126/jtiik.202295663>
- Yutika, C. H., Adiwijaya, A., & Faraby, S. Al. (2021). Analisis Sentimen Berbasis Aspek pada Review Female Daily Menggunakan TF-IDF dan Naïve Bayes. *Jurnal Media Informatika Budidarma*, 5(2), 422. <https://doi.org/10.30865/mib.v5i2.2845>
- Yuwana, R. S., Fauziah, F., Heryana, A., Krisnandi, D., Kusumo, R. B. S., & Pardede, H. F. (2020). Data Augmentation using Adversarial Networks for Tea Diseases Detection. *Jurnal Elektronika Dan Telekomunikasi*, 20(1), 29. <https://doi.org/10.14203/jet.v20.29-35>
- Zhao, Y., Gong, L., Huang, Y., & Liu, C. (2016). A review of key techniques of vision-based control for harvesting robot. *Computers and Electronics in Agriculture*, 127, 311–323. <https://doi.org/10.1016/j.compag.2016.06.022>