

**PENERAPAN CNN UNTUK MENDETEKSI DAN  
MENGKLASIFIKASIKAN PENYAKIT DAUN JAGUNG  
SEBAGAI PENDUKUNG SISTEM PERTANIAN DIGITAL**

**Proposal Tugas Akhir**

Diajukan Untuk Memenuhi  
Persyaratan Guna Meraih Gelar Sarjana  
Informatika Universitas Muhammadiyah Malang



Muhammad Aries Ramadhan  
201810370311076

**Bidang Minat**  
Sains Data

**PROGRAM STUDI INFORMATIKA FAKULTAS TEKNIK  
UNIVERSITAS MUHAMMADIYAH MALANG**

**2022**

**LEMBAR PERSETUJUAN**

**PENERAPAN CNN UNTUK MENDETEKSI DAN  
MENGKLASIFIKASIKAN PENYAKIT DAUN JAGUNG SEBAGAI  
PENDUKUNG SISTEM PERTANIAN DIGITAL**


**Muhammad Aries Ramadhan**

**201810370311076**


Telah Direkomendasikan Untuk Diajukan Sebagai  
Judul Tugas Akhir Di  
Program Studi Informatika Universitas Muhammadiyah Malang

Menyetujui,

Dosen 1

  
**Agus Eko Minarno, S.Kom., M.Kom.**  
**NIP. 108.1410.0540**

Dosen 2

  
**Yufis Azhar, S.Kom., M.Kom.**  
**NIP. 108.1410.0544**

## LEMBAR PENGESAHAN

**Penerapan Cnn Untuk Mendeteksi Dan Mengklasifikasikan Penyakit Daun  
Jagung Sebagai Pendukung Sistem Pertanian Digital**

### TUGAS AKHIR

Sebagai Persyaratan Guna Meraih Gelar Sarjana Strata 1 Informatika Universitas  
Muhammadiyah Malang

Disusun Oleh:

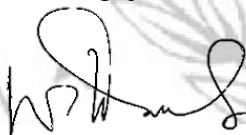
**Muhammad Aries Ramadhan**

**NIM. 201810370311076**

Tugas Akhir ini telah diuji dan dinyatakan lulus melalui sidang majelis penguji  
Pada tanggal 18 Oktober 2022

Menyetujui,

Penguji I



**Wildan Suharso, S.Kom., M.Kom**

**NIP. 108.1703.0596**

Penguji II

**Vinna Rahmayanti S. S.Si., M.Si**

**NIP. 108.3060.71990**

Mengetahui,

Ketua Jurusan Informatika

**Ir. Galih Wasis Wicaksono, S.Kom, M.Cs.**

**NIP. 108.1410.0541**

## LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini:

**Nama : MUHAMMAD ARIES RAMADHAN**

**NIM : 201810370311076**

**FAK/JUR : TEKNIK/INFORMATIKA**

Dengan ini saya menyatakan bahwa Tugas Akhir dengan judul “**PENERAPAN CNN UNTUK MENDETEKSI DAN MENGLASIFIKASIKAN PENYAKIT DAUN JAGUNG SEBAGAI PENDUKUNG SISTEM PERTANIAN DIGITAL**”, beserta seluruh isinya adalah karya saya sendiri dan bukan merupakan karya tulis orang lain, baik sebagian maupun seluruhnya, kecuali dalam bentuk kutipan yang telah disebutkan sumbernya.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya. Apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya ini atau ada klaim dari pihak lain terhadap keaslian karya saya ini maka saya siap menanggung segala bentuk resiko/sanksi yang berlaku.

Mengetahui,

Dosen Pembimbing



**Agus Eko Minarno, S.Kom., M.Kom.**

Malang, 30 September 2022

Yang Membuat Pernyataan



**Muhammad Aries Ramadhan**

## ABSTRAK

Jagung adalah salah satu biji-bijian yang paling banyak dibudidayakan di seluruh dunia. Jagung memiliki banyak manfaat bagi tubuh manusia karena kandungan gizinya yaitu di antara manfaatnya dapat menurunkan hipertensi sehingga penderita penyakit jantung dapat melakukan pencegahan sejak dini. Tanaman jagung sangat rentan terhadap penyakit daun tertentu seperti hawar daun utara, karat biasa, dan cercospora. Pada penelitian ini dilakukan klasifikasi untuk mendeteksi penyakit pada citra daun jagung menggunakan CNN. Data yang terdapat dalam penelitian ini akan dilakukan data preprocessing terlebih dahulu, data preprocessing sangat penting dilakukan untuk menghindari data yang tidak seimbang. Penelitian ini menggunakan augmentasi data untuk menghasilkan model yang lebih baik. Augmentasi data merupakan proses dalam pengolahan data citra. Augmentasi data adalah strategi untuk meningkatkan keragaman data yang tersedia untuk model pelatihan, tanpa benar-benar mengumpulkan data baru. Data yang digunakan dalam penelitian ini diperoleh dari dataset: Penyakit Daun Tanaman, dengan jumlah 3852 yang terbagi dalam 4 kelas, yaitu: healthy, cercospora, common rust dan northern leaf blight. Ekstraksi fitur dan klasifikasi menggunakan convolutional neural network (CNN) menggunakan 11 model, yaitu: AlexNet, VGG16, VGG19, ResNet50, ResNet101, Inception-V3, DenseNet201, Xception, MobileNetV2, Inception ResNet V2, ResNet152 V2. Berdasarkan hasil penelitian, klasifikasi terbaik diperoleh oleh DenseNet201 dengan tingkat akurasi 97%. ResNet152 dengan akurasi 96% dan VGG16 mendapatkan hasil akurasi yang memuaskan sebesar 95%. DenseNet201 dengan data augmentasi mendapatkan tingkat akurasi sebesar 97%. DenseNet201 pada penelitian ini merupakan model yang paling cocok untuk ekstraksi ciri dan klasifikasi citra pada penyakit daun jagung. Dengan melihat hasil akurasi yang telah dilakukan, terbukti bahwa Data Augmentation yang diterapkan pada data latih dapat meningkatkan hasil akurasi untuk kebutuhan klasifikasi citra.

**Kata Kunci:** AlexNet, VGG, ResNet, Inception-V3, DenseNet, Xception, MobileNet, Inception ResNet, Convolutional Neural Network, Transfer Learning, Image Classification, Maize Leaf Image.

## ABSTRACT

Corn is one of the most widely cultivated grains worldwide. Corn has many benefits for the human body because of its nutritional content, namely among the benefits it can reduce hypertension so that people with heart disease can take early prevention. Corn plants are very susceptible to certain leaf diseases such as northern leaf blight, common rust, and cercospora. In this study, classification was carried out to detect disease in corn leaf images using CNN. The data contained in this study will be preprocessed first, data preprocessing is very important to do to avoid unbalanced data. This study uses data augmentation to produce a better model. Data augmentation is a process in image data processing. Data augmentation is a strategy of increasing the diversity of data available for a training model, without actually collecting new data. The data used in this study was obtained from the dataset: Plant Leaf Diseases, with a total of 3852 which is divided into 4 classes, namely: healthy, cercospora, common rust and northern leaf blight. Feature extraction and classification using a convolutional neural network (CNN) using 11 models, namely: AlexNet, VGG16, VGG19, ResNet50, ResNet101, Inception-V3, DenseNet201, Xception, MobileNetV2, Inception ResNet V2, ResNet152 V2. Based on the results of the study, the best classification was obtained by DenseNet201 with an accuracy rate of 97%. ResNet152 with 96% accuracy and VGG16 get satisfactory accuracy results of 95%. DenseNet201 with augmentation data gets an accuracy rate of 97%. DenseNet201 in this study is the most suitable model for feature extraction and image classification in corn leaf disease. By looking at the accuracy results that have been carried out, it is proven that Data Augmentation applied to training data can improve accuracy results for image classification needs.

**Keywords:** AlexNet, VGG, ResNet, Inception-V3, DenseNet, Xception, MobileNet, Inception ResNet, Convolutional Neural Network, Transfer Learning, Image Classification, Maize Leaf Image.

## KATA PENGANTAR

Puji Syukur Alhamdulillah senantiasa Penulis ucapkan atas kehadiat Allah SWT yang telah melimpahkan Rahmat, Taufik serta Hidayah-Nya sehingga pada akhirnya mampu menyelesaikan tugas akhir dengan judul **“Penerapan Cnn Untuk Mendeteksi Dan Mengklasifikasikan Penyakit Daun Jagung Sebagai Pendukung Sistem Pertanian Digital”**, meskipun masih terdapat banyak kekurangan. Shalawat serta salam semoga selalu tercurah kepada junjungan kita Nabi Besar Muhammad SAW.

Penyusunan Tugas Akhir ini diajukan untuk memenuhi syarat akademis dalam rangka menyelesaikan Studi S1 Program Studi Teknik Informatika di Fakultas Teknik Universitas Muhammadiyah Malang.

Penulis menyadari bahwa dalam penyusunan tugas akhir ini tidak lepas dari bimbingan dan bantuan berbagai pihak. Oleh karena itu dalam kesempatan ini ucapkan terima kasih penulis sampaikan kepada:

1. Orang tua tercinta Ayah Syahrul dan Ibu Norhikmah yang selalu memberikan semangat, doa, kebahagiaan, nasehat, motivasi, materi, dan semua hal baik yang tidak bisa disebutkan secara rinci.
2. Bapak Agus Eko Minarno, S.Kom., M.kom., selaku Dosen Pembimbing 1 dan Bapak Yufis Azhar, S.Kom., M.Kom., selaku Dosen Pembimbing 2 yang selalu bersedia meluangkan waktu dan tenaga untuk memberikan arahan, petunjuk, serta saran dengan sabar karena saya sadari saya masih lambat dalam memahami sesuatu.
3. Seluruh sahabat Kelas B angkatan 2018 yang menjadi sahabat seperjuangan dalam menuntut ilmu, yang tak pernah mengenal budaya, kasta, dan materi.
4. Seluruh teman – teman yang tidak bisa disebutkan satu persatu yang memberikan motivasi selama penyusunan tugas akhir ini.
5. Penulis menyadari masih banyak kekurangan dalam penulisan tugas akhir ini. Untuk itu, penulis sangat mengharapkan saran yang membangun agar tulisan ini dapat berguna untuk perkembangan ilmu pengetahuan kedepannya.

Malang, 1 Oktober 2022



**Muhammad Aries Ramadhan**



## DAFTAR ISI

<b>LEMBAR PERSETUJUAN .....</b>	<b>i</b>
<b>LEMBAR PENGESAHAN.....</b>	<b>ii</b>
<b>LEMBAR PERNYATAAN .....</b>	<b>iii</b>
<b>ABSTRAK .....</b>	<b>ii</b>
<b>ABSTRACT .....</b>	<b>iii</b>
<b>KATA PENGANTAR .....</b>	<b>iv</b>
<b>DAFTAR ISI.....</b>	<b>v</b>
<b>DAFTAR GAMBAR.....</b>	<b>1</b>
<b>DAFTAR TABEL.....</b>	<b>2</b>
<b>BAB 1 .....</b>	<b>3</b>
<b>PENDAHULUAN.....</b>	<b>3</b>
1.1.    Pendahuluan.....	3
1.2.    Rumusan Masalah.....	5
1.3.    Tujuan Penelitian.....	5
1.4.    Batasan Masalah .....	5
<b>BAB 2 .....</b>	<b>7</b>
<b>LANDASAN TEORI.....</b>	<b>7</b>
2.1    Studi Literatur.....	7
2.2    Konsep Klasifikasi.....	7
2.3    Convolutional Neural Network.....	8
2.4    Model AlexNet .....	8
2.5    Model VGG16.....	8
2.6    Model VGG19.....	9
2.7    Model ResNet50 .....	9
2.8    Model ResNet101 .....	9
2.9    Model Inception-V3.....	9
2.10    Model DenseNet201 .....	10
2.11    Model Exception.....	10
2.12    Model MobileNet V2.....	10
2.13    Model Inception ResNet V2 .....	10
2.14    Model ResNet152 V2 .....	10
2.15    Tensorflow.....	10
2.16    Matplotlib .....	11
2.17    Seaborn .....	11
2.18    Pengukuran performa.....	11
<b>BAB 3 .....</b>	<b>12</b>



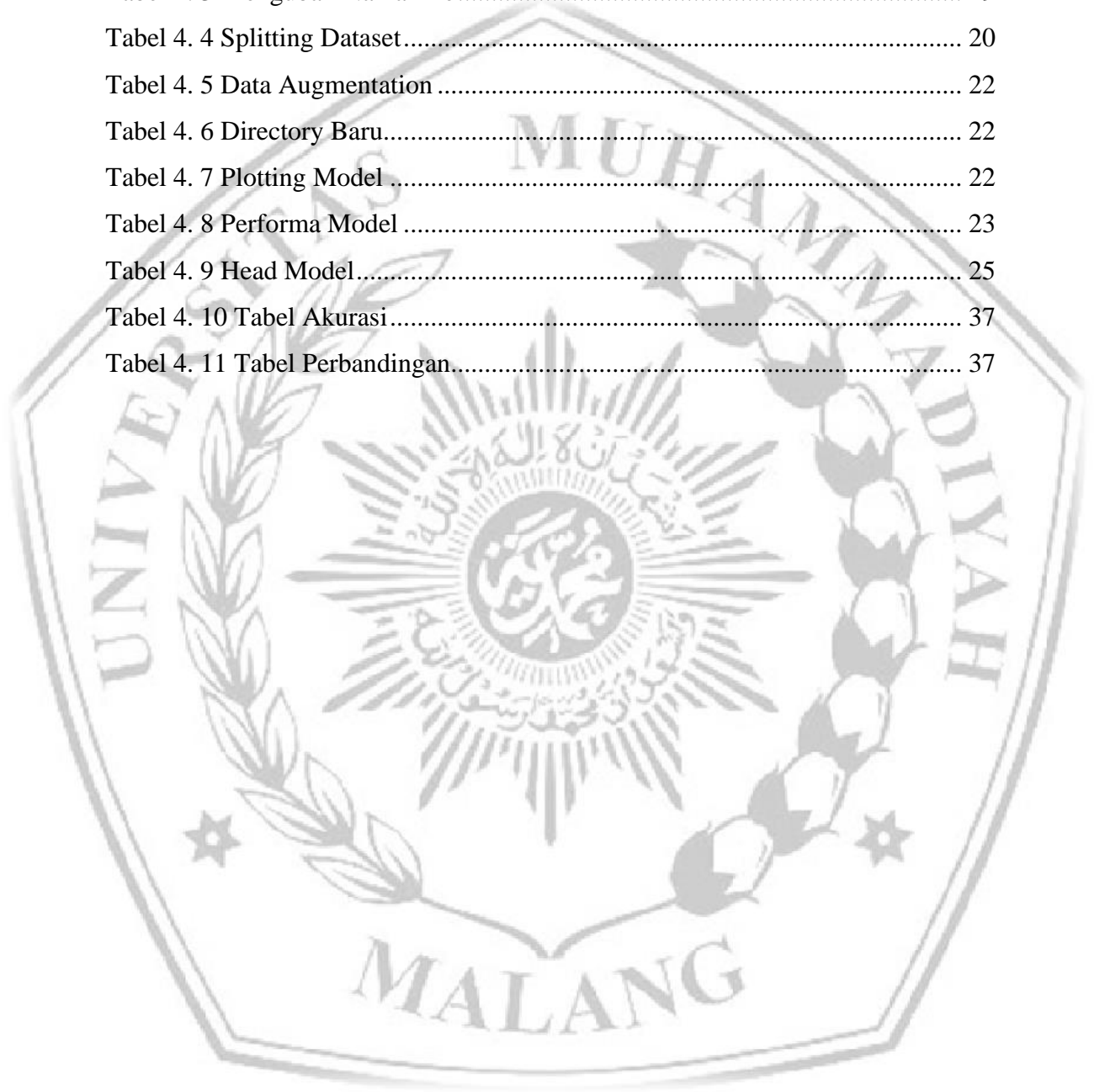
<b>METODOLOGI PENELITIAN .....</b>	<b>12</b>
3.1 Rancangan penelitian.....	12
3.2 Perancangan Model .....	14
3.3 Dataset .....	15
3.4 Skenario Pengujian .....	16
<b>BAB 4 .....</b>	<b>17</b>
<b>IMPLEMENTASI DAN PENGUJIAN .....</b>	<b>17</b>
4.1 Import Library .....	17
4.2 Load Dataset.....	19
4.3 Splitting Dataset .....	20
4.4 Data Augmentation.....	21
4.5 Plotting Model .....	22
4.6 Confusion Matrix.....	23
4.7 Implementasi Model.....	25
4.7.1 DenseNet201 .....	25
4.7.2 ResNet152 V2.....	26
4.7.3 VGG16 .....	27
4.7.4 MobileNet V2.....	28
4.7.5 Alexnet .....	29
4.7.6 VGG19 .....	30
4.7.7 Inception V3 .....	31
4.7.8 Xception.....	32
4.7.9 Inception ResNet V2.....	33
4.7.10 ResNet50 .....	34
4.7.11 ResNet101 .....	35
4.8 Hasil Klasifikasi Model .....	36
<b>BAB 5 .....</b>	<b>38</b>
<b>PENUTUP.....</b>	<b>38</b>
5.1 Kesimpulan.....	38
5.2 Saran .....	38
<b>DAFTAR PUSTAKA .....</b>	<b>39</b>

## DAFTAR GAMBAR

Gambar 2. 1 Konsep CNN .....	8
Gambar 3. 1 Rancangan Penelitian .....	13
Gambar 3. 2 Penyakit Daun Jagung .....	15
Gambar 4. 1 DenseNet201 Accuracy and Loss .....	26
Gambar 4. 2 Confusion Matrix DenseNet201 .....	26
Gambar 4. 3 ResNet152 V2 Accuracy and Loss .....	27
Gambar 4. 4 Confusion Matrix ResNet152 V2 .....	27
Gambar 4. 5 VGG16 Accuracy and Loss .....	28
Gambar 4. 6 Confusion Matrix VGG16 .....	28
Gambar 4. 7 MobileNet V2 Accuracy and Loss .....	29
Gambar 4. 8 Confusion Matrix MobileNet V2 .....	29
Gambar 4. 9 Alexnet Accuracy and Loss .....	30
Gambar 4. 10 Confusion Matrix Alexnet .....	30
Gambar 4. 11 VGG19 Accuracy and Loss .....	31
Gambar 4. 12 Confusion Matrix VGG19 .....	31
Gambar 4. 13 Inception V3 Accuracy and Loss .....	32
Gambar 4. 14 Confusion Matrix Inception V3 .....	32
Gambar 4. 15 Xception Accuracy and Loss .....	33
Gambar 4. 16 Confusion Matrix Xception .....	33
Gambar 4. 17 Inception ResNet V2 Accuracy and Loss .....	34
Gambar 4. 18 Confusion Matrix Inception ResNet V2 .....	34
Gambar 4. 19 ResNet50 Accuracy and Loss .....	35
Gambar 4. 20 Confusion Matrix ResNet50 .....	35
Gambar 4. 21 ResNet101 Accuracy and Loss .....	36
Gambar 4. 22 Confusion Matrix ResNet101 .....	36

## DAFTAR TABEL

Tabel 4. 1 Import Library .....	17
Tabel 4. 2 Ekstraksi File.....	19
Tabel 4. 3 Mengubah Nama File.....	19
Tabel 4. 4 Splitting Dataset.....	20
Tabel 4. 5 Data Augmentation .....	22
Tabel 4. 6 Directory Baru.....	22
Tabel 4. 7 Plotting Model .....	22
Tabel 4. 8 Performa Model .....	23
Tabel 4. 9 Head Model.....	25
Tabel 4. 10 Tabel Akurasi.....	37
Tabel 4. 11 Tabel Perbandingan.....	37



## **BAB 1**

### **PENDAHULUAN**

#### **1.1. Pendahuluan**

Tanaman jagung sangat rentan terhadap penyakit daun tertentu seperti northern leaf blight, common rust, and cercospora. Gejala penyakit daun ini tidak dapat dibedakan pada tahap awal. Oleh karena itu, penelitian saat ini menghadirkan solusi melalui deep learning agar kesehatan tanaman dapat dipantau dan akan mengarah pada peningkatan kuantitas serta kualitas produksi tanaman.

Pada penelitian ini dilakukan klasifikasi untuk mendeteksi penyakit pada citra daun jagung menggunakan CNN. Klasifikasi citra adalah topik penelitian yang sedang hangat di masyarakat saat ini dan merupakan arahan penting dalam bidang penelitian pengolahan citra [1]. Convolutional neural network adalah jaringan syaraf tiruan yang terdiri dari puluhan hingga ratusan lapisan. CNN merupakan metode dalam deep learning yang dapat melakukan berbagai tugas seperti klasifikasi citra, segmentasi, pengenalan, dan deteksi objek. Ada banyak pendekatan dalam klasifikasi citra, dan metode yang paling umum adalah convolutional neural networks[2]

Convolutional neural network memiliki banyak kelebihan seperti mendukung pembelajaran berdasarkan representasi selama proses pelatihan, hal ini membuat CNN dapat secara otomatis bias beradaptasi dengan data dan tugas prediksi pada bidang tertentu. CNN mempunyai peningkatan kekuatan dalam proses komputasinya, sehingga dapat memfasilitasi pembuatan neural network yang lebih dalam serta menunjukkan kemampuan dalam merepresentasikan model dengan kinerja yang baik dan lebih akurat.

Dari berbagai kelebihan yang dimiliki CNN, metode ini mempunyai kekurangan yaitu saat membangun dan melakukan training pada data yang banyak akan membutuhkan waktu yang lama dan mengakibatkan beban komputasi yang semakin berat. Meskipun memiliki kelebihan, CNN tidak luput pula dari kekurangannya [3]. Metode Convolutional Neural Network (CNN) sering digunakan untuk memproses klasifikasi gambar. CNN memiliki kelemahan

lainnya yaitu proses pelatihan model yang lama. Salah satu penelitian sebelumnya yang melakukan klasifikasi penyakit daun jagung menggunakan CNN adalah Sibiya & Sumbwanyambe [4]. Mereka menggunakan 3 kelas klasifikasi penyakit yaitu northern leaf blight, common rust, and cercospora. Pada penelitian sebelumnya, Sibiya dan Sumbwanyambe dengan kelas penyakit yang sama menggunakan 100 gambar per kelas dengan rasio 70% untuk training dan 30% untuk testing mendapatkan hasil akurasi sebesar 92.85% [4]. Chen J pada penelitiannya menggunakan mengusulkan metode AlexNet 3D untuk segmentasi otomatis gambar resonansi magnetik kanker prostat, kinerja jaringan 3D AlexNet lebih unggul dalam hal waktu pelatihan dan jumlah parameter, atau evaluasi kinerja jaringan [6]. Saleem G melakukan analisa fitur bentuk daun visual untuk klasifikasi tanaman menggunakan classifier seperti k-nearest neighbor (KNN), decision tree, naïve Bayes, dan multi-support vector machines (SVM) menghasilkan masing-masing 96,1% dan 97,3% hasil pengukuran presisi dan recall.

Dengan banyaknya penelitian yang menggunakan CNN ini membuktikan akan performa dari CNN. Penelitian lain yang menggunakan adalah Ji M identifikasi penyakit daun anggur [7], S Guan deteksi kanker payudara [8], Ting F CNNI klasifikasi kanker payudara [9], Geetharamani G identifikasi penyakit daun menggunakan 9 layer CNN [10], Hekler A klasifikasi kanker kulit superior [11], Medus L melakukan klasifikasi menggunakan CNN untuk mendeteksi kemasan makanan yang rusak dengan kontaminasi objek [12], dan Utoyo M segmentasi semantik pembuluh darah retina arteri-vena menggunakan CNN sederhana mendapatkan akurasi tertinggi lebih dari 98.35% [13].

Deteksi penyakit daun jagung yang benar, cepat dan dini serta pencegahan dan pengendaliannya pada tahap awal sangat menguntungkan. Model CNN yang ditingkatkan digunakan untuk pelatihan dan pengujian empat jenis gambar daun jagung yang dicapai dengan menambahkan fungsi aktivasi unit linier yang diperbaiki dan pengoptimal adam, dengan menyesuaikan parameter, operasi penyatuan, dan mengurangi jumlah pengklasifikasi.

Penelitian ini akan berfokus untuk memperbaiki performa dari semua model yang telah diusulkan sebelumnya dan tidak terdapat bagian preprocessing

data yang jelas sehingga hasil akurasi masih tergolong rendah [5]. Preprocessing data sangat penting dilakukan untuk menghindari data yang tidak seimbang. Dataset yang tidak seimbang dapat membuat distribusi antara class mayoritas (Negative) dan minoritas (positive) tidak seimbang [14]. Thenmozhi K melakukan augmentasi data pada penelitiannya untuk menghindari overfitting [15]. Yan Zhang pada penelitian menggunakan augmentasi data Snapmix dan Mosaic yang terbukti meningkatkan performa dari model ResNet50 [27].

Penelitian ini mengusulkan preprocessing menggunakan data augmentation dengan penambahan model AlexNet, VGG16, VGG19, ResNet50, ResNet101, Inception-V3, DenseNet201, Xception, MobileNetV2, Inception Resnet V2, ResNet152 V2 dan CNN sebagai classifier. Penambahan callback pada model juga dibutuhkan agar dapat memonitor kuantitas dan jika tidak ada perkembangan pada 'patience' dari epoch, model akan otomatis mengurangi tingkat learning rate.

## **1.2. Rumusan Masalah**

Penelitian ini dilaksanakan untuk memecahkan masalah berikut:

1. Apakah metode augmentasi data dapat memberikan hasil akurasi yang lebih tinggi dari pada yang tidak menggunakan augmentasi data?
2. Metode manakah yang paling cocok digunakan untuk feature extraction dan image classification pada penyakit daun jagung?

## **1.3. Tujuan Penelitian**

Tujuan penelitian ini bertujuan menerapkan augmentasi data pada metode CNN dalam mendeteksi dan mengklasifikasikan penyakit daun jagung menggunakan feature extraction dan classification untuk mendapatkan performa yang lebih baik dari penelitian sebelumnya menggunakan 11 jenis model dan modifikasi data augmentasi.

## **1.4. Batasan Masalah**

Adapun batasan masalah yang dikemukakan dalam penelitian ini adalah sebagai berikut:

1. Data penelitian ini menggunakan dataset PlantVillage-Dataset (terdiri dari 4 jenis penyakit jagung).
2. Penelitian ini menggunakan deep learning arsitektur CNN (Convolutional Neural Network).





## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Studi Literatur**

Adapun penelitian sebelumnya yang membahas tentang penyakit daun jagung dan menggunakan data yang sama. Penelitian tersebut juga dijadikan sebagai referensi dalam pembuatan laporan pada tugas akhir ini.

Penelitian Malusi Sibiya dan Mbuyu Sumbwanyambe yaitu “A Computational Procedure for the Recognition and Classification of Maize Leaf Diseases Out of Healthy Leaves Using Convolutional Neural Networks”. Pada penelitian ini dijelaskan bahwa penyakit daun tanaman dapat mempengaruhi daun tanaman sampai batas tertentu yang dapat mengakibatkan tanaman tersebut roboh dan mati. Penyakit daun ini juga dapat mempengaruhi pasokan sayuran dan buah-buahan ke pasar dan dapat mempengaruhi ekonomi pertanian. Banyak penelitian yang telah menggunakan deteksi penyakit daun tanaman. Salah satunya adalah menggunakan metode Convolutional Neural Network (CNN) untuk memodelkan jaringan untuk keperluan pengenalan dan klasifikasi daun tanaman.

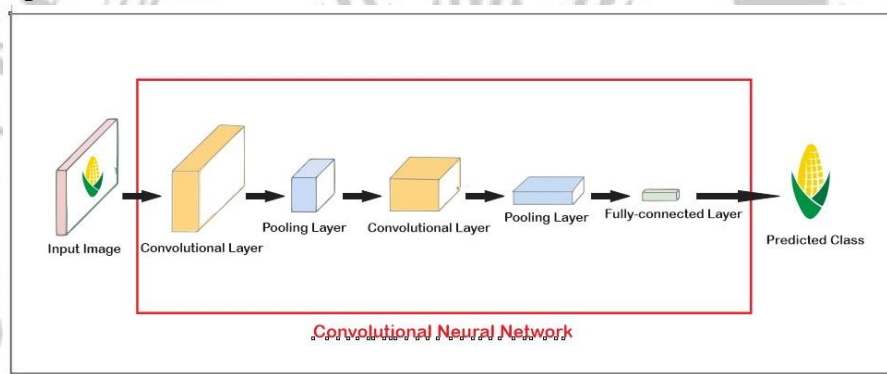
Dataset yang digunakan pada penelitian ini didapatkan dari Plant Village's Datasets. Jumlah data yang terdapat pada Plant Village's Datasets sebanyak 54.000 data. Penelitian ini menggunakan 400 gambar untuk 4 jenis penyakit daun jagung. Data dibagi menjadi 70% untuk data training, 30% untuk data testing dan data validation menggunakan data yang sama seperti data tes. Hasil dari penelitian ini mendapat akurasi sebesar 97%.

#### **2.2 Konsep Klasifikasi**

Konsep klasifikasi dalam deep learning berarti proses memprediksi kelas atau kategori data gambar dengan memanfaatkan informasi atau data yang terdapat pada setiap gambar yang digunakan. Pada penelitian ini ada 11 model digunakan untuk melakukan klasifikasi gambar pada satu set data yang telah diberi label berbeda pada setiap kelasnya.

### 2.3 Convolutional Neural Network

Convolutional neural network (CNN) adalah salah satu jenis algoritma dari deep learning yang bisa menerima inputan berupa gambar dan menentukan objek apa saja yang terdapat pada sebuah gambar yang dapat digunakan mesin untuk belajar mengenali gambar dan membedakan suatu gambar dari yang lainnya. CNN merupakan metode dalam deep learning yang dapat melakukan berbagai tugas seperti klasifikasi citra, segmentasi, pengenalan, dan deteksi objek. Ada banyak pendekatan dalam klasifikasi citra, dan metode yang paling umum adalah Convolutional Neural Networks (CNN)[2]. Cara kerja CNN adalah memanfaatkan proses konvolusi dengan menggerakkan sebuah filter berukuran tertentu ke sebuah gambar dan mendapatkan beberapa informasi dari filter lalu informasi tersebut digunakan untuk melakukan klasifikasi gambar. Pada Gambar 2.1 merupakan konsep dari CNN.



Gambar 2. 1 Konsep CNN

### 2.4 Model AlexNet

AlexNet digunakan sebagai model transfer learning dasar[17]. Alexnet mempunyai 8 layer yang terdiri dari: 5 convolution layer, 3 fully-connected layer dan menggunakan aktivasi relu di setiap layer kecuali dibagian output [18]. Mempunyai 2 Dropout layer, fungsi aktifasi yang digunakan adalah Softmax, dan total jumlah parameternya 62 juta.

### 2.5 Model VGG16

Banyak penelitian yang menggunakan VGG16 [19], [20]. VGG16 mengacu pada lapisan layernya yang memiliki 16 layer. VGG16 dapat mengekstrak beberapa fitur gambar penginderaan jauh, menghapus informasi yang

berlebihan, dan mengenali dan mengklasifikasikan gambar penginderaan jauh[21]. VGG16 berfokus pada convolution layer 3x3, maxpool dan padding yang sama yaitu 2x2 layer, kemudian convolution dan maxpool layer yang konsisten di seluruh arsitekturnya. Terdapat 2 fully-connected layer dan aktivasi layer yaitu softmax. VGG16 merupakan model klasifikasi yang telah dilatih sebelumnya pada dataset ImageNet dan disesuaikan untuk klasifikasi emosi [22]. Majid A menggunakan VGG19 pada penelitiannya klasifikasi infeksi perut dan mendapatkan akurasi 96.5% [23].

### **2.6 Model VGG19**

VGG19 merupakan varian dari model VGG. Berbeda dengan VGG16, model ini memiliki 19 layer yaitu: 16 convolution layer, 3 fully-connected layer, 5 maxpooling, dan 1 softmax layer. VGG19 mempunyai 19 miliar parameter.

### **2.7 Model ResNet50**

Salah satu dari model ResNet yaitu ResNet50 mempunyai 48 convolution layer, 1 maxpooling layer, dan 1 average pool layer. Arsitektur ini bisa digunakan pada computer vision task seperti image classification, object localization, dan object deteksi. Maeda pada penelitiannya juga menggunakan ResNet [24].

### **2.8 Model ResNet101**

ResNet101 adalah convolutional neural network yang mempunyai 101 layer. ResNet101 mempunyai versi pretrained yang telah dilatih sebelumnya dengan lebih dari 1 juta gambar dari database ImageNet. Pretrained yang telah dilatih dapat mengklasifikasikan gambar ke dalam 1000 kategori objek.

### **2.9 Model Inception-V3**

Model ini merupakan neural network yang memiliki 48 layer. Model ini mempunyai pretrained model yang telah dilatih pada lebih dari satu juta gambar dari ImageNet. Pretrained modelnya telah dilatih ke dalam 1000 kategori objek seperti rumah, gunung, bus dan hewan. Yadav S menggunakan Inception-V3 pada penelitiannya yaitu klasifikasi diagnosis penyakit [25].

### **2.10 Model DenseNet201**

Sama seperti Inception-V3, hanya saja DenseNet201 merupakan neural network yang memiliki 201 layer. DenseNet201 juga mempunyai pretrained model yang telah dilatih ke lebih dari 1000 kategori objek.

### **2.11 Model Xception**

Xception adalah model CNN yang dikembangkan oleh google. Xception mempunyai 71 layer dan mempunyai parameter yang sama seperti inception dengan metode komputasi yang lebih efisien.

### **2.12 Model MobileNet V2**

MobileNetV2 merupakan model yang memiliki komputasi yang lebih efisien dari versi sebelumnya. Model ini memiliki 53 layer dan mempunyai pretrained model yang dapat digunakan untuk klasifikasi, object detection, dan semantic segmentation.

### **2.13 Model Inception ResNet V2**

Inception ResNet V2 adalah model CNN yang termasuk dari keluarga inception dengan menggabungkan koneksi residual. Mempunyai 164 layer dan dapat mengklasifikasikan 1000 kategori objek.

### **2.14 Model ResNet152 V2**

ResNet dapat memiliki layer yang sangat dalam hingga 152 layer. ResNet menggunakan skip connection yaitu menyesuaikan input dari layer sebelumnya ke layer berikutnya tanpa modifikasi input apa pun.

### **2.15 Tensorflow**

TensorFlow merupakan library perangkat lunak yang bersifat open source untuk pemrograman di berbagai bidang. TensorFlow adalah sebuah framework machine learning yang digunakan untuk keperluan memproses data. Penelitian ini menggunakan TensorFlow untuk membangun model. Banyak perusahaan besar yang menggunakan TensorFlow untuk menerapkan machine learning dan memecahkan masalah mereka seperti: Google, Lenovo, Coca-Cola.

## 2.16 Matplotlib

Matplotlib merupakan library python yang paling populer untuk melakukan visualisasi data yang lebih menarik dan mudah dipahami sehingga banyak yang menggunakan library ini. Matplotlib dapat digunakan untuk memvisualisasikan data secara 2D ataupun 3D dan menghasilkan gambar yang berkualitas dan dapat disimpan dalam berbagai macam format seperti JPEG dan PNG.

## 2.17 Seaborn

Seaborn merupakan library visual python yang berlandaskan pada matplotlib. Seaborn mempunyai antar-muka tingkat tinggi untuk menangani permasalahan terkait visualisasi data secara statistik agar lebih mudah dipahami dan tentunya agar terlihat lebih menarik. Penelitian ini menggunakan seaborn agar proses eksplorasi dan pemahaman data dapat ditampilkan secara baik menggunakan visualisasi dari seaborn.

## 2.18 Pengukuran performa

Akurasi model adalah sebagai jumlah klasifikasi yang diprediksi dengan benar oleh model yang dibuat lalu dibagi dengan jumlah total prediksi yang dibuat. Akurasi merupakan metode untuk menilai kinerja dari sebuah model. Rumus akurasi ditulis dengan persamaan:

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (R1)$$

TP mewakili: Berapa banyak sampel uji yang diprediksi benar dan sebenarnya sampel ini dianotasi ke kelas benar. TN mewakili: Berapa banyak sampel uji yang diprediksi salah dan sebenarnya sampel ini dianotasi ke kelas benar. FP mewakili: Berapa banyak sampel uji yang diprediksi benar dan sebenarnya sampel ini dianotasi ke kelas salah. FN mewakili: Berapa banyak sampel uji yang diprediksi salah dan sebenarnya sampel ini dijelaskan ke kelas salah. Maka semua harus dikalikan 100% untuk mencari persentasenya.

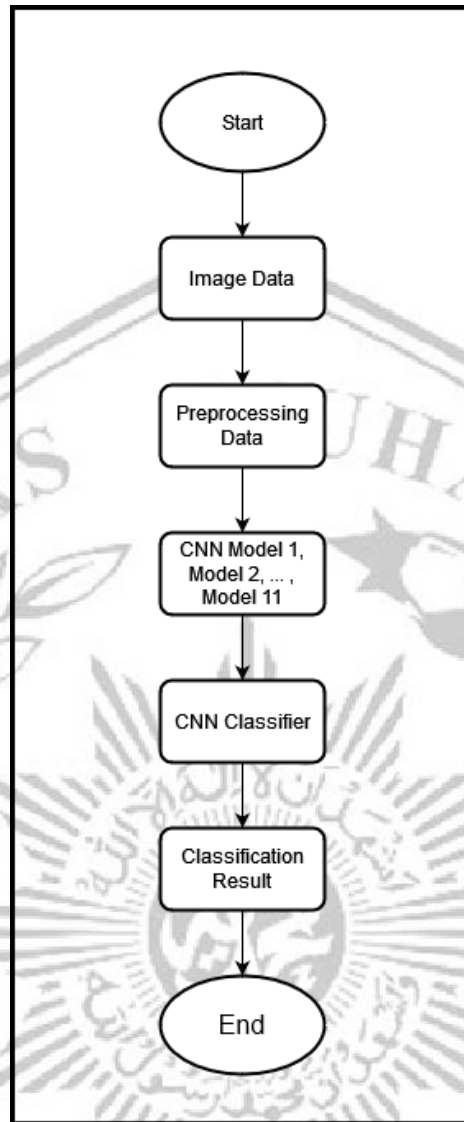
## BAB 3

### METODOLOGI PENELITIAN

Pada penelitian sebelumnya, Sibiya dan Sumbwanyambe dengan kelas penyakit yang sama menggunakan 100 gambar per kelas dengan rasio 70% untuk training dan 30% untuk testing mendapatkan hasil akurasi sebesar 92.85% [4]. Mohammad Syarief dan Wahyudi Setiawan menggunakan tujuh CNN model yaitu: Alexnet, VGG16, VGG19, GoogleNet, Inception-V3, Resnet50, dan Resnet101 dengan klasifikasi menggunakan metode machine learning seperti: KNN, Decision tree, dan SVM. Berdasarkan hasil yang didapat, klasifikasi terbaik menggunakan Alexnet dan SVM dengan akurasi, sensitifitas, dan spesifitas sebesar 95.83%, 100%, dan 95% [5]. Setiawan W pada penelitiannya klasifikasi menggunakan model Convolutional Neural Network (CNN) dan metode klasifikasi seperti Support Vector Machine, k-Nearest Neighbor, Naïve Bayes classifier, Discriminant Analysis, dan Decision Tree mendapatkan hasil pengujian menggunakan tiga skenario diperoleh akurasi klasifikasi 90%-100% dengan linear loss cross validation 0%-26,67% [16].

#### 3.1 Rancangan penelitian

Pada tahap rancangan penelitian akan menjelaskan tahapan yang perlu dilakukan. Data yang terdapat pada penelitian ini akan dilakukan preprocessing data. Data yang diperoleh terbagi menjadi 4 kelas yaitu Healty, Cercospora, Common Rust, Northern Leaf Blight. Data gambar yang digunakan memiliki  $224 \times 224$  piksel lalu akan diubah ukurannya menjadi  $224 \times 224$  piksel dan data siap digunakan. Untuk lebih jelasnya terkait dengan rancangan penelitian dapat dilihat pada Gambar 3.1.



*Gambar 3. 1 Rancangan Penelitian*

Data yang sudah siap digunakan akan dilakukan splitting data menjadi: data training 70% dan data tes 30% seperti pada penelitian Sibiya [4]. Selanjutnya akan dilakukan augmentasi data menggunakan library keras ImageDataGenerator untuk menghindari adanya overfitting data. Berikut merupakan penjelasan dari data augmentasi yang digunakan:

1. Rescale

Rescale akan melakukan konversi pixel dari  $[0, 255]$  ke  $[0, 1]$ . Proses ini juga dikenal dengan Normalizing, scaling semua gambar agar memiliki pixel  $[0, 1]$ . Rescale membuat gambar berkontribusi lebih merata terhadap total loss.

2. Rotation\_image



Metode ini digunakan untuk melakukan rotasi gambar dengan nilai 0 hingga 360 derajat searah jarum jam.

3. Zoom\_range

ImageDataGenerator mempunyai zoom\_range untuk memperbesar gambar. Value yang lebih besar dari 1 akan melakukan zoom out pada gambar, sebaliknya jika value kurang dari 1 maka akan melakukan zoom in.

4. Width\_shift\_range dan Height\_shift\_range

ImageDataGenerator mempunyai argument ini untuk menggeser gambar secara horizontal dan vertical, jika valuenya berupa float itu berarti jumlah persentasi dari lebar dan tinggi gambar yang bergeser. Jika valuenya berupa integer maka lebar dan tinggi bergeser berdasarkan jumlah pixel.

5. Shear\_range

Shear\_range berarti melakukan distorsi pada gambar sepanjang sumbunya. Biasanya untuk membuat atau memperbaiki sudut persepsi.

6. Horizontal\_flip

Horizontal\_flip memiliki fungsi untuk membalikkan gambar secara horizontal axis. Teknik ini harus menyesuaikan dengan objek gambar.

7. Fill\_mode

Saat gambar bergeser 20% maka akan ada ruang yang tersisa, fill\_mode akan mengisi kekosongan tersebut dengan beberapa method seperti constant, nearest, reflect, dan wrap.

### 3.2 Perancangan Model

Tahapan selanjutnya adalah membangun model yang akan digunakan. Pada penelitian ini akan menggunakan transfer learning model convolutional neural network (CNN). Model yang akan digunakan pada penelitian ini yaitu: AlexNet, VGG16, VGG19, ResNet50, ResNet101, Inception-V3, DenseNet201, Xception, MobileNetV2, Inception Resnet V2, dan ResNet152 V2.

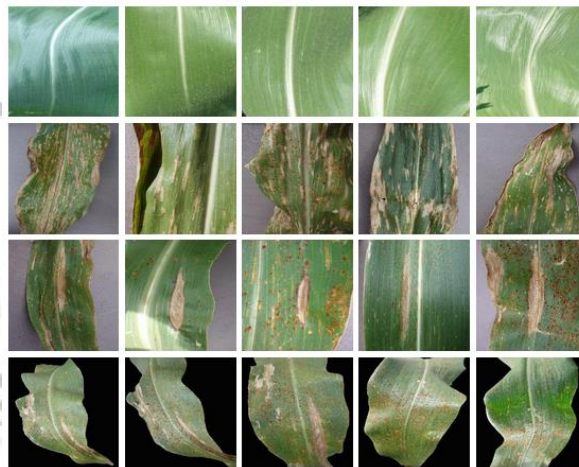
Model yang sudah dibangun akan dilakukan proses training dengan menggunakan 100 epoch. Callback menggunakan library keras ReduceLROnPlateau dengan spesifikasi: monitor = 'val\_accuracy', mode = 'max',

patience = 5, restore\_best\_weight = True. Callback digunakan untuk mengurangi kecepatan pembelajaran ketika metrik berhenti meningkat.

Setelah melakukan proses training selanjutnya menampilkan hasil dari penelitian berupa plotting, confusion matrix dan melakukan save terhadap semua model yang digunakan. Tahap terakhir melakukan klasifikasi menggunakan semua model yang digunakan.

### 3.3 Dataset

Data yang digunakan merupakan data yang bersumber dari PlantVillage. Data yang digunakan pada penelitian ini adalah corn leaf disease yang dapat diakses di PlantVillage-Dataset. Dataset tersebut juga digunakan pada penelitian sebelumnya [28]. Dataset ini mempunyai 4 kelas sebagai berikut: Healthy: 1162, Cercospora: 513, Common Rust: 1192, Northern Leaf Blight: 958. Data tersebut digunakan dengan rasio 70% dan 30% untuk tes. Pada Gambar 3.2 merupakan gambar penyakit jagung yang terdapat pada dataset ini.



Gambar 3. 2 Penyakit Daun Jagung

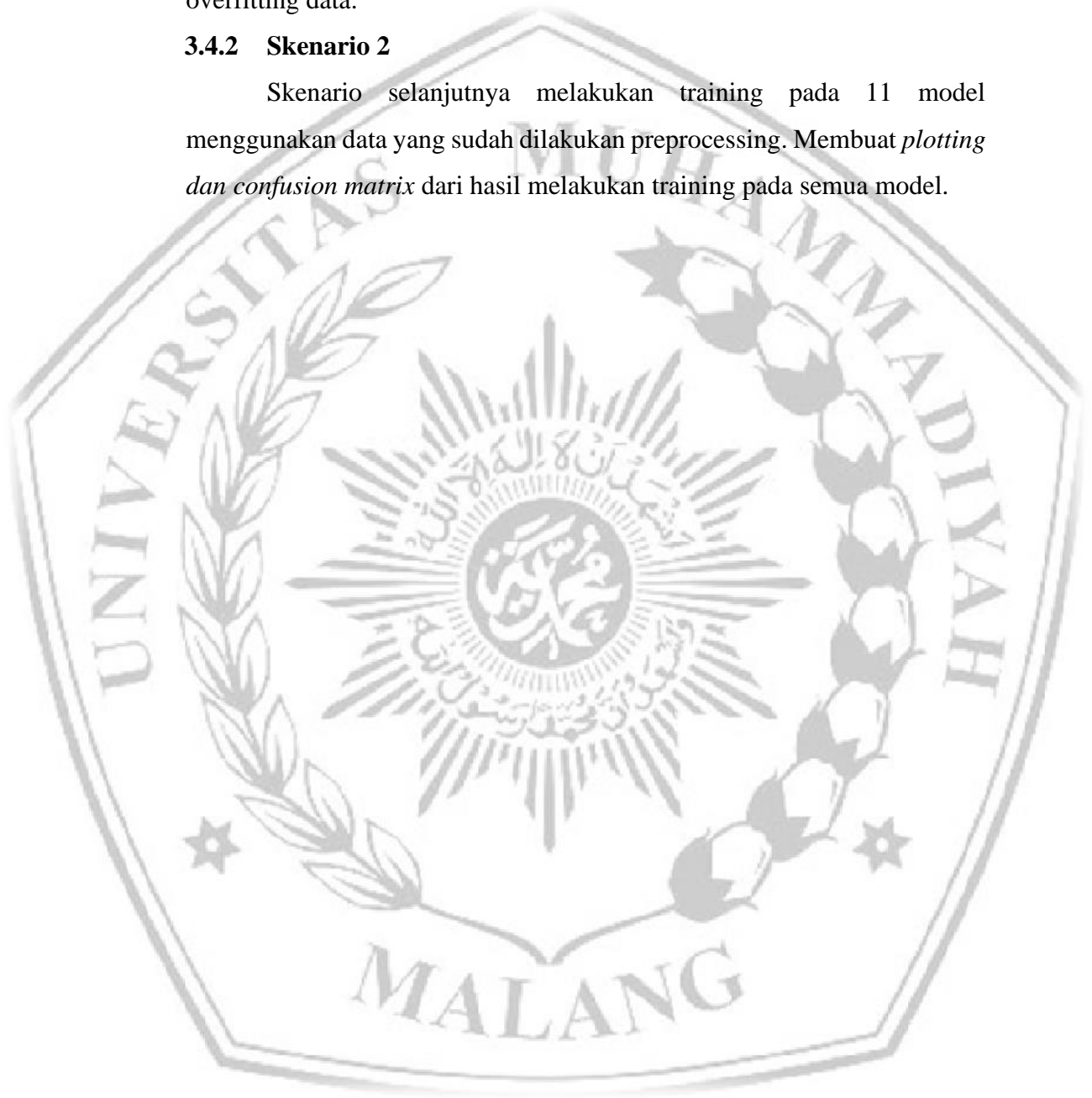
### 3.4 Skenario Pengujian

#### 3.4.1 Skenario 1

Skenario pertama melakukan preprocessing data pada dataset meliputi rescale, splitting data, dan augmentasi data untuk menghindari overfitting data.

#### 3.4.2 Skenario 2

Skenario selanjutnya melakukan training pada 11 model menggunakan data yang sudah dilakukan preprocessing. Membuat *plotting* dan *confusion matrix* dari hasil melakukan training pada semua model.



## BAB 4

### IMPLEMENTASI DAN PENGUJIAN

#### 4.1 Import Library

Pada penelitian ini menggunakan beberapa library yang berfungsi untuk melakukan proses klasifikasi gambar.

*Tabel 4. 1 Import Library*

```
import seaborn as sns
import matplotlib.pyplot as plt
import glob
import cv2
import os
import zipfile
import shutil

import tensorflow as tf
from tensorflow import keras
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras import layers
from tensorflow.keras import Model
from tensorflow.keras import models
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, BatchNormalization, Dropout

#Pretrained Model
from tensorflow.keras.layers import Input
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.applications.vgg19 import VGG19
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.applications.resnet import ResNet101
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.applications.densenet import DenseNet201
from tensorflow.keras.applications.xception import Xception
from tensorflow.keras.applications.mobilenet_v2 import MobileNetV2
from tensorflow.keras.applications.inception_resnet_v2 import InceptionResNetV2
from tensorflow.keras.applications.resnet_v2 import ResNet152V2

#Classification Report
```

```
from sklearn.metrics import classification_report, confusion_matrix
```

Pada penelitian ini library digunakan untuk memudahkan penulis untuk memproses semua data. Beberapa library pada Tabel 4.1 digunakan untuk proses perhitungan akurasi, memanggil semua model yang akan digunakan, dan visualisasi hasil training. Berikut penjelasan mengenai library yang digunakan dalam penelitian ini:

1. Matplotlib, library ini digunakan untuk visualisasi data seperti histogram, barchart, linechar, dan lain-lain. Pada penelitian ini Matplotlib digunakan untuk membuat plotting dari hasil akurasi dan loss dari model yang ditraining. Library ini juga digunakan untuk memvisualisasikan confusion matrix.
2. Seaborn, library ini digunakan untuk visualisasi data yang dibangun dari matplotlib. Seaborn memiliki lebih banyak fungsi visualisasi dan sangat mudah digunakan. Pada penelitian ini seaborn digunakan untuk memberikan warna pada confusion matrix.
3. Os, library ini untuk memudahkan peneliti dalam mengakses folder untuk keperluan penamaan folder, membuat folder, dan mendapatkan list dari semua file dan penyimpanan yang digunakan.
4. Zipfile, library ini digunakan untuk membuat, membaca, menulis, dan melakukan ekstrak dari file Zip. Pada penelitian ini menggunakan data berupa file Zip lalu dilakukan proses ekstraksi pada file Zip.
5. Shutil, library ini digunakan untuk melakukan pemindahan data dari folder ke folder lain untuk memudahkan preprocessing data.
6. tensorflow keras, library ini digunakan untuk proses pengembangan machine learning. Library ini digunakan untuk memanggil model dan melakukan proses training pada semua model yang digunakan pada penelitian ini.
7. keras ImageDataGenerator, library ini termasuk kedalam tensorflow. Library ini digunakan untuk proses data sebelum di load dan melakukan proses data augmentasi. Pada penelitian ini menggunakan berbagai



macam fungsi seperti rescale, rotation, zoom, width shift, height shift, shear range, flip, dan fill mode

8. sklearn metrics, library ini digunakan untuk keperluan classification report dan pembuatan confusion matrix. Classification report berisi laporan tentang proses training dari model yang akan digunakan dan untuk melihat seberapa tinggi akurasi dari semua model.

#### 4.2 Load Dataset

Dataset yang digunakan pada penelitian ini disimpan dalam bentuk Zip File yang sudah disimpan pada penyimpanan lokal. Pada Tabel 4.2 peneliti melakukan proses ekstraksi file menggunakan library zipfile. File ekstraksi berisi 4 macam folder yang terbagi menjadi beberapa macam penyakit daun jagung yaitu: Corn Cercospora, Corn Common Rust, Corn Northern Leaf Blight, dan Corn Healthy. Setelah melakukan proses ekstraksi pada Tabel 4.3, peneliti melakukan perubahan nama sesuai dengan nama penyakitnya menggunakan library os pada semua data yang terdapat pada setiap folder untuk memudahkan klasifikasi data.

*Tabel 4. 2 Ekstraksi File*

```
import zipfile
local_zip = '/content/PlantZ.zip'
zip_ref = zipfile.ZipFile(local_zip, 'r')
zip_ref.extractall('/content')
zip_ref.close()
```

*Tabel 4. 3 Mengubah Nama File*

```
path = '/content/Plant/Corn_Cercospora/'
files = os.listdir(path)

for index, file in enumerate(files):
    os.rename(path+file, path + 'Corn_Cercospora' + str(index)
    + '.jpg')

path = '/content/Plant/Corn_Common_rust/'
files = os.listdir(path)

for index, file in enumerate(files):
    os.rename(path+file, path + 'Corn_Common_rust' + str(index)
    + '.jpg')
```

```

path = '/content/Plant/Corn_Northern_Leaf_Blight/'
files = os.listdir(path)

for index, file in enumerate(files):
    os.rename(path+file, path + 'Corn_Northern_Leaf_Blight' +
str(index)+ '.jpg')

path = '/content/Plant/Corn_healthy/'
files = os.listdir(path)

for index, file in enumerate(files):
    os.rename(path+file, path + 'Corn_healthy' + str(index)+ '
.jpg')

```

### 4.3 Splitting Dataset

Splitting dataset diperlukan untuk melakukan proses training data. Dataset pada penelitian ini dibagi menjadi 2 yaitu data training dan data validasi dengan rasio 70% untuk data training dan 30% untuk data tes yang bisa dilihat pada Tabel 4.4.

*Tabel 4. 4 Splitting Dataset*

```

# TULIS KODE ANDA DISINI
import random

def train_valid_test_split(source, train_dir, valid_dir, test_dir, train_size, valid_size, test_size):
    # get files
    files = []
    for filename in os.listdir(source):
        file = source + filename
        if os.path.getsize(file) > 0:
            files.append(filename)

    # train valid split

    train_size = int(len(files) * (1 - (valid_size + test_size)))
    valid_size = int(len(files)* (1 - (valid_size + test_size)))
    test_size = int(len(files)* (1 - train_size))

```



```

# shuffle the dataset
shuffled_files = random.sample(files, len(files))

train_set = shuffled_files[train_size:valid_size]
valid_set = shuffled_files[valid_size:]
test_set = shuffled_files[:train_size]

for filename in train_set:
    filepath = source + filename
    destination = train_dir + filename
    copyfile(filepath, destination)

for filename in valid_set:
    filepath = source + filename
    destination = valid_dir + filename
    copyfile(filepath, destination)

for filename in test_set:
    filepath = source + filename
    destination = test_dir + filename
    copyfile(filepath, destination)

SOURCE = "/content/corn/dataset/"

for label in ['Corn_Cercospora', 'Corn_Common_rust', 'Corn_Northern_Leaf_Blight', 'Corn_healthy']:
    print(label)
    train_valid_test_split(SOURCE+label+'/', TRAIN_DIR +label+'/', VALID_DIR +label+'/', TEST_DIR +label+'/', train_size=0.70, valid_size=0.0, test_size=0.30)
    print(len(os.listdir(TRAIN_DIR +label+'/')))
    print(len(os.listdir(VALID_DIR +label+'/')))
    print(len(os.listdir(TEST_DIR +label+'/')))

```

#### 4.4 Data Augmentation

Pada Tabel 4.5 peneliti melakukan data augmentation pada data training berupa Rescale, Rotation Image, Zoom Range, Width Shift Range, Height Shift Range, Shear Range, Horizontal Flip, dan Fill Mode. Setelah data augmentation dilakukan selanjutnya memisahkan dataset menjadi dua bagian data yang dapat dilihat pada Tabel 4.6, lalu mengganti besaran piksel menjadi  $224 \times 224$  piksel untuk menyesuaikan ukuran input data untuk semua model.

*Tabel 4. 5 Data Augmentation*

```
from keras.preprocessing.image import ImageDataGenerator

train_generator = ImageDataGenerator(
    rescale = 1./255,
    rotation_range = 90,
    zoom_range = 0.2,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.2,
    horizontal_flip=True,
    fill_mode="nearest"
)
```

*Tabel 4. 6 Directory Baru*

```
train_images = train_generator.flow_from_directory(
    TRAIN_DIR,
    target_size=(SIZE, SIZE),
    color_mode='rgb',
    class_mode='categorical',
    batch_size=32,
    shuffle=True
)

val_images = test_generator.flow_from_directory(
    VALID_DIR,
    target_size=(SIZE, SIZE),
    color_mode='rgb',
    class_mode='categorical',
    batch_size=32,
    shuffle=True,
    subset = 'training'
)
```

#### **4.5 Plotting Model**

Pada bagian ini peneliti membuat fungsi untuk memvisualisasikan hasil dari training data menggunakan library matplotlib yang dapat dilihat pada Tabel 4.7. Fungsi ini berisi pembuatan plotting dari hasil akurasi dan loss plot. Peneliti juga melakukan proses menyimpan hasil plotting ke dalam direktori lokal.

*Tabel 4. 7 Plotting Model*

```
import matplotlib.pyplot as plt
def eval_plot(history, plotName):
```

```

plt.figure(figsize=(14, 5))

# Accuracy plot
plt.subplot(1, 2, 1)
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
epochs = range(len(acc))
acc_plot, = plt.plot(epochs, acc, 'r')
val_acc_plot, = plt.plot(epochs, val_acc, 'b')
plt.title('Training and Validation Accuracy')
plt.legend([acc_plot, val_acc_plot], ['Training Accuracy',
'Validation Accuracy'])

# Loss plot
plt.subplot(1, 2, 2)
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(len(loss))
loss_plot, = plt.plot(epochs, loss, 'r')
val_loss_plot, = plt.plot(epochs, val_loss, 'b')
plt.title('Training and Validation Loss')
plt.legend([loss_plot, val_loss_plot], ['Training Loss', 'V
alidation Loss'])
plt.savefig("/content/drive/MyDrive/Seminar Proposal Model/
Evalplot/"+plotName+".svg")

```

#### 4.6 Confusion Matrix

Pada tahap ini peneliti menggunakan library sklearn metrics dan seaborn. Peneliti perlu melihat hasil dari model menggunakan classification report dan melakukan uji klasifikasi menggunakan confusion matrix. Classification report berisi evaluasi tentang performa dari setiap model yang digunakan yang berisi akurasi, presisi, dan recall. Confusion matrix diperlukan untuk menguji klasifikasi yang berguna untuk mengukur tingkat performa dari setiap model yang dibuat, peneliti menggunakan library seaborn untuk melakukan plotting dari confusion matrix. Pada Tabel 4.8 peneliti menggunakan sklearn metrics dan seaborn.

*Tabel 4. 8 Performa Model*

```

from sklearn.metrics import classification_report, confusion_
matrix

```

```

import seaborn as sns

def evaluate(Model, modelName):

    BS = 32;

    valid_generator = train_generator.flow_from_directory(
        TRAIN_DIR,
        target_size=(SIZE,SIZE),
        batch_size= BS,
        class_mode='categorical',
        shuffle = False)

    batch_size = BS
    num_of_test_samples = len(valid_generator.filesnames)

    Y_pred = Model.predict_generator(valid_generator, num_of_test_samples // batch_size+1)
    y_pred = np.argmax(Y_pred, axis=1)

    print('\nConfusion Matrix ' + modelName + '\n')
    cm = confusion_matrix(valid_generator.classes, y_pred)
    plt.figure(figsize=(8, 8))
    sns.heatmap(cm, annot=True, fmt='g', vmin=0, cmap='Blues',
cbar=False)
    plt.xticks(ticks=[0.5, 1.5, 2.5, 3.5], labels=['Corn_Cercospora', 'Corn_Common_rust', 'Corn_Northern_Leaf_Blight', 'Corn_healthy'])
    plt.yticks(ticks=[0.5, 1.5, 2.5, 3.5], labels=['Corn_Cercospora', 'Corn_Common_rust', 'Corn_Northern_Leaf_Blight', 'Corn_healthy'])
    plt.xlabel("Predicted")
    plt.ylabel("Actual")
    plt.title("Confusion Matrix " + modelName)
    plt.show()
    # plt.savefig("/content/drive/MyDrive/model/ConfMat"+modelName+".svg")

    print('\n\nClassification Report\n')
    target_names = ['Corn_Cercospora', 'Corn_Common_rust', 'Corn_Northern_Leaf_Blight', 'Corn_healthy']
    print(classification_report(valid_generator.classes, y_pred, target_names=target_names))

```

## 4.7 Implementasi Model

Pada tahap ini akan dilakukan proses pengujian terhadap semua model. Dibawah ini merupakan hasil training dari semua model dan menampilkan hasil *accuracy*, *loss*, dan *confusion matrix* dari semua model. Pada Tabel 4.9 merupakan inisialisasi headmodel yang digunakan pada semua model. Layer Dropout menggunakan aktivasi softmax dan optimizers menggunakan Adam Optimizers.

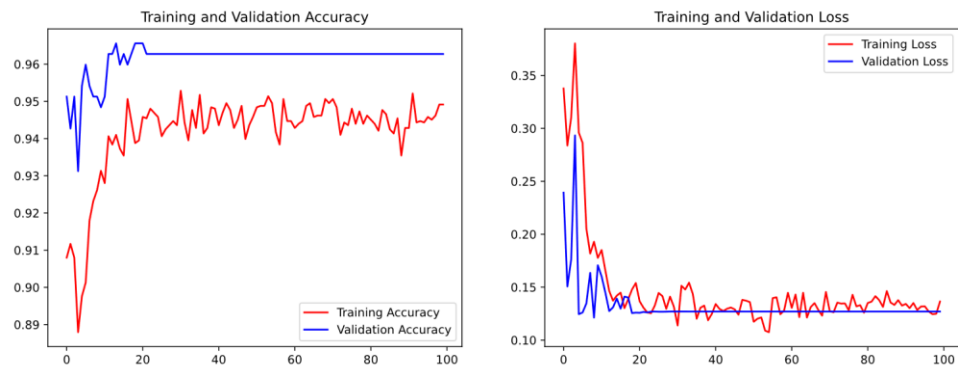
Tabel 4. 9 Head Model

```
class FCHeadNet:
    def build(baseModel, classes, D):
        headModel = baseModel.output
        headModel = Flatten(name="flatten")(headModel)
        headModel = Dense(D, activation='elu')(headModel)
        headModel = Dropout(0.5)(headModel)
        headModel = Dense(classes, activation='softmax')(headModel)
    return headModel

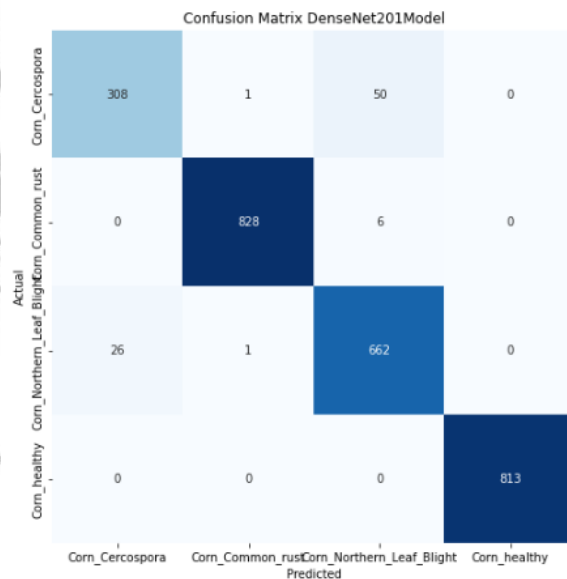
headModel = FCHeadNet.build(baseModelvgg16, 4, 256)
modelvgg16 = Model(inputs= baseModelvgg16.input, outputs=headModel)
```

### 4.7.1 DenseNet201

DenseNet201 merupakan convolutional neural network yang mempunyai layer sebanyak 201 layer. DenseNet2101 merupakan salah satu arsitektur yang lebih efisien dari arsitektur lainnya. DenseNet201 berfokus membuat jaringan deep learning menjadi lebih dalam lagi dengan 201 layernya menggunakan koneksi yang lebih pendek antar lapisannya. Pada penelitian ini diterapkan proses transfer learning menggunakan DenseNet201. Lalu akan dilakukan training model dengan epochs 100. Pada Gambar 3 merupakan hasil dari training model DenseNet201 dengan menggunakan 100 epoch dan DenseNet201 mendapatkan akurasi sebesar 97%.



Gambar 4. 1 DenseNet201 Accuracy and Loss

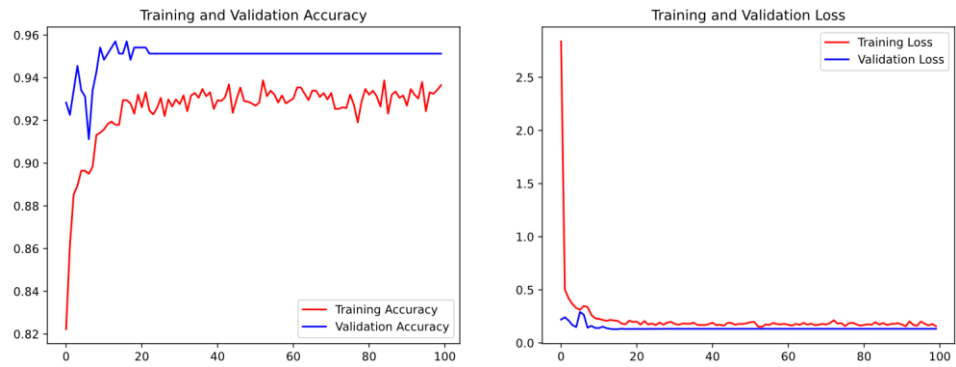


Gambar 4. 2 Confusion Matrix DenseNet201

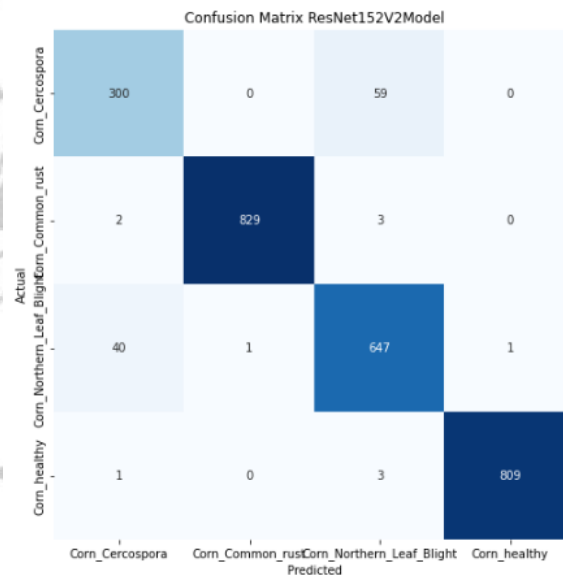
#### 4.7.2 ResNet152 V2

ResNet152 V2 merupakan deep learning yang mempunyai layer sebanyak 152 layer dengan mempelajari fungsi representasi residual. ResNet memperkenalkan skip connection untuk menyesuaikan input dari layer sebelumnya ke layer berikutnya tanpa ada modifikasi dari inputnya. Lalu akan dilakukan training model dengan epochs 100. Pada Gambar 3.2 merupakan hasil dari training model ResNet152 V2 dengan menggunakan 100 epoch dan ResNet152 V2 mendapatkan akurasi sebesar 96%.





Gambar 4. 3 ResNet152 V2 Accuracy and Loss

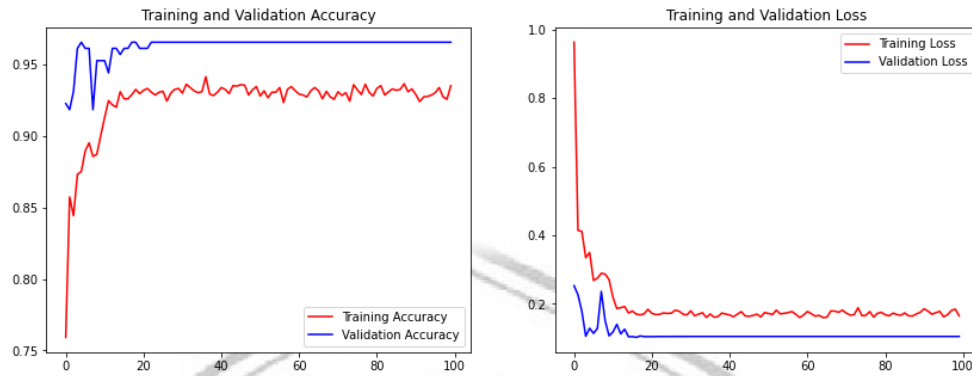


Gambar 4. 4 Confusion Matrix ResNet152 V2

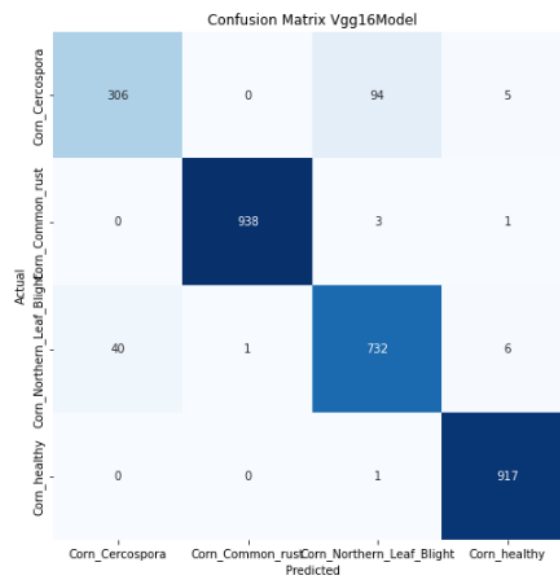
#### 4.7.3 VGG16

Arsitektur VGG16 merupakan convolutional neural network yang mempunyai 16 layer yang terdiri dari convolutional, max pooling, layer activation, dan fully connected layer. Berikutnya dilakukan training model dengan epochs 100. Keunggulan dari VGG16 adalah menampilkan arsitektur yang sangat homogen. VGG16 hanya melakukan convolutional 3x3 dan pooling 2x2 dari awal hingga akhir. Pada Gambar 4.5 merupakan hasil dari training model VGG16 dengan menggunakan 100 epoch dan VGG16 mendapatkan akurasi sebesar 95%.





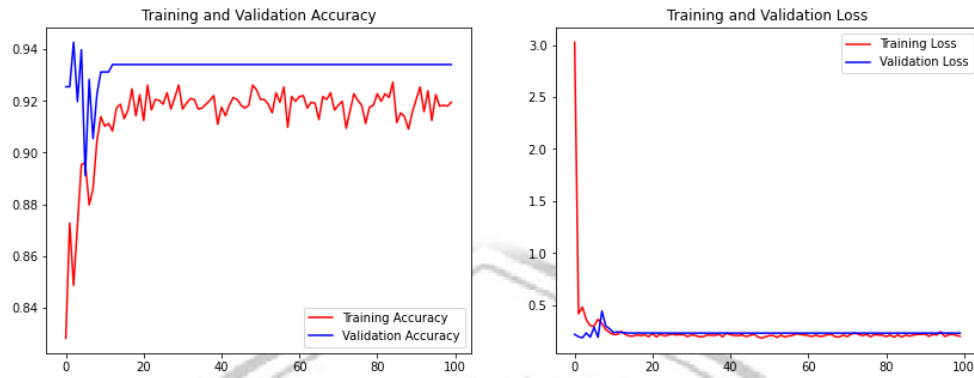
Gambar 4. 5 VGG16 Accuracy and Loss



Gambar 4. 6 Confusion Matrix VGG16

#### 4.7.4 MobileNet V2

MobileNet V2 sangat mirip dengan MobileNet, MobileNet V2 memiliki jumlah parameter yang jauh lebih rendah dari MobileNet. Model ini mendukung ukuran input yang lebih besar dari 32x32 dengan ukuran gambar yang lebih besar dan hal inilah yang membuat kinerja dari MobileNet V2 adalah salah satu dengan kinerja yang baik. Pada Gambar 4.7 merupakan hasil dari training model MobileNet V2 dan mendapatkan akurasi sebesar 95%. Gambar 4.8 merupakan confusion matrix dari MobileNet V2.



Gambar 4. 7 MobileNet V2 Accuracy and Loss

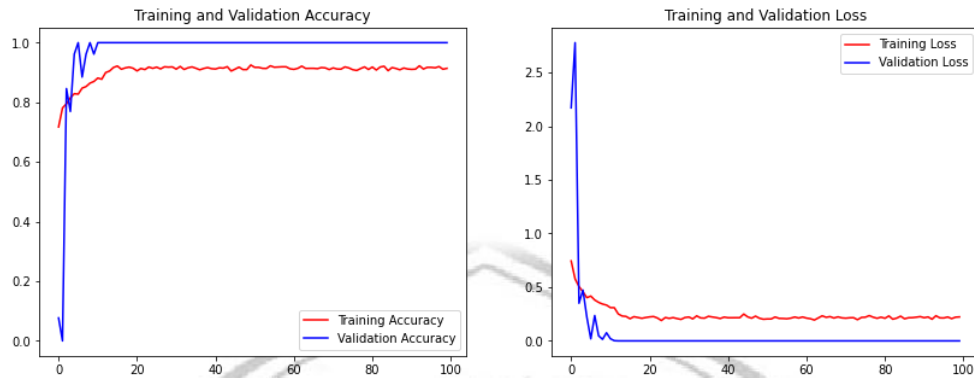
Confusion Matrix MobileNetV2Model

	Corn_Cercospora	Corn_Common_rust	Corn_Northern_Leaf_Blight	Corn_healthy
Actual	294	1	64	0
Corn_Cercospora	2	827	5	0
Corn_Common_rust	46	1	641	1
Corn_Northern_Leaf_Blight	1	0	1	811
Corn_healthy				
Predicted	Corn_Cercospora	Corn_Common_rust	Corn_Northern_Leaf_Blight	Corn_healthy

Gambar 4. 8 Confusion Matrix MobileNet V2

#### 4.7.5 Alexnet

Alexnet adalah salah satu model pertama yang menggunakan GPU untuk melakukan training dan hal inilah yang membuat Alexnet dapat melakukan training lebih cepat. Alexnet mempunyai arsitektur dengan 8 layer yang membuatnya dapat melakukan ekstraksi fitur dengan lebih baik. Gambar 4.9 merupakan kinerja dari Alexnet dengan akurasi mencapai 94% dan Gambar 4.10 adalah coonfusion matrix dari model ini.



Gambar 4. 9 Alexnet Accuracy and Loss

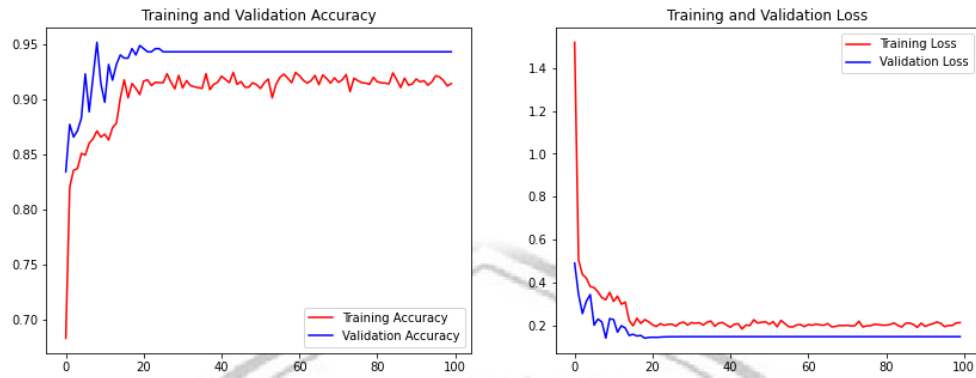
Confusion Matrix AlexnetModel

	Corn_Cercospora	Corn_Common_rus	Corn_Northern_Leaf_Blight	Corn_healthy
Actual	<div>Corn_Cercospora</div> <div>301</div> <div>0</div> <div>103</div> <div>1</div>	<div>Corn_Common_rus</div> <div>3</div> <div>938</div> <div>1</div> <div>0</div>	<div>Corn_Northern_Leaf_Blight</div> <div>55</div> <div>0</div> <div>722</div> <div>2</div>	<div>Corn_healthy</div> <div>1</div> <div>0</div> <div>2</div> <div>553</div>
Predicted	Corn_Cercospora	Corn_Common_rus	Corn_Northern_Leaf_Blight	Corn_healthy

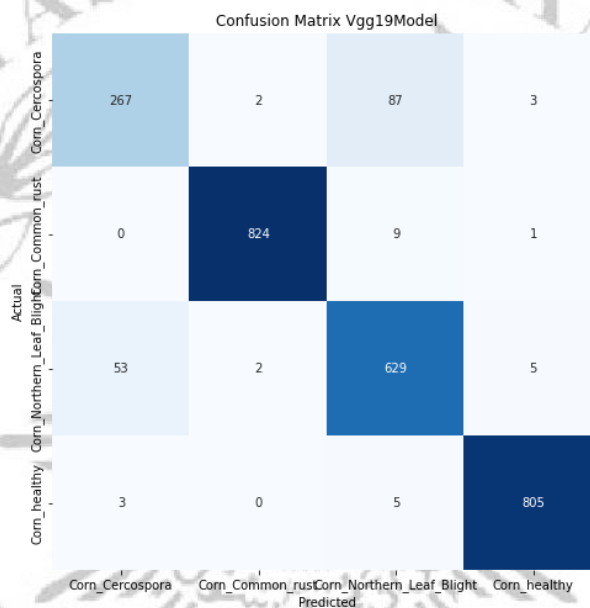
Gambar 4. 10 Confusion Matrix Alexnet

#### 4.7.6 VGG19

VGG19 adalah CNN dengan lapisan yang telah dilatih dan memiliki pemahaman yang baik tentang mendefinisikan gambar dalam hal warna, struktur, dan bentuk. VGG19 telah dilatih pada jutaan gambar yang beragam dengan klasifikasi yang sangat kompleks. Karena hal ini VGG19 pada penelitian ini termasuk kedalam model dengan akurasi yang cukup baik. Gambar 4.11 merupakan tingkat akurasi dari VGG19 dan Gambar 4.12 adalah confusion matrix dari VGG19.



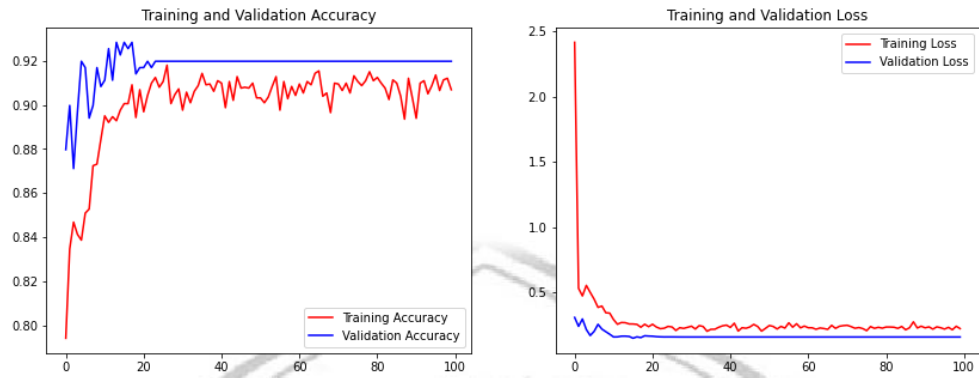
Gambar 4. 11 VGG19 Accuracy and Loss



Gambar 4. 12 Confusion Matrix VGG19

#### 4.7.7 Inception V3

Model Inception V3 menggunakan beberapa teknik untuk mengoptimalkan jaringan untuk adaptasi model yang lebih baik. Inception V3 memiliki jaringan yang lebih dalam dibandingkan dengan model Inception V1 dan V2, tetapi dengan kecepatan yang tetap terjaga. Inception memiliki beberapa keunggulan seperti pemanfaatan sumber daya komputasi yang efisien dengan peningkatan beban komputasi yang minimal untuk output kinerja yang tinggi. Kemampuan untuk mengekstrak fitur dari data input pada berbagai skala melalui pemanfaatan berbagai ukuran filter convolutional. Akurasi dari Inception V3 dapat dilihat pada Gambar 4.13 dan confusion matrix pada Gambar 4.14.



Gambar 4. 13 Inception V3 Accuracy and Loss

Confusion Matrix InceptionV3Model

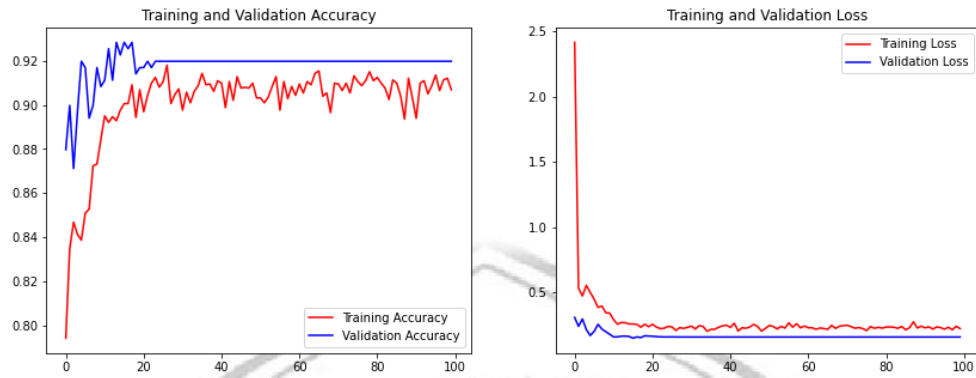
	Corn_Cercospora	Corn_Common_rust	Corn_Northern_Leaf_Blight	Corn_healthy
Actual				
Corn_Cercospora	272	2	82	3
Corn_Common_rust	0	825	9	0
Corn_Northern_Leaf_Blight	41	2	641	5
Corn_healthy	1	1	6	805
Predicted	Corn_Cercospora	Corn_Common_rust	Corn_Northern_Leaf_Blight	Corn_healthy

Gambar 4. 14 Confusion Matrix Inception V3

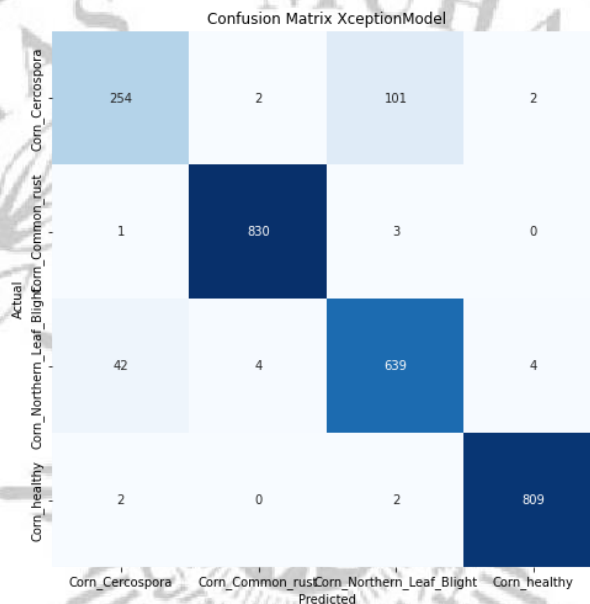
#### 4.7.8 Xception

Xception merupakan hasil pengembangan dari model Inception V3. Xception adalah Inception dengan konvolusi yang dapat dipisahkan secara mendalam. Xception pada penelitian ini mendapatkan akurasi yang tidak jauh dengan pendahulunya Inception V3 dengan tingkat akurasi sebesar 94%. Pada Gambar 4.15 merupakan akurasi dari Xception dan untuk confusion matrix dapat dilihat pada Gambar 4.16.





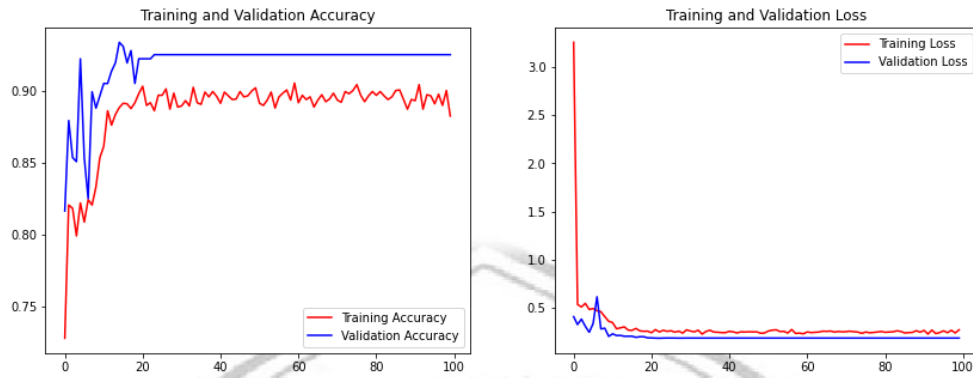
Gambar 4. 15 Xception Accuracy and Loss



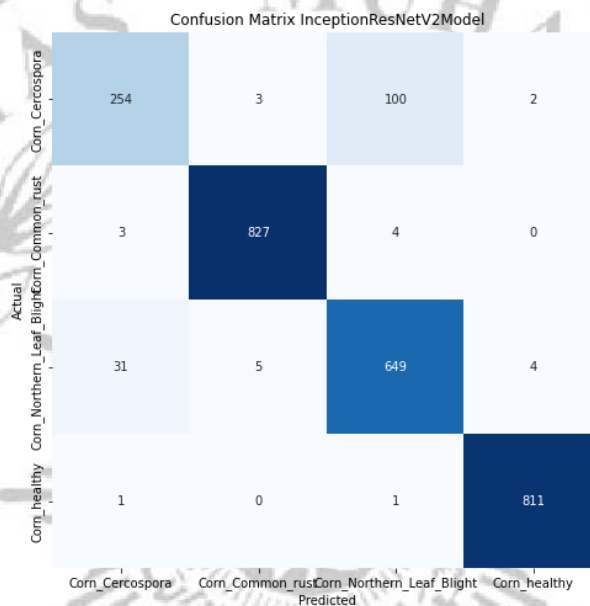
Gambar 4. 16 Confusion Matrix Xception

#### 4.7.9 Inception ResNet V2

Inception ResNet V2 adalah arsitektur saraf convolutional yang dibangun di atas keluarga arsitektur Inception tetapi menggabungkan koneksi residual (menggantikan tahap penyambungan filter dari arsitektur Inception). Inception dan ResNet merupakan salah satu model yang mengalami peningkatan kinerja dan biaya komputasi yang rendah. Jaringan ini memiliki kedalaman 164 layer dan dapat mengklasifikasikan gambar ke dalam 1000 kategori objek, seperti keyboard, mouse, pensil, dan binatang. Maka dari itu Inception ResNet V2 telah mempelajari representasi fitur yang kaya untuk berbagai gambar. Gambar 4.17 merupakan hasil akurasi dari model Inception ResNet V2 dan Gambar 4.18 merupakan confusion matrix dari Inception ResNet V2.



Gambar 4. 17 Inception ResNet V2 Accuracy and Loss

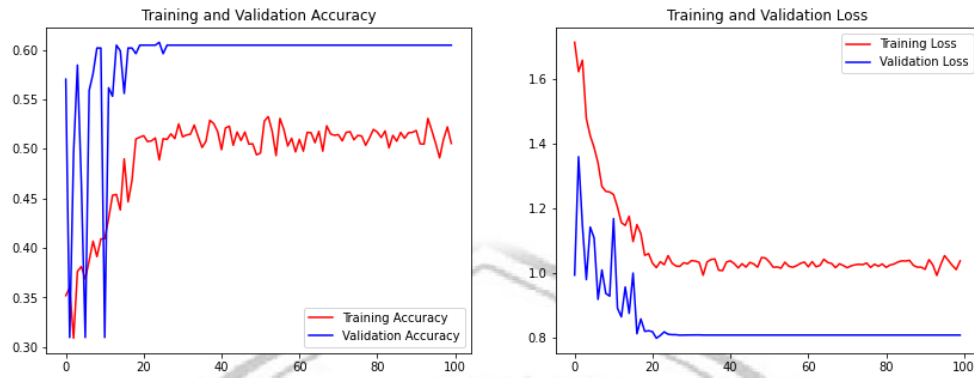


Gambar 4. 18 Confusion Matrix Inception ResNet V2

#### 4.7.10 ResNet50

Pada penelitian ini ResNet50 mendapatkan tingkat akurasi yang sangat rendah sebesar 60%. ResNet50 mempunyai satu masalah yaitu semakin dalam jaringan yang digunakan maka akan semakin lama waktu yang dibutuhkan untuk melakukan training. Hal ini membuat ResNet50 tidak layak digunakan pada real-world applications. ResNet50 tidak selalu bisa beradaptasi dengan data baru dan model ini memiliki tingkat kegagalan yang cukup tinggi. Terlihat pada Gambar 4.19 tingkat akurasi ResNet50 hanya mampu berada ditingkat 60% dan pada Gambar 4.20 merupakan confusion matrix dari model ResNet50, terlihat banyak sekali kesalahan prediksi dengan nilai aktualnya.





Gambar 4. 19 ResNet50 Accuracy and Loss

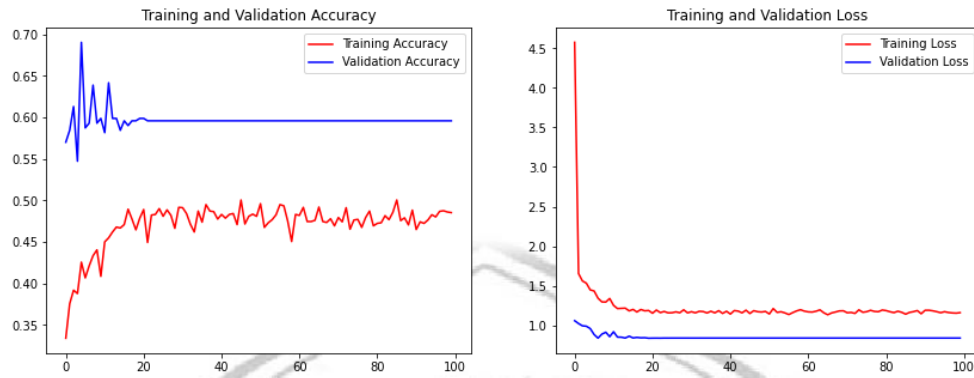
Confusion Matrix Resnet50Model

Actual \ Predicted	Corn_Cercospora	Corn_Common_rus	Corn_Northern_Leaf_Blight	Corn_healthy
Corn_Cercospora	0	76	36	247
Corn_Common_rus	0	814	9	11
Corn_Northern_Leaf_Blight	0	18	21	650
Corn_healthy	0	26	16	771

Gambar 4. 20 Confusion Matrix ResNet50

#### 4.7.11 ResNet101

Tidak jauh berbeda dengan ResNet50, ResNet101 juga mendapatkan nilai akurasi yang tergolong rendah yaitu 58%. ResNet101 tidak hanya lebih besar dari ResNet50, tetapi juga membutuhkan lebih banyak waktu untuk melakukan training dan lebih mudah terjadi overfitting. Dataset yang digunakan pada penelitian ini termasuk data yang tingkat kompleksitasnya tergolong tinggi, dan hal inilah juga yang menyebabkan tingkat akurasi ResNet101 menjadi lebih rendah yang dapat dilihat pada Gambar 4.21. Confusion matrix pada Gambar 4.22 memperlihatkan banyak sekali kesalahan prediksi yang dilakukan ResNet101.



Gambar 4. 21 ResNet101 Accuracy and Loss

Actual \ Predicted				
	Corn_Cercospora	Corn_Common_rus	Corn_Northern_Leaf_Blight	Corn_healthy
Corn_Cercospora	0	81	10	268
Corn_Common_rus	0	772	1	61
Corn_Northern_Leaf_Blight	0	16	1	672
Corn_healthy	0	33	1	779

Gambar 4. 22 Confusion Matrix ResNet101

#### 4.8 Hasil Klasifikasi Model

Setelah data dibagi menjadi data train dan data tes, semua model akan dilakukan proses training pada data training dengan masing-masing 100 epoch dengan batch size 32. Berdasarkan hasil training model, model klasifikasi terbaik didapatkan oleh DenseNet201 dengan tingkat akurasi sebesar 97%. DenseNet201 memiliki lapisan yang dimana setiap lapisannya memperoleh masukan tambahan dari semua lapisan sebelumnya dan meneruskan peta fiturnya sendiri ke semua lapisan berikutnya. Karena setiap lapisan menerima peta fitur dari semua lapisan sebelumnya, jaringan yang dibuat bisa lebih tipis dan jumlahnya bisa lebih sedikit sehingga mempunyai efisiensi komputasi dan efisiensi memori yang lebih baik. Hal inilah yang membuat DenseNet201 lebih baik dibandingkan dengan arsitektur yang lain dengan ukuran yang sama.

Model dengan tingkat akurasi tertinggi kedua dimiliki oleh ResNet152 V2 dengan tingkat akurasi sebesar 96% dan VGG16 pada peringkat ketiga mendapatkan akurasi sebesar 95%. Setelah proses training selesai dilakukan, didapatkan hasil akurasi semua model yang dapat dilihat pada Tabel 4.13.

*Tabel 4. 10 Tabel Akurasi*

<b>CNN Model</b>	<b>Accuracy (%)</b>
DenseNet201	97
ResNet152 V2	96
VGG16	95
MobileNet V2	95
Alexnet	94
VGG19	94
Inception V3	94
Xception	94
Inception ResNet V2	94
ResNet50	60
ResNet101	58

*Tabel 4. 11 Tabel Perbandingan*

<b>CNN Model</b>	<b>Accuracy (%)</b>
Sibiya et al. [4]	92.85
Syarief et al. [5]	95
Mohit et al. [26]	94
<b>Metode yang diusulkan</b>	<b>97</b>

Dengan melihat hasil akurasi yang sudah dilakukan, terbukti Data Augmentation yang diterapkan pada data training dapat meningkatkan hasil akurasi untuk kebutuhan klasifikasi gambar. Untuk hasil perbandingan pada penelitian sebelumnya, pada Tabel 4 metode yang diusulkan peneliti terbukti menghasilkan model yang lebih baik dari pada penelitian lainnya yang belum menggunakan Data Augmentation. Metode yang diusulkan memiliki tingkat akurasi sebesar 97%, lebih baik dari ketiga penelitian lain.

## **BAB 5**

### **PENUTUP**

#### **5.1 Kesimpulan**

Jagung memiliki banyak manfaat bagi tubuh manusia karena kandungan nutrisinya, yaitu diantara manfaatnya dapat menurunkan hipertensi sehingga bagi penderita penyakit jantung dapat melakukan pencegahan dini. Daun jagung memiliki beberapa penyakit daun seperti northern leaf blight, common rust, and cercospora. Pada tahap awal penyakit ini tidak dapat dibedakan. Pada penelitian ini akan dilakukan pengujian dan peningkatan performa menggunakan 11 model CNN yang telah dilakukan augmentasi data. Augmentasi data dapat mengurangi overfitting sehingga dapat menghasilkan model dengan performa terbaik. Metode yang peneliti usulkan telah dilakukan proses training dan testing data pada Plant's Village Dataset. DenseNet201 dengan data augmentation mendapatkan tingkat akurasi sebesar 97%.

#### **5.2 Saran**

Pada penelitian klasifikasi penyakit tanaman terutama pada tanaman jagung dapat ditingkatkan lagi dengan metode-metode lainnya yang mampu menghasilkan nilai yang lebih maksimal. Peneliti dapat meningkatkan akurasi lebih baik lagi dengan menambahkan optimisasi metode pada CNN arsitekturnya. DenseNet201 pada penelitian ini merupakan model yang paling cocok untuk fitur ekstraksi dan klasifikasi gambar pada penyakit daun jagung.

## DAFTAR PUSTAKA

- [1] P. Wang, E. Fan, and P. Wang, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning," *Pattern Recognition Letters*, vol. 141, pp. 61–67, Jan. 2021, doi: 10.1016/J.PATREC.2020.07.042.
- [2] X. Jiang, Y. Wang, W. Liu, S. Li, and J. Liu, "CapsNet, CNN, FCN: Comparative performance evaluation for image classification," *International Journal of Machine Learning and Computing*, vol. 9, no. 6, pp. 840–848, 2019, doi: 10.18178/IJMLC.2019.9.6.881.
- [3] X. Lei, H. Pan, and X. Huang, "A dilated cnn model for image classification," *IEEE Access*, vol. 7, pp. 124087–124095, 2019, doi: 10.1109/ACCESS.2019.2927169.
- [4] M. Sibiya and M. Sumbwanyambe, "A Computational Procedure for the Recognition and Classification of Maize Leaf Diseases out of Healthy Leaves Using Convolutional Neural Networks," Feb. 2019, doi: 10.20944/PREPRINTS201902.0203.V1.
- [5] M. Syarief and W. Setiawan, "Convolutional neural network for maize leaf disease image classification," *TELKOMNIKA (Telecommunication Comput. Electron. Control.)*, vol. 18, no. 3, pp. 1376–1381, Jun. 2020, doi: 10.12928/TELKOMNIKA.V18I3.14840.
- [6] J. Chen *et al.*, "Medical image segmentation and reconstruction of prostate tumor based on 3D AlexNet," *Computer Methods and Programs in Biomedicine*, vol. 200, p. 105878, Mar. 2021, doi: 10.1016/J.CMPB.2020.105878.
- [7] M. Ji, L. Zhang, and Q. Wu, "Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks," *Information Processing in Agriculture*, vol. 7, no. 3, pp. 418–426, Sep. 2020, doi: 10.1016/J.INPA.2019.10.003.
- [8] G. S and L. M, "Breast cancer detection using synthetic mammograms from generative adversarial networks in convolutional neural networks," *Journal of medical imaging (Bellingham, Wash.)*, vol. 6, no. 3, p. 1, Mar. 2019, doi: 10.1117/1.JMI.6.3.031411.
- [9] F. F. Ting, Y. J. Tan, and K. S. Sim, "Convolutional neural network improvement for breast cancer classification," *Expert Systems with Applications*, vol. 120, pp. 103–115, Apr. 2019, doi: 10.1016/J.ESWA.2018.11.008.
- [10] G. Geetharamani and A. P. J., "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," *undefined*, vol. 76, pp. 323–338, Jun. 2019, doi: 10.1016/J.COMPELECENG.2019.04.011.
- [11] A. Hekler *et al.*, "Superior skin cancer classification by the combination of human and artificial intelligence," *European Journal of Cancer*, vol. 120, pp. 114–121, Oct. 2019, doi: 10.1016/J.EJCA.2019.07.019.
- [12] L. D. Medus, M. Saban, J. v. Francés-Víllora, M. Bataller-Mompeán, and A. Rosado-Muñoz, "Hyperspectral image classification using CNN: Application to industrial food packaging," *Food Control*, vol. 125, p. 107962, Jul. 2021, doi: 10.1016/J.FOODCONT.2021.107962.
- [13] W. Setiawan, M. I. Utoyo, R. Rulaningtyas, and A. Wicaksono, "Semantic segmentation of artery-venous retinal vessel using simple convolutional neural network," *IOP Conference Series: Earth and Environmental Science*, vol. 243, no. 1, Apr. 2019, doi: 10.1088/1755-1315/243/1/012021.
- [14] V. H. Trong, Y. Gwanghyun, K. Jinyoung, and P. T. Bao, "Yielding multi-fold training strategy for image classification of imbalanced weeds," *Applied Sciences (Switzerland)*, vol. 11, no. 8, Apr. 2021, doi: 10.3390/AP11083331.

- [15] K. Thenmozhi and U. Srinivasulu Reddy, "Crop pest classification based on deep convolutional neural network and transfer learning," *Computers and Electronics in Agriculture*, vol. 164, Sep. 2019, doi: 10.1016/J.COMPAG.2019.104906.
- [16] W. Setiawan, M. I. Utoyo, and R. Rulaningtyas, "Classification of neovascularization using convolutional neural network model," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 1, pp. 463–472, Feb. 2019, doi: 10.12928/TELKOMNIKA.V17I1.11604.
- [17] S. H. Wang *et al.*, "Alcoholism identification based on an Alexnet transfer learning model," *Frontiers in Psychiatry*, vol. 10, no. APR, 2019, doi: 10.3389/FPSYT.2019.00205.
- [18] S. Samir, E. Emary, K. El-Sayed, and H. Onsi, "Optimization of a pre-trained AlexNet model for detecting and localizing image forgeries," *Information (Switzerland)*, vol. 11, no. 5, May 2020, doi: 10.3390/INFO11050275.
- [19] C. Sitaula and M. B. Hossain, "Attention-based VGG-16 model for COVID-19 chest X-ray image classification," *Applied Intelligence*, vol. 51, no. 5, p. 1, May 2021, doi: 10.1007/S10489-020-02055-X.
- [20] Q. Guan *et al.*, "Deep convolutional neural network VGG-16 model for differential diagnosing of papillary thyroid carcinomas in cytological images: a pilot study," *Journal of Cancer*, vol. 10, no. 20, pp. 4876–4882, 2019, doi: 10.7150/JCA.28769.
- [21] M. Ye *et al.*, "A Lightweight Model of VGG-16 for Remote Sensing Image Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 6916–6922, 2021, doi: 10.1109/JSTARS.2021.3090085.
- [22] G. P. Kusuma, A. Jonathan, and P. Lim, "Emotion recognition on FER-2013 face images using fine-tuned VGG-16," *Advances in Science, Technology and Engineering Systems*, vol. 5, no. 6, pp. 315–322, 2020, doi: 10.25046/AJ050638.
- [23] A. Majid, M. A. Khan, M. Yasmin, A. Rehman, A. Yousafzai, and U. Tariq, "Classification of stomach infections: A paradigm of convolutional neural network along with classical features fusion and selection," *Microscopy Research and Technique*, vol. 83, no. 5, pp. 562–576, May 2020, doi: 10.1002/JEMT.23447.
- [24] V. Maeda-Gutiérrez *et al.*, "Comparison of convolutional neural network architectures for classification of tomato plant diseases," *Applied Sciences (Switzerland)*, vol. 10, no. 4, Feb. 2020, doi: 10.3390/APP10041245.
- [25] S. S. Yadav and S. M. Jadhav, "Deep convolutional neural network based medical image classification for disease diagnosis," *Journal of Big Data*, vol. 6, no. 1, pp. 1–18, Dec. 2019, doi: 10.1186/S40537-019-0276-2/TABLES/16.
- [26] M. Agarwal, V. K. Bohat, M. D. Ansari, A. Sinha, S. K. Gupta and D. Garg, "A Convolution Neural Network based approach to detect the disease in Corn Crop," 2019 IEEE 9th International Conference on Advanced Computing (IACC), 2019, pp. 176–181, doi: 10.1109/IACC48062.2019.8971602.
- [27] Zhang, Y., Wa, S., Liu, Y., Zhou, X., Sun, P., & Ma, Q. (2021). High-Accuracy Detection of Maize Leaf Diseases CNN Based on Multi-Pathway Activation Function Module. *Remote Sensing 2021, Vol. 13, Page 4218, 13*(21), 4218. <https://doi.org/10.3390/RS13214218>

