

**KLASIFIKASI PENYAKIT PADA DAUN JERUK  
MENGUNAKAN CONVOLUTIONAL NEURAL NETWORK  
(CNN) DENGAN ARSITEKTUR XCEPTION**

**LAPORAN TUGAS AKHIR**

Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang



**DISUSUN OLEH :**

**MOH. REZA PALEVIE**

**32602000042**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM SULTAN AGUNG  
SEMARANG**

**2024**

***FINAL PROJECT***

***DISEASE CLASSIFICATION IN CITRUS LEAVES USING  
CONVOLUTIONAL NEURAL NETWORK (CNN) WITH XCEPTION  
ARCHITECTURE***

*Proposed to complete the requirement to obtain a bachelor's degree (S1)  
at Informatics Engineering Departement of Industrial Technology Faculty  
Sultan Agung Islamic University*



**MOH. REZA PALEVIE**

**32602000042**

***MAJORING OF INFORMATICS ENGINEERING  
INDUSTRIAL TECHNOLOGY FACULTY  
SULTAN AGUNG ISLAMIC UNIVERSITY  
SEMARANG***

**2024**

**LEMBAR PENGESAHAN  
TUGAS AKHIR**

**KLASIFIKASI PENYAKIT PADA DAUN JERUK  
MENGUNAKAN CONVOLUTIONAL NEURAL NETWORK  
(CNN) DENGAN ARSITEKTUR XCEPTION**

**MOH. REZA PALEVIE  
NIM 32602000042**

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir  
Program Studi Teknik Informatika  
Universitas Islam Sultan Agung  
Pada tanggal : 10 Juni 2025

**TIM PENGUJI UJIAN SARJANA :**

**Andi Riansyah, ST, M. Kom.**

NIK. 0609108802

(Ketua Penguji)

10 Juni 2025

**Arief Marwanto, Ph.D**

NIK. 0628097501

(Anggota Penguji)

10 Juni 2025

**Moch Taufik MIT**

NIK. 0622037502

(Pembimbing)

10 Juni 2025

Semarang, 10 Juni 2025

Mengetahui,

Kaprodi Teknik Informatika  
Universitas Islam Sultan Agung

**Moch Taufik MIT**

NIK. 0622037502

## SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Moh. Reza Palevie

NIM : 32602000042

Judul Tugas Akhir : Klasifikasi Penyakit Pada Daun Jeruk Menggunakan  
Convolutional Neural Network (CNN) dengan Arsitektur  
Xception

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 10 Juni 2025

Yang Menyatakan,



Moh. Reza Palevie

## PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Moh. Reza Palevie  
NIM : 32602000042  
Program Studi : Teknik Informatika  
Fakultas : Teknologi Industri

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul :  
KLASIFIKASI PENYAKIT PADA DAUN JERUK MENGGUNAKAN  
CONVOLUTIONAL NEURAL NETWORK (CNN) DENGAN ARSITEKTUR  
XCEPTION

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan  
Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan  
pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan  
akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta.  
Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari  
terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka  
segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi  
tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 10 Juni 2025

Yang menyatakan,



Moh. Reza Palevie

**Moch Taufik MIT**  
NIK. 0622037502  
**KATA PENGANTAR**

Dengan mengucapkan syukur alhamdulillah atas kehadiran Allah SWT yang telah memberikan rahmat dan hidayahnya kepada penulis sehingga dapat menyelesaikan Tugas Akhir dengan judul “Klasifikasi Penyakit Pada Daun Jeruk Menggunakan Convolutional Neural Network (CNN) dengan Arsitektur Xception” ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis. Oleh karena itu, saya selaku penulis mengucapkan terima kasih kepada :

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H., yang mengizinkan penulis menimba ilmu dikampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Hj. Novi Marlyana S.T., M.T., IPU., ASEAN.Eng.
3. Dosen Pembimbing, Moch Taufik, ST, MIT yang telah membimbing dan memberikan banyak nasehat dan saran.
4. Orang tua dan keluarga penulis yang menjadi *support system* dan mengizinkan untuk menyelesaikan laporan ini.
5. Teman-teman mirza kos yang menjadi support dalam menyelesaikan pembuatan aplikasi mobile.
6. Dan kepada semua pihak yang tidak dapat penulis sebutkan satu persatu atas doa, *support* serta bantuannya.

Dengan segala kerendahan hati, penulis menyadari masih terdapat banyak kekurangan dari segi kualitas atau kuantitas maupun dari ilmu pengetahuan dalam penyusunan laporan sehingga penulis mengharapkan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan ini di masa mendatang.

Semarang, 21 Agustus 2024

**Moh. Reza Palevie**

## DAFTAR ISI

COVER .....	i
LEMBAR PENGESAHAN .....	iii
TUGAS AKHIR.....	iii
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH .....	v
Moch Taufik MIT .....	vi
KATA PENGANTAR.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR .....	x
DAFTAR TABEL .....	xi
ABSTRAK .....	xiii
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	3
1.3 Pembatasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	3
<b>BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....</b>	<b>5</b>
2.1 Tinjauan Pustaka.....	5
2.2 Dasar Teori .....	10
2.2.1 Penyakit Daun Jeruk.....	10
2.2.2 <i>Machine Learning</i> .....	15
2.2.3 <i>Deep Learning</i> .....	17
2.2.4 <i>Arsitektur Xception</i> .....	18
2.2.5 <i>Convolutional Neural Network</i> .....	20
2.2.6 <i>TensorFlow</i> .....	24
2.2.7 <i>Keras</i> .....	24
2.2.8 <i>TensorFlow Lite</i> .....	25
<b>BAB III METODE PENELITIAN .....</b>	<b>26</b>

3.1	Metode Penelitian .....	26
3.1.1	Studi <i>Literature</i> .....	26
3.1.2	Pengumpulan Data ( <i>Data Collecting</i> ) .....	26
3.1.3	<i>Data Cleaning</i> .....	27
3.1.4	<i>Data Preprocessing</i> .....	28
3.1.5	<i>Training</i> .....	30
3.1.6	Pengujian pada Arsitektur <i>Xception</i> .....	30
3.1.7	<i>Deployment Model</i> .....	31
3.1.8	Pembuatan Aplikasi <i>Mobile</i> .....	31
3.2	Analisis Kebutuhan .....	32
3.2.1	Tips Penggunaan Aplikasi.....	32
3.2.2	Unggah Gambar.....	33
3.2.3	<i>Resize Gambar</i> .....	33
3.2.4	Melakukan Prediksi.....	33
3.2.5	Menampilkan Hasil Prediksi.....	33
3.3	Analisis Sistem .....	33
3.4	Desain <i>User Interface</i> .....	34
3.4.1	Halaman <i>Splash Screen</i> .....	35
3.4.2	Halaman Klasifikasi Penyakit Daun Jeruk .....	36
3.4.3	Halaman Profil.....	37
<b>BAB IV HASIL DAN ANALISIS PENELITIAN.....</b>		<b>38</b>
4.1	Hasil dan Analisis Model CNN.....	38
4.1.1	Pengujian Model <i>Xception</i> dengan Hyperparamater.....	38
4.1.2	Pengujian dengan Optimizer SGD .....	39
4.1.3	Hasil Akurasi.....	39
4.1.4	Hasil <i>Testing Model CNN</i> .....	40
4.2	Cara Kerja Sistem Klasifikasi atau Deteksi.....	44
4.2.1.	Unggah Gambar.....	44
4.2.2.	Konversi Gambar ke <i>TensorImage</i> .....	44
4.2.3.	Konversi <i>TensorImage</i> ke <i>TensorBuffer</i> .....	44
4.2.4.	Proses Klasifikasi .....	44
4.2.5.	Hasil Klasifikasi .....	45
4.3	Implementasi <i>User Interface (UI)</i> .....	45

4.3.1	Halaman <i>SplashScreen</i> .....	45
4.3.2	Halaman <i>Home</i> .....	46
4.4	Pengujian Sistem.....	48
4.5	Hasil <i>Testing</i> Aplikasi.....	49
<b>BAB V KESIMPULAN DAN SARAN .....</b>		<b>53</b>
5.1	Kesimpulan.....	53
5.2	Saran .....	54
<b>DAFTAR PUSTAKA.....</b>		<b>55</b>
<b>LAMPIRAN.....</b>		<b>62</b>



## DAFTAR GAMBAR

Gambar 2. 1 Penyakit Powdery Mildew.....	11
Gambar 2. 2 Penyakit Mealybug .....	12
Gambar 2. 3 Penyakit Citrus Vein Phloem Degeneration.....	13
Gambar 2. 4 Citrus Cancer .....	14
Gambar 2. 5 Penyakit Spiny Whiteflt.....	15
Gambar 2. 6 Original Depthwise Convolution .....	18
Gambar 2. 7 Modified Depthwise Convolution.....	19
Gambar 2. 8 Arsitektur Xception.....	20
Gambar 2. 9 Arsitektur CNN.....	21
Gambar 2. 10 Convolutional layer .....	22
Gambar 2. 11 Pooling Layer.....	22
Gambar 2. 12 Cross Channel Normalization .....	23
Gambar 3. 1 Alur Deployment Model .....	31
Gambar 3. 2 flowchart aplikasi.....	32
Gambar 3. 3 Tampilan Ui Splash Screen .....	35
Gambar 3. 4 Halaman Deteksi Penyakit Daun Jeruk.....	36
Gambar 3. 5 Halamn Profil .....	37
Gambar 4. 1 Confusion Matriks Optimizer SGD .....	41
Gambar 4. 5 Halaman SplashScreen .....	46



## DAFTAR TABEL

Tabel 2. 1 Kelompok penyakit daun jeruk .....	<b>Error! Bookmark not defined.</b>
Tabel 3. 1 Konfigurasi Pengujian Model Arsitektur Xception....	<b>Error! Bookmark not defined.</b>
Tabel 4. 1 Plot Accuracy dan Loss .....	<b>Error! Bookmark not defined.</b>
Tabel 4. 2 Model Terbaik Berdasarkan Hasil Akurasi.....	<b>Error! Bookmark not defined.</b>
Tabel 4. 3 Accuracy, Precision, Recall, & F1-score Konfigurasi 1 .....	<b>Error! Bookmark not defined.</b>
Tabel 4. 4 Precision, Recall, & F1-score Kelas ( Konfigurasi 1).....	41
Tabel 4. 5 TP, FP, FN, & TN Konfigurasi 1 .....	<b>Error! Bookmark not defined.</b>
Tabel 4. 6 Perhitungan Manual 1 (Konfigurasi 1)..	<b>Error! Bookmark not defined.</b>
Tabel 4. 7 Perhitungan Manual 2 (Konfigurasi 1)..	<b>Error! Bookmark not defined.</b>
Tabel 4. 8 Accuracy, Precision, Recall, & F1-score Konfigurasi 2.....	<b>Error! Bookmark not defined.</b>
Tabel 4. 9 Precision, Recall, & F1-score Kelas ( Konfigurasi 2)	<b>Error! Bookmark not defined.</b>
Tabel 4. 10 TP, FP, FN, & TN Konfigurasi 2 .....	<b>Error! Bookmark not defined.</b>
Tabel 4. 11 Perhitungan Manual 1 (Konfigurasi 2)	<b>Error! Bookmark not defined.</b>
Tabel 4. 12 Perhitungan Manual 2 (Konfigurasi 2)	<b>Error! Bookmark not defined.</b>
Tabel 4. 13 Accuracy, Precision, Recall, & F1-score Konfigurasi 3.....	<b>Error! Bookmark not defined.</b>
Tabel 4. 14 Precision, Recall, & F1-score Kelas ( Konfigurasi 3).....	<b>Error! Bookmark not defined.</b>
Tabel 4. 15 TP, FP, FN, & TN Konfigurasi 3 .....	<b>Error! Bookmark not defined.</b>
Tabel 4. 16 Perhitungan Manual 1 (Konfigurasi 3)	<b>Error! Bookmark not defined.</b>
Tabel 4. 17 Perhitungan Manual 2 (Konfigurasi 3)	<b>Error! Bookmark not defined.</b>
Tabel 4. 18 Accuracy, Precision, Recall, & F1-score Konfigurasi 4.....	<b>Error! Bookmark not defined.</b>

Tabel 4. 19 Precision, Recall, & F1-score Kelas (Konfigurasi 4).....	<b>Error!</b>
<b>Bookmark not defined.</b>	
Tabel 4. 20 TP, FP, FN, & FN Konfigurai 4.....	<b>Error! Bookmark not defined.</b>
Tabel 4. 21 Perhitungan Manual 1 (Konfigurasi 4)	<b>Error! Bookmark not defined.</b>
Tabel 4. 22 Perhitungan Manual 2 (Konfigurasi 4)	<b>Error! Bookmark not defined.</b>
Tabel 4. 23 Hasil Blackbox Testing .....	<b>Error! Bookmark not defined.</b>
Tabel 4. 24 Hasil Testing Prediksi Aplikasi .....	50



## ABSTRAK

Penyakit pada daun jeruk menjadi penyebab Utama diantara banyaknya factor kegagalan panen dalam budidaya tanaman jeruk. akibat yang ditimbulkan bisa merugikan petani maupun pembudidaya tanaman jeruk dalam segi produksi buah jeruk. macam penyakit yang ada pada tanaman jeruk sangat bervariasi dan mempunyai penyebab yang berbeda-beda. factor lingkungan seperti media tanam, salahnya perawatan dalam budidaya jeruk, serta serangan hama, virus, serta pathogen yang menular menyebabkan kegagalan panen bagi pembudidaya jeruk. diagnosis dini tentang penularan hama dan penyakit serta penggunaan varietas yang tahan akan penyakit merupakan solusi yang paling efektif untuk mengendalikan kegagalan panen pada budidaya jeruk. gejala yang umum terjadi pada tanaman jeruk yang terinfeksi penyakit dapat dilihat dari anomaly pada daun tanaman jeruk. hal ini bisa dibedakan dari warna daun serta kondisi tulang daun pada tanaman jeruk. dengan demikian penanganan awal dalam upaya pencegahan penyakit pada daun jeruk bisa menjadi solusi untuk mengurangi terjadinya gagal panen dalam budidaya tanaman jeruk. salah satu cara pencegahan yang bisa dilakukan antara lain dengan melakukan klasifikasi daun tanaman jeruk menggunakan pembelajaran mesin dengan metode Convolutional Neural Network. Penelitian ini menggunakan metode algoritma CNN dengan arsitektur Xception yang kemudian implementasikan dalam aplikasi android yang nantinya akan dibuat menggunakan Bahasa kotlin. dengan menggunakan 2400 data yang terbagi menjadi enam kelas, model mampu menghasilkan klasifikasi citra gambar dengan baik. hal ini ditunjukkan dengan nilai accuracy 98%, val accuracy 83%, loss 9%, dan val loss 12%..

## **ABSTRACT**

*Citrus leaf disease is the main cause among the many factors of crop failure in citrus cultivation. the consequences can be detrimental to farmers and cultivators of citrus plants in terms of citrus fruit production. the kinds of diseases that exist in citrus plants are very varied and have different causes. environmental factors such as planting media, wrong care in citrus cultivation, and infectious pests, viruses, and pathogens cause crop failure for citrus cultivators. early diagnosis of pest and disease transmission and the use of disease-resistant varieties are the most effective solutions to control crop failure in citrus cultivation. symptoms that commonly occur in citrus plants infected with disease can be seen from anomalies in the leaves of citrus plants. this can be distinguished from the color of the leaves and the condition of the leaf bones in citrus plants. thus early handling in an effort to prevent disease in citrus leaves can be a solution to reduce the occurrence of crop failure in citrus cultivation. one way of prevention that can be done, among others, is by classifying the leaves of citrus plants using machine learning with the Convolutional Neural Network method. This research uses the CNN algorithm method with Xception architecture which is then implemented in an android application which will be created using Kotlin language. by using 2400 data divided into six classes, the model is able to produce image classification well. this is shown with 98% accuracy value, 83% val accuracy, 9% loss, and 12% val loss.*

*Keywords: Citrus Diseases Classification, Convolutional Neural Network, Kotlin for Mobile, Xception, Tensorflow Lite.*

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

*Citrus* (genus *Citrus L.*) atau lebih dikenal dengan jeruk, yang termasuk genus citrus dalam keluarga *Rutaceae* dan sub-keluarga *Aurantioideae* berdasarkan klasifikasi botani, saat ini merupakan salah satu tanaman budidaya paling penting secara ekonomi dalam area yang luas dan nilai produksi di seluruh dunia. Jeruk dipercaya berasal dari daerah Baratdaya Himalaya China, Timurlaut India, utara Burma, dan sekarang tersebar di lebih dari 140 negara. Berdasarkan Statistika dari *Food and Agriculture Organization of the United Nations* (FAO), produksi jeruk secara global sebanyak 144 juta ton pada tahun 2019, dimana dengan 9,89 hektar area tanam. hasil dan budidaya jeruk di China mencapai masing-masing 38 juta ton dan 2,88 juta hektar menempati posisi pertama di seluruh dunia. *Sweet orange* (*Citrus sinensis*), *sour orange* (*C. aurantium*), *mandarin* (*C. reticulata*), *grapefruit* (*C. paradisi*), *pummelo* (*C. grandis*), *lemon* (*C. limon*), *citron* (*C. medica*), *lime* (*C. aurantifolia*), *kumquat* (*C. japonica*), dan hibrida diketahui sebagai spesies jeruk komersial yang sangat penting (Liu dkk, 2022).

Meskipun tanaman jeruk memberikan kontribusi yang signifikan pada produksi buah secara global, fokus penelitian ini adalah pada kesehatan tanaman jeruk, terutama penyakit yang terdapat pada daun jeruk. Hal ini ditunjukkan dengan organisme parasit yang menyerang tanaman jeruk seperti penyakit jamur (*sweet orange scab*, *citrus black spot*, *powdery mildew*), penyakit bakteri (penyakit *pierce*, *citrus variegated chlorosis*, *citrus greening*, *citrus canker*) dan penyakit virus (*citrus tristeza*, *citrus ringspot*, dll.) serta nematoda yang kompleks membatasi produksi jeruk, nilai gizi, dan kualitas pasar (Ahmed et al., 2020). Selain gangguan kesehatan yang disebabkan organisme parasite penyakit tanaman jeruk juga dapat disebabkan oleh serangan hama seperti *Mealybug* (*Hemiptera: Pseudococcidae*) yang menyebabkan kerontokan premature pada daun jeruk, keterlambatan pertumbuhan dan kematian pada tanaman jeruk (Ibrahim dkk, 2021). Terdapat juga serangga *spiney whitefly* yang menyebabkan kerusakan secara langsung maupun

tidak langsung. Kerusakan langsung, seperti Penutupan stomata oleh *honey dew* yang dikeluarkan oleh nimfa dan *shoot dew* yang tumbuh pada lapisan *honey dew*; Bercak-bercak klorotik pada daun yang mengakibatkan kerusakan jaringan akibat tusukan stylet, terbentuknya pigmen antosianin; dan daun rontok sehingga dapat menghambat pertumbuhan tanaman (Merthaningsih dkk, 2020). Pada kasus yang berbeda terdapat serangan penyakit pada tanaman jeruk yang disebabkan oleh penyakit *citrus vein phloem degeneration* (CVPD) atau dikenal dengan nama internasional sebagai *Huanglongbing* yang pertamakali di temukan di Indonesia pada tahun 1964, menyebabkan gangguan pada pertumbuhan tanaman jeruk (Tuwo dkk., 2024). Pada wilayah subtropis dan beriklim sedang seperti wilayah Asia Tenggara terdapat penyakit *citrus cancer* (CC) yang disebabkan oleh virus *Xanthomonas citri subsp. citri* (Xcc), yang merupakan penyakit paling merusak serta menular yang menyerang seluruh spesies kultivar pada tanaman jeruk (Naqvi dkk., 2022).

Penerapan teknologi dalam sektor *agriculture* atau pertanian menjadi salah satu cara untuk mendukung serta mengatasi masalah dan tantangan ini. Saat ini penerapan teknologi informatika, seperti kecerdasan artifisial atau *artificial intelligence* (AI) (Tuwo et dkk, 2024), internet of things (Song dkk, 2024), dan big data (Yang & Yu, 2024) dimanfaatkan untuk mengolah data-data dalam sektor pertanian dan menghasilkan informasi yang dapat digunakan untuk meningkatkan produksi serta kualitas pangan.

Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sebuah sistem klasifikasi penyakit pada daun jeruk yang dapat membantu di sektor pertanian sehingga dapat membantu semua kalangan terutama petani jeruk dalam upaya pencegahan penularan penyakit pada tanaman jeruk.

Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sebuah sistem deteksi penyakit daun pada guna memperoleh hasil produksi yang efisien bagi setiap orang.

## 1.2 Perumusan Masalah

1. Bagaimana sistem algoritma *convolutional neural network* dapat diterapkan di sektor *agriculture* atau pertanian untuk klasifikasi penyakit pada daun tanaman jeruk.

## 1.3 Pembatasan Masalah

Adapun batasan masalah dari penulisan Tugas Akhir sebagai berikut :

1. Penelitian ini berfokus klasifikasi penyakit *citrus vein phloem degeneration*(CVPD), *powdery mildew*, *mealybug*, *citrus canker*, dan *spiny whitefly* pada tanaman jeruk melalui gejala yang timbul pada daun tanaman jeruk..

## 1.4 Tujuan

Tujuan dari penelitian tugas akhir ini adalah mengembangkan sistem klasifikasi penyakit yang dapat dilihat melalui gejala pada daun tanaman jeruk menggunakan algoritma *Convolutional neural network*.

## 1.5 Manfaat

Manfaat yang didapatkan dari penelitian ini adalah agar supaya dapat mengimplementasikan model *Xception* dalam mengklasifikasikan penyakit yang terdapat dalam daun tanaman jeruk dengan hasil akurasi yang baik

## 1.6 Sistematika Penulisan

Sistematika penulisan yang akan digunakan pada pembuatan laporan tugas akhir adalah sebagai berikut :

BAB I : Pada bab 1 mengutarakan latar belakang pemilihan judul, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian dan sistematika penulisan.

BAB II : Pada bab II memuat penelitian – penelitian sebelumnya dan dasar teori untuk membantu dalam memahami teori yang berhubungan dengan

algoritma *Convolutional Neural Network* dengan model arsitektur *Xception* sehingga dapat menyelesaikan penelitian ini.

BAB III : Pada bab III mengungkapkan proses tahapan – tahapan penelitian yang dimulai dari mendapatkan data hingga proses klasifikasi data yang ada.

BAB IV : Pada bab IV mengungkapkan hasil klasifikasi dari penelitian yang menggunakan *Convolutional Neural Network* dengan model arsitektur *Xception* beserta pengujian klasifikasinya.

BAB V : Pada bab V memaparkan kesimpulan dari keseluruhan proses penelitian dari awal hingga akhir.



## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Dalam penelitian berjudul “Klasifikasi Hama Serangga pada Pertanian Menggunakan Metode *Convolutional Neural Network*” yang dilakukan oleh Harry Ramza, pada tahun 2023. Penelitian ini dilakukan untuk mengklasifikasikan jenis hama serangga menggunakan model pelatihan CNN dengan arsitektur *Xception*. Preprocessing data dilakukan dalam penelitian ini dengan menyeleksi dataset awal yang sebelumnya berjumlah 1591 citra menjadi 1363 citra serta melakukan *resizing* untuk semua citra yang terdapat dalam *dataset* menjadi  $224 \times 224$  *pixel* dengan format RGB dan dinormalisasi dengan pembagian 255 sehingga setiap *pixel* akan memiliki nilai antara 0 hingga 1 guna untuk mempercepat proses komputasi. Hasil pelatihan pada penelitian mencapai 93,81% untuk akurasi *training* dan 81,75% untuk akurasi validasi dengan menggunakan *optimizer Adam* serta menggunakan tingkat pembelajaran *Learning rate* sebesar 0,0001 pada *epoch* 50. Model dalam penelitian tersebut memiliki performa yang baik dalam mengenali dan mengklasifikasi hama serangga yang bisa berguna dalam upaya pengendalian hama dalam pertanian (Akram dkk, 2023).

Terdapat penelitian sebelumnya dengan judul “Analisis Performa Akurasi Klasifikasi Citra Jenis Sayur Selada menggunakan Arsitektur VGG16, Xception dan NasNetMobile” yang dilakukan oleh Nurafiya dan Albert Yokobus Chandra. Penelitian ini menggunakan algoritma CNN dengan arsitektur VGG16. Pengumpulan data citra gambar yang dilakukan dalam penelitian tersebut sebanyak 5000 citra gambar sayur selada yang terbagi dalam 5 kelas jenis sayur selada. Kemudian untuk membantu menilai kemampuan model dalam membuat prediksi, data dibagi menjadi *data training*, *data test*, dan *data validation* dengan presentase pembagian data masing-masing 70%, 20%, dan 10%. Pada pemodelan menggunakan hyperparameter seperti *input size* yang digunakan dalam model pembelajaran tersebut memiliki nilai  $128 \times 128$  *pixel* yang digunakan pada arsitektur *xception*. Kemudian dalam pelatihan model dilakukan *epoch* sebanyak 200 kali

dengan nilai *batch size* 64 yang berarti pada setiap *epoch* dilakukan iterasi sebanyak 15 kali. Dalam pelatihan pemodelan menggunakan *Xception* menunjukkan *accuracy*, *precision*, dan *recall* masing-masing 97.62%, 97.8%, dan 97.9%(Vgg dkk., 2024).

Kemudian terdapat penelitian dengan judul “Klasifikasi Daun Teh Klon Seru GMB Menggunakan Convolutional Neural Network dengan Arsitektur VGG16 dan *Xception*”. Teknik *rescale* dilakukan dengan cara mengubah skala gambar [0, 255] menjadi [0, 1] yang merupakan bagian dari preprocessing gambar menggunakan Pustaka keras data bertujuan untuk membantu memperkecil data loss dan menambah akurasi ketika pelatihan *data set*. *Data set* yang bersumber dari *Kaggle* terdiri dari enam kelas. Dalam proses preprocessing, dataset yang berjumlah 2434 citra gambar dibagi menjadi *data training* yang berjumlah 2190 citra gambar, data validasi berjumlah 122 citra gambar, serta data testing berjumlah 122 citra gambar. Selanjutnya augmentasi yang digunakan dalam *preprocessing data* antara lain rotasi 15 derajat, *rescale* 1./255, dan digeser acak secara horizontal dan vertical 10%, transformasi shear 20%, zoom acak 20%, dan di flipping secara horizontal. Model arsitektur menggunakan input gambar berukuran 128×128 *pixel* dengan 3 saluran warna. Dalam pemodelan dilakukan uji coba tanpa dilakukan augmentasi data dan didapatkan nilai akurasi data uji tertinggi pada *epoch* ke-13 sementara *loss* pada pengujian data menurun dan tetap stabil mendekati nol, mencapai nilai akhir sebesar 0.0766. hasil performa model menunjukkan nilai akurasi sebesar 98%. Model tanpa augmentasi data lebih efektif dalam konteks *dataset* terbatas dimana augmentasi dapat menyebabkan *overfitting* dengan membuat model menyesuaikan diri dengan data pelatihan dan tidak mampu beradaptasi dengan data baru secara efektif(Rinaldi dkk., 2024).

Pada penelitian berjudul “Klasifikasi Kualitas Permukaan Jalan Raya Menggunakan Metode CNN Berbasis Arsitektur *Xception*”. *Dataset* yang berjumlah keseluruhan 419 citra gambar dilakukan pembagian dengan rasio untuk data latih dan data uji dengan perbandingan masing-masing 80:20. Dengan rasio tersebut didapatkan 328 citra gambar untuk data katih dan 91 gambar untuk data uji. Selanjutnya dilakuka *pre-processing* berupa perubahan format data, pengurangan nilai rata-rata *pixel* pada setiap chanel gambar, serta melakukan resize

299×299 *pixel* pada *input size* untuk arsitektur *Xception*. Kemudian dilakukan pengujian dengan variasi *batch size* dan banyaknya *epoch* dan didapatkan akurasi terbaik 90%(Gaho dkk., 2024).

Penelitian yang berjudul “Klasifikasi Hama Ulat Pada Citra Daun Sawi Berbasis Convolutional Neural Network dengan Model *Xception*”. Akuisisi pada citra dalam penelitian tersebut dilakukan dengan dua cara yaitu dengan dilakukannya pengambilan data secara langsung menggunakan *smartphone* dan mengambil citra sawi dari *google images*. Sehingga jumlah *dataset* keseluruhan berjumlah 1000 citra gambar berformat PNG yang terdiri dari 2 kelas berupa sawi yang masih bagus (tidak terdapat hama) yang berjumlah 500 citra gambar dan 500 citra gambar daun sawi yang terdapat hama. Selanjutnya dilakukan proses *pre-processing* yang terdiri dari *cropping*, *resize* citra menjadi ukuran 128×128 *pixel*. Kemudian dilakukan augmentasi data yang bertujuan untuk menambah *dataset* pada data pelatihan dan terakhir menghilangkan *noise* pada citra gambar dengan Teknik *median filter*. Klasifikasi hama pada citra tanaman daun sawi memberikan hasil akurasi sebesar 96%, sensitifitas 96%, dan spesifisitas sebesar 97%(Rasyid & Wisudawati, 2024.).

Adapun penelitian yang berjudul “Klasifikasi Ras pada Kucing Menggunakan Algoritma Convolutional Neural Network(CNN)”. Pada penelitian tersebut dilakukan proses *pre-processing* sebelum tahap *training* atau *testing* model meliputi *resize*, augmentasi, dan normalisasi. *Resize* yang dilakukan pada tahap *pre-processing* merupakan pembuatan titik-titik baru yang disebut *interpolation* dimana umumnya *interpolation* terbagi menjadi dua yaitu *Non-adaptive* dan *Adaptive*. Teknik *bilinear interpolation* yang termasuk dalam *Non-Adaptive* digunakan pada tahapan *pre-processing*. Dataset yang digunakan dalam penelitian tersebut terdiri dari 12 kelas yang terbagi menjadi data *training* sebanyak 1068 citra gambar dan validasi sebanyak 528 citra gambar, serta *testing* sebanyak 797 gambar. Parameter yang digunakan yaitu *RMSprop* sebagai *optimizer* dengan *learning rate* sebesar 0.0001 dengan menerapkan citra input pada arsitektur *Xception* yaitu 299×299×3. Pada penelitian model *Xception + Transfer Learning + Fine Tuning* menghasilkan akurasi, *precision*, *recall*, dan *f1-score* dengan nilai masing-masing 93,75%, 93,74%, 93,56%, dan 93,64%(Fawwaz dkk., 2020).

Selanjutnya terdapat penelitian yang berjudul “Machine Learning Model For Meat Freshness Classification Using Xception Transfer Learning Architecture”. Penelitian tersebut dilakukan dengan penyiapan *dataset* yang di unduh daeri *Kaggle*. *Dataset* berisi 2269 citra gambar yang terbagi menjadi dua yaitu data pelatihan dan data pengujian dimana masing-masing terdiri dari tiga kelas antara lain *fresh*, *half-fresh*, dan *spoiled*. Ukuran untuk semua citra gambar yang terdapat dalam dataset yakni  $416 \times 416$  *pixel* yang memiliki 3 komponen warna (RGB). Selanjutnya dilakukan tahap *pre-processing* data dengan deklarasi *hyperparameter* untuk pengolahan data dengan *image width* dan *image height* masing-masing dengan nilai 150. Setelah itu dilakukan *resize* menjadi ukuran  $150 \times 150$  *pixel* dan *batch size* 100 yang bertujuan untuk mengurangi jumlah parameter *rescaling* dengan parameter  $1./255$  serta dilakukan augmentasi data yang bertujuan untuk mengurangi *overfitting*. Klasifikasi data untuk membangun arsitektur model *pre-trained* dengan *input shape* bernilai  $150 \times 150 \times 3$  dan menggunakan *optimizer Adam* dengan *learning rate* 0.001. Hasil evaluasi model di dapatkan nilai *loss*, *accuracy*, *precision*, *recall*, dan *f1-force* masing-masing secara berturut-turut adalah 29,30%; 86,92%; 87,25%; 86,47%; dan 87,59% (Cahyo dkk., 2023)

Adapun pada penelitian sebelumnya yang berjudul “Perbandingan Model Arsitektur CNN Dengan Metode Transfer Learning Untuk Klasifikasi Ikan Laut”. *Dataset* yang digunakan dalam penelitian tersebut bersumber dari *Large-Scale Dataset for fish Segmentation and Classification*. Sebanyak 9000 citra ikan laut terbagi menjadi 9 kelas dengan ukuran  $590 \times 445$  *pixel*. Kemudian pada tahap *pre-processing dataset* dibagi menjadi tiga bagian dengan presentase 70% data training, 10% data validasi, dan 20% *data testing*. *Resize* citran gambar dilakukan pada tahap *pre-processing* menjadi  $224 \times 224$  *pixel* serta dilakukan normalisasi dengan cara membagi nilai intensitas tiap *pixel* dengan 255 sehingga rentang nilai tiap *pixel* adalah -1 sampai 1. Augmentasi juga dilakukan untuk mengurangi *overfitting* dengan menerapkan pembalik acak *horizontal* dan *vertical*, serta *random rotation* dengan nilai 0,2. Hal tersebut bertujuan agar model dapat mempelajari citra dengan transformasi yang berbeda. Dengan begitu model dapat lebih mampu mengenali citra spesies ikan laut dengan berbagai variasinya. Pengujian dilakukan dengan

melatih data *training* kedalam *fit model* dengan menggunakan *epoch* sebanyak 50 kali, *batch size* 32 dengan menggunakan *Adam optimizer*. Model menunjukkan konsistensi dalam hal *precision*, *recall*, dan *F1-Score*, dengan nilai yang sama yakni 97%(Anugrah dkk., 2024)

Pada penelitian lain yang berjudul “Klasifikasi Penyakit Daun Jagung Menggunakan Loightweight Convolutional Neural Network” menggunakan arsitektur *Xception*. Dilakukan pengumpulan *dataset Corn or Mize Leaf Diseases Dataset* yang bersumber dari *Kaggle*. *Dataset* yang digunakan dalam penelitian tersebut berjumlah 4188 citra gambar jagung yang terdiri dari 4 kelas. Selanjutnya pada tahap perancangan model dilakukan *resize* citra gambar menjadi ukuran  $224 \times 224$  *pixel*. Kemudian dilakukan pembagian *dataset* dengan rasio antara data pengujian atau training dengan dataset validasi yakni 70:30. Setelah tahap *pre-processing* tahap pelatihan model dilakukan dengan menggunakan *optimizer Adam* dengan nilai *batch size* 10 dengan jumlah *epoch* 30 untuk arsitektur *Xception*. Dari rangkaian diatas kemudian dilakukan compile dengan loss categorical *crossentropy* yaitu *Cross Entropy* bertujuan untuk meminimalkan targer dan label yang diprediksi untuk setiap *pixel*. Model yang dibuat menggunakan bobot imagenet serta menambahkan *Dropout* ditutup dengan *Dense layer* dengan keluaran 4 kelas dengan *activation 'softmax'* Hasil dari *confusion* matrik dari model *Xception* dilakukan perhitungan *Precision*, *Recall*, *Accuracy*, dan *f1-score* terbaik masing-masing adalah 0.93, 0.90, 0.91(Hermawan, 2023).

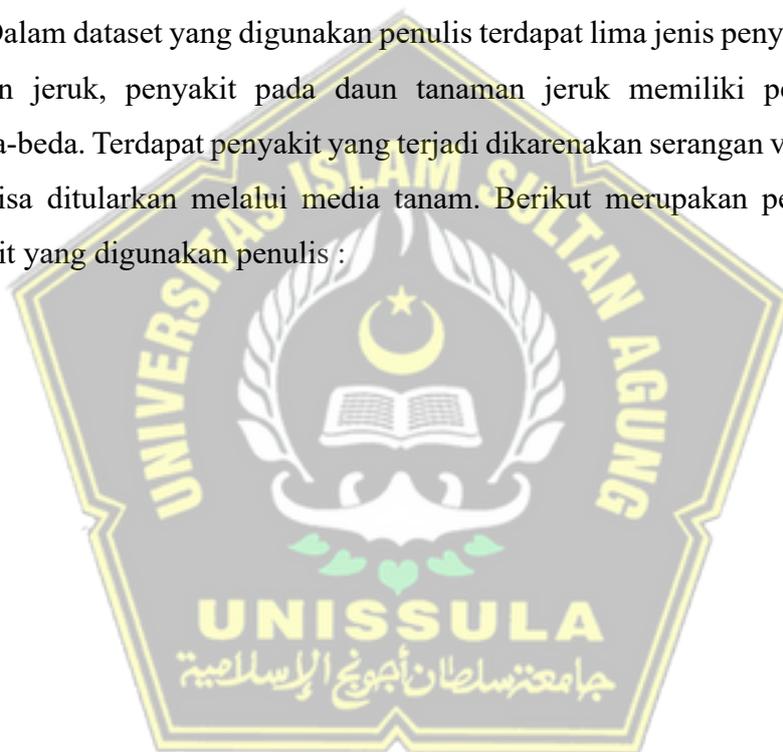
Terdapat penelitian sebelumnya yang berjudul “Klasifikasi Ketertarikan Belajar Anak PAUD Melalui Video Ekspresi Wajah dan Gestur Menggunakan Convolutional Neural Network” menggunakan arsitektur *Xception*. Pada tahap *pre-processing* deteksi dilakukan dengan memanfaatkan metode *Dlib* untuk mendapatkan landmark wajah sehingga didapatkan citra gambar potongan area wajah yang akan digunakan untuk melakukan pengenalan ekspresi seperti area mata, area mulut dan area wajah. Dari rangkaian tersebut didapatkan *dataset* 12878 citra gambar yang dibagi menjadi tiga kelas dengan rincian 5064 citra gambar ‘tertarik’, 4235 citra gambar pada kelas ‘cukup tertarik’, dan 3579 citra gambar pada kelas ‘tidak tertarik’. Kemudian data tersebut di distribukan menjadi data

training sebanyak 11590 citra gambar, data validasi 644 citra gambar, dan data testing 644 citra gambar. Kemudian digunakan hyperparameter berupa input size  $64 \times 64$  *pixel* dengan *range* warna (RGB). Kemudian dilakukan pengujian video yang berdurasi 60 detik. Dengan rangkaian tersebut mendapatkan hasil validasi akurasi 86%(Kusumastuti dkk., 2021)

## 2.2 Dasar Teori

### 2.2.1 Penyakit Daun Jeruk

Dalam dataset yang digunakan penulis terdapat lima jenis penyakit pada daun tanaman jeruk, penyakit pada daun tanaman jeruk memiliki penyebab yang berbeda-beda. Terdapat penyakit yang terjadi dikarenakan serangan virus dan jamur yang bisa ditularkan melalui media tanam. Berikut merupakan penjelasan lima penyakit yang digunakan penulis :



### 1. *Powdery Mildew*

*Powdery mildew* paling banyak ditemukan di bagian timur laut India selain Sikkim dan Darjeeling di hampir sebagian besar spesies jeruk. Awalnya, bagian udara tanaman dihindangi oleh jamur yang secara bertahap menyebar pada tanaman yang baru muncul di pembibitan dan kebun. Berdasarkan Berdasarkan tingkat keparahan penyakit, daun dan buah rontok sebelum waktunya. Ranting yang terinfeksi menunjukkan gejala mati gejala. Kematian tunas yang sedang tumbuh mengarah pada perkembangan beberapa tunas lateral dengan demikian menyebabkan penampilan yang lebat (Garofalo dkk., 2020).



Gambar 2. 1 Penyakit Powdery Mildew

Gambar 2.1 merupakan penyakit powdery meal dew yang menyenrang tanaman jeruk. Powdery mildew. Jeruk yang terkena penyakit powdery Mildew dapat dilihat gejalanya melalui daun. Dimana pada jeruk yang terkena penyakit ini akan mengalami ganguang serta dampak terburuknya akan mengalami kematian pada tanaman.

### 2. *Mealybug*

Wilayah Mediterania di Turki dengan produksi tahunan berbagai jenis hama yang menyerang kebun jeruk, *Planococcus citri* (Hemiptera: Pseudococcidae) dianggap sebagai hama utama pada jeruk kebun jeruk di wilayah Mediterania di dunia termasuk Turki. Kutu putih jeruk, *P. citri* adalah pertama kali ditemukan oleh Risso pada tahun 1813 dari sampel *Citrus spp.* di Prancis selatan. Kutu putih jeruk kutu putih adalah kutu putih yang tersebar luas, polifag dan salah satu salah satu serangga yang paling merusak (Ibrahim dkk., 2021). Kepadatan populasi *P. citri*

mencapai puncaknya puncaknya selama awal musim panas di Mediterania wilayah. *Nimfa P. citri* yang baru muncul menetap di berbagai bagian tanaman seperti bagian bawah bagian bawah daun, ranting, dan buah yang belum matang, dan mulai makan dengan menghisap cairan tanaman dan buah. Buah yang belum matang jatuh gugur lebih awal karena kerusakan akibat makan *P. citri* pada pertumbuhan buah dan tangkai buah yang sedang tumbuh.



Gambar 2. 2 Penyakit Mealybug

Gambar 2.2 merupakan citra daun jeruk yang terjangkit penyakit mealybug. Pada tanaman jeruk yang terkena penyakit mealybug umumnya dapat dilihat dari gejala yang terdapat pada daun. Penyakit mealybug disebabkan oleh parasite rayap yang memakan nutrisi pada daun jeruk. Hal ini dapat menyebabkan jeruk mengalami gangguan pertumbuhan bahkan kematian pada tanaman jeruk.

### 3. *Citrus Vein Phloem Degeneration(Huanglongbing)*

Gejala HLB bervariasi, dan dapat menyerupai gangguan lainnya. Namun, ada beberapa ciri khas yang unik. Pohon yang terinfeksi di lapangan biasanya akan mengembangkan satu atau lebih tunas berwarna kuning, sesuai dengan nama penyakitnya. Jika bagian lain dari pohon tetap sehat atau tidak menunjukkan gejala, maka penyakit ini akan terlihat seperti penyakit sektoral. Daun yang terserang mengembangkan pola area kuning dan hijau yang tidak memiliki batas yang jelas di antara warna-warna tersebut, memberikan penampilan "bercak bercak". Ini adalah gejala daun yang paling khas dan polanya tidak simetris pada dua bagian daun. Daun juga dapat menjadi lebih tebal, dengan urat-urat yang membesar dan tampak seperti gabus. Pada tahap selanjutnya, gejala defisiensi seng dapat berkembang, diikuti dengan daun rontok dan ranting mati (Home dkk., 20 C.E.)



Gambar 2.3 Penyakit Citrus Vein Phloem Degeneration

Gambar 2.3 merupakan citra gambar daun jeruk yang terjangkit penyakit huanglongbing. Gejala umum yang paling sering adalah terjadi perubahan warna daun menjadi kekuningan pada tanaman jeruk. Sama seperti gejala penyakit yang terdapat pada Gambar 2.1, huanglongbing juga bisa mengakibatkan gangguan pertumbuhan pada tanaman jeruk hingga kematian pada tanaman.

### 4. *Citrus Cancer*

*Citrus canker*, juga dikenal sebagai *Asiatic CC*, awalnya dilaporkan di Amerika Serikat pada awal tahun 1900-an setelah terjadi wabah di berbagai negara bagian tenggara. Pada tahun 1914, Hasse menerima sampel dari Florida, Texas, dan Mississippi, dan berhasil mengisolasi bakteri tersebut. Setelah menyelesaikan uji

karakterisasi dan patogenisitas, Hasse menamai bakteri tersebut dengan nama *Pseudomonas citri*. Sejak saat itu, bakteri ini diklasifikasikan ke dalam beberapa genus, termasuk *Bacterium*, *Phytomonas*, dan terakhir *Xanthomonas citri* pada tahun 1939. Genus *Xanthomonas* terdiri dari 27 fitopatogen yang menyebabkan penyakit kritis pada tanaman hias dan tanaman lainnya. Genus ini memiliki berbagai macam keluarga inang yang terdiri dari 68 famili, lebih dari 240 genera dan 140 patovar yang berbeda. Genus *Xanthomonas* dapat menginfeksi lebih dari 350 spesies, termasuk 268 dikotil dan 124 monokotil, termasuk biji-bijian, buah-buahan, kacang-kacangan, dan tanaman yang termasuk dalam famili *Brassicaceae* dan *Solanaceae* (Naqvi et al., 2022).



Gambar 2. 4 Citrus Cancer

Pada Gambar 2.4 merupakan citra gambar daun jeruk yang terkena penyakit cancer. Gejala utama yang dapat terlihat ialah kerusakan pada daun yang mengakibatkan gangguan pertumbuhan pada tanamannya.

##### 5. *Spiny Whitefly*

*Orange spiny whitefly*, *Aleurocanthus spiniferus* (Quaintance), dan *citrus blackfly*, *A. woglumi* Ashby (Hemiptera: Aleyrodidae), adalah dua hama jeruk yang paling penting dan berbahaya. Kedua spesies tersebut diatur sebagai hama karantina untuk Uni Eropa. *Aleurocanthus woglumi* belum menginvasi Eropa. Namun, pemodelan iklim memprediksi bahwa daerah di cekungan Mediterania cocok untuk berkembang biak. *Aleurocanthus spiniferus* telah menginvasi Eropa, dengan catatan pertama di Italia pada tahun 2008. Sejak saat itu, distribusinya di Uni Eropa telah meluas, termasuk Albania, Kroasia, Yunani, dan Montenegro. Kedua spesies ini menyebabkan gejala yang sama pada tanaman inang yang terserang. Di Hawaii,

selama periode serangan berat pada tahun 1998, pohon yang terserang tidak berbuah, dengan kisaran 50->600 pupa kutu kebul per daun yang terserang. Inang utama adalah spesies jeruk, tetapi mereka cukup polifag, dengan kisaran inang yang luas yang mencakup mangga dan kopi. Kerusakan langsung disebabkan oleh nimfa yang menghisap getah dari daun, menyebabkan kekurangan nutrisi dan air. Selain itu, nimfa mengeluarkan tetesan embun madu yang jatuh ke daun, di mana pertumbuhan jamur jelaga dipromosikan, yang akhirnya menutupi seluruh permukaan atas daun yang terserang. seluruh permukaan atas daun yang terserang. Baik *A. spiniferus* maupun *A. woglumi* berasal dari Asia Tenggara. Di daerah asalnya, mereka tidak dianggap sebagai hama utama; namun, selama abad ke-20, kedua spesies tersebut menginvasi bagian lain dunia, menyebabkan kerusakan substansial, terutama pada tanaman jeruk. Setelah pengenalan mereka secara tidak sengaja ke daerah baru, program pengendalian biologis dimulai untuk mengurangi kerusakan yang disebabkan oleh wabah kedua lalat putih tersebut



Gambar 2. 5 Penyakit Spiny Whitefly

Pada Gambar 2.5 merupakan gambar daun tanaman jeruk yang terinfeksi *spiny whitefly* atau kutu yang menjadi parasite pada daun. Daun yang terkena *whitefly* tidak akan bisa berbuah (Merthaningsih et al., 2020)

### 2.2.2 Machine Learning

*Machine learning* merupakan cabang dari kecerdasana buatan. Kecerdasan buatan merupakan salah satu bidang keilmuan yang mempelajari tentang komputer sains yang bertujuan untuk membuat komputer bisa memiliki kepintaran, kecerdasan untuk membantu manusia dalam menyelesaikan tugas-tugasnya. *Machine learning* secara khusus akan menggunakan metode statistika untuk

mempelajari data sehingga komputer dapat mempelajari pola pada data tanpa perlu diprogram secara eksplisit. Machine learning memiliki cabang yang menggunakan algoritma jaringan saraf tiruan untuk dapat belajar beradaptasi terhadap sejumlah data yang besar. Algoritma jaringan syaraf tiruan pada *deep learning* terinspirasi dari struktur otak manusia. Algoritma ini memungkinkan mesin untuk melihat pola dari data yang tidak terstruktur atau data yang fiturnya tidak dapat ditentukan secara langsung. Contohnya, data gambar, teks, audio, dan video. Belajar yang dimaksud tidaklah sama dengan proses manusia belajar, mesin mampu belajar apabila ia mampu meng-upgrade parameter, dimana parameter tersebut kurang-lebih merepresentasikan pengetahuan mesin *Machine Learning* membutuhkan Data yang valid sebagai bahan belajar (ketika proses training) sebelum digunakan testing untuk hasil output yang optimal. *Machine learning* dibagi menjadi 4 bagian berdasarkan metode pembelajarannya yaitu : (a)*supervised learning*, (b)*unsupervised learning*, (c)*semi-supervised learning* dan (d)*reinforcement learning*. Masing-masing metode ini memiliki cara yang berbeda dalam mengenali, mengolah data sehingga menghasilkan keluaran yang benar. Proses belajar machine learning berdasarkan data yang disebut dengan dataset, data set yang telah terkumpul akan dibagi menjadi dua, yaitu data training dan data testing. Data tersebut akan digunakan pada jenis machine learning yang sesuai. *Supervised learning* adalah bagian *machine learning* yang mempelajari sebuah *dataset* yang telah memiliki label, serta menghasilkan solusi yang diinginkan. Dataset yang telah memiliki label kemudian dipelajari oleh algoritma dan algoritma mempelajari pola dari pasangandata dan label tersebut. Terdapat banyak metode yang ada *regresi linier*, *random forest* dan *support vector machine*. *Unsupervised learning* bersifat deskriptif, yang akan berguna untuk mengelompokkan atau mengkategorikan data. Algoritma ini tidak mendapatkan *training dataset*, karena algoritma ini bukan bersifat prediktif, sehingga membutuhkan pembelajaran dari data yang telah ada. *Unsupervised* bekerja dengan mengelompokkan data berdasarkan kedekatannya disebut dengan *clustering*. Metode ini bekerja dengan menganalisis data yang tidak berlabel untuk menemukan pola tersembunyi dan menentukan korelasinya. Beberapa contoh algoritma yang dapat digunakan dalam *unsupervised learning*

seperti, *K-Means*, *Hierarchical clustering*, *DBSCAN*, dan *Fuzzy C-Means*. *Semi-Supervised Learning* merupakan gabungan dari *supervised learning* dan *unsupervised learning*. *Dataset* yang digunakan ada yang memiliki label ada juga yang tidak memiliki label. Untuk menyelesaikan jenis dataset seperti ini dapat menggunakan algoritma *semi supervised learning*. Algoritma ini berada di tengah-tengah antara algoritma *supervised* dan *unsupervised learning*. Cara kerja algoritma ini adalah dengan menggunakan teknik *unsupervised learning* untuk menemukan dan mempelajari struktur dalam variabel input. Setelah itu, sistem akan menggunakan algoritma *supervised learning* untuk membuat prediksi terbaik dari data tak berlabel dan memasukkan kembali data tersebut ke algoritma *supervised learning* sebagai data latih, kemudian menggunakan model tersebut untuk membuat prediksi baru dari data input baru. Salah satu contoh dari model *semi supervised learning* adalah *Deep Belief Network*(DBNs). *Reinforcement learning* dikenal sebagai model yang belajar menggunakan sistem reward dan penalti. Algoritma *reinforcement learning* didefinisikan sebagai metode *machine learning* yang berkaitan dengan cara software agent mengambil tindakan di environmentnya. *Reinforcement learning* adalah teknik yang mempelajari bagaimana membuat keputusan terbaik, secara berurutan, untuk memaksimalkan ukuran sukses kehidupan nyata (Pandia, 2024).

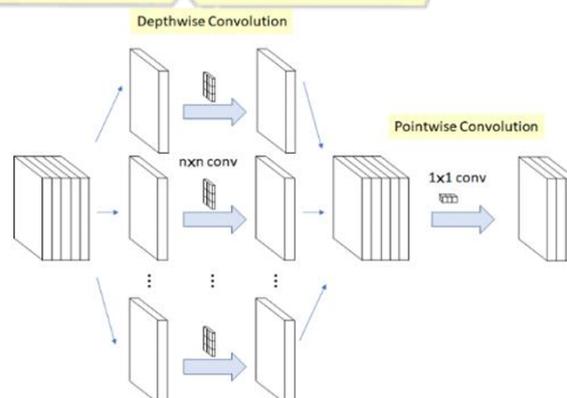
### 2.2.3 Deep Learning

*Deep Learning* merupakan salah satu bagian dari berbagai macam metode *machine learning* yang menggunakan *Artificial Neural Networks*(ANN). Metode ini dapat disebut sebagai tahap *advance* dari teknik *machine learning* karena memiliki *neural networks* yang lebih dalam dan memiliki sistem hitung yang lebih kompleks menggunakan layer non-linear berganda. Keuntungan *deep learning* dibandingkan dengan metode *machine learning* biasa adalah ekstraksi fitur yang lebih kompleks, lebih sedikit pemodelan dan memiliki prediksi yang lebih akurat walau dibayar dengan komputasi yang lebih tinggi. Beberapa model berbasis *deep learning* yang memiliki akurasi yang tinggi adalah ketika digunakan untuk deteksi wajah, pemrosesan gambar, sistem rekomendasi, *natural language processing*, dan

prediksi *time series*. Jenis pembelajaran dalam *deep learning* dapat berupa *supervised*, *semi supervised*, dan *unsupervised*. Secara sederhana *deep learning* melakukan metode pembelajaran dengan cara meniru bagaimana sistem dasar otak manusia bekerja yang disebut *neural networks*, itulah kenapa *deep learning* disebut menggunakan *artificial neural networks*. Dalam *deep learning* ada beberapa parameter yang harus di tuning untuk mendapatkan hasil yang paling baik atau optimal sesuai dengan data yang diproses, parameter ini disebut *hyperparameter*. Salah satu parameter yang mempengaruhi performa dari *model deep learning* adalah tipe dari fungsi aktivasi yang digunakan. Fungsi aktivasi tidak mungkin menghasilkan performa yang sama walaupun digunakan dalam satu arsitektur yang sama, akan sangat penting sekali untuk memilih fungsi aktivasi yang tepat untuk suatu model (Sanjaya & Budi, 2020).

#### 2.2.4 Arsitektur Xception

Arsitektur *Xception (Extreme Inception)* pertama kali dikenalkan dari pengembangan model *Inception* oleh F. Chollet. *Inception* merupakan model arsitektur *Convolutional Neural Network* dengan mempelajari filtrasi dalam ruang 3 dimensi; 2 dimensi spasial (lebar dan tinggi) dan saluran dimensi. Sebuah *convolutional kernel* bertugas untuk korelasi pemetaan lintas saluran dan korelasi spasial. Bertujuan agar proses menjadi lebih mudah dan efisien dengan membagikan secara *factorial* menjadi sekumpulan operasi yang secara independent mengawasi *cross-channel correlations* dan *spatial correlations* (Sirait dkk., 2022)

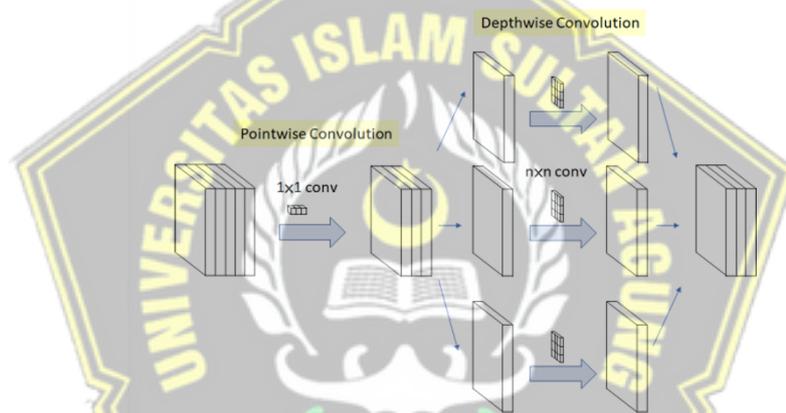


Gambar 2. 6 Original Depthwise Convolution

Gambar 2.6 merupakan Original Depthwise Convolution yang bersumber dari(Sirait et al., 2022). Merupakan konvolusi kedalaman yang diikuti oleh konvolusi pointwise.

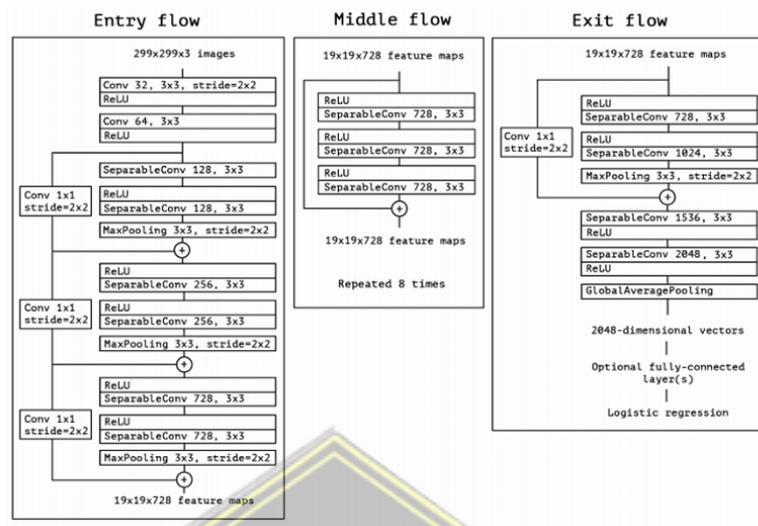
1. Depthwise convolution adalah konvolusi spasial  $n \times n$  saluran. Missal pada gambar diatas, kita memiliki 5 saluran, maka kita akan memiliki konvolusi spasial  $5(n \times n)$ .
2. Pointwise convolution adalah lilitan  $1 \times 1$  untuk mengubah dimensi.

Dibandingkan dengan konvolusi konvensional, kita tidak perlu melakukan konvolusi di semua saluran. Artinya jumlah koneksinya lebih sedikit dan modelnya lebih ringan.



Gambar 2. 7 Modified Depthwise Convolution

Pada gambar 2.7, merupakan Modified Depthwise Separable Convolution Xception(Sirait et al., 2022). Merupakan pointwise convolution yang di ikuti oleh depthwise convolution. Modifikasi ini dimotivasi oleh modul awal dalam Inception-v3 bahwa konvolusi spasial  $1 \times 1$  dilakukan terlebih dahulu sebelum konvolusi spasial  $n \times n$ . jadi, ini sedikit berbeda dari yang asli. ( $n = 3$  disini karena konvolusi spasial  $3 \times 3$  digunakan dalam Inception V3).



Gambar 2. 8 Arsitektur Xception

Gambar 2.8 merupakan arsitektur Xception yang terdiri dari 3 bagian utama yaitu Entry Flow, Middle Flow, dan Exit Flow. Pada Entry Flow pada arsitektur Xception terdiri dari konvolusi awal dan beberapa depthwise separable convolution untuk mengekstrak fitur dari gambar input. Kemudian terdapat 1 lapisan konvolusi standar di awal Entry Flow, di ikuti oleh 3 set depthwise separable convolution yang masing-masing memiliki 3 lapisan. Selanjutnya bagian kedua merupakan Middle Flow yang terdiri dari 36 lapisan dan pada tahap ini sebagian besar ekstraksi fitur terjadi. Middle Flow terdiri dari 8 modul identik dan masing-masing modul memiliki 3 lapisan depthwise separable convolution. Dengan demikian terdapat total 24 depthwise separable convolution. Terakhir terdapat Exit Flow dimana jaringan ini menyelesaikan ekstraksi fitur dan menghasilkan output akhir. Exit Flow terdiri dari 2 set depthwise convolution diikuti oleh lapisan konvolusi standar dan lapisan pooling.

### 2.2.5 Convolutional Neural Network

Metode *Convolutional Neural Network (CNN)* merupakan salah satu algoritma bagian dari *deep learning*. CNN termasuk kedalam jenis *deep neural network* karena kedalaman jaringannya yang tinggi dan banyak diaplikasikan pada data citra. CNN mampu melakukan perhitungan matematis pada input yang terdiri dari banyak lapisan tersembunyi. Lapisan tersembunyi mengelola input gambar dan

mengirimkan hasil yang diproses di bagian *output*. CNN memiliki lapisan konvolusi yang berguna untuk mengekstraksi fitur dan bagian klasifikasi untuk mengklasifikasi fitur ke dalam kelas yang ditentukan pada pelatihan. Fase pelatihan membutuhkan lebih banyak waktu untuk menyelesaikan dan membutuhkan perangkat komputasi yang mumpuni. Namun proses prediksi CNN cukup cepat dan akurat. Jaringan arsitektur *convolutional neural network* disajikan pada gambar 6.4 berikut

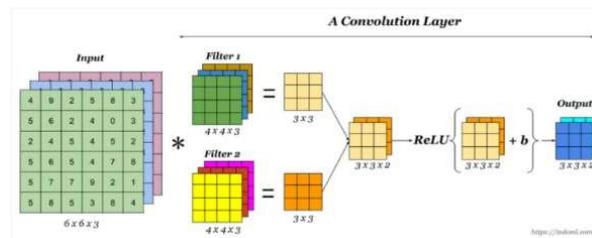


Gambar 2. 9 Arsitektur CNN

Gambar 2.9 Berdasarkan gambar 6.6, dapat dilihat CNN memiliki dua bagian utama yaitu feature learning dan bagian classification. Feature learning terdiri dari lapisan konvolusi, pooling, dan fungsi aktivasi. Bagian classification terdiri dari flatten, fully connected layer, dan fungsi aktivasi.

#### 1. Convolutional Layer

Proses konvolusi merupakan hasil dari perkalian *dot product* dari matriks input dan matrik filter. Matrik filter adalah matrik segi empat (*height-width*) yang berupa kumpulan nilai bobot (*weight*), sering juga disebut dengan matriks kernel. Ukuran *height* dan *weight* selalu lebih kecil dari input, tapi nilai *depth* (*channel*) sama dengan *depth input*. Proses konvolusi dijelaskan pada Gambar 6.7 dibawah ini (Bakti & Firdaus, 2023)



Gambar 2. 10 Convolutional layer

Pada gambar 2.10 merupakan proses yang terdapat pada *convolutional layer*. Perkanlian dari matrik input dengan matriks filter dilakukan dalam proses konvolusi tersebut.

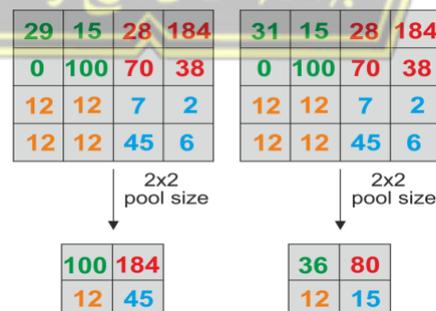
## 2. Activation Layer

Fungsi aktivasi merupakan suatu fungsi yang digunakan untuk menstranformasikan dengan perhitungan fungsi *non-linear*. Fungsi aktivasi yang digunakan pada CNN adalah *Rectified Linear Unit (ReLU)*. Fungsi ini digunakan untuk menghilangkan nilai aktivasi pada citra serta membantu mempercepat waktu komputasi(Bakti & Firdaus, 2023).

$$R(z) = \max(0, z)$$

## 3. Pooling Layer

Paad CN pooling layer mengguakan fitur map ataupun activation map. Keuntungan pooling layer adalah mengurangi dimensi volume output pada fitur map sehingga mengurangi overfitting. Pooling layer dijelaskan pada Gambar 6.8 dibawah ini(Bakti & Firdaus, 2023)



Gambar 2. 11 Pooling Layer

Pada Gambar 2.11 merupakan *pooling layer* yang digunakan dalam *convolutional neural network (CNN)*. Pada gambar tersebut terdapat dua pooling yang digunakan yaitu *average pooling* dan *max pooling*

## 4. Cross Channel Normalization

*Cross Channel Normalization* merupakan layer yang digunakan untuk mengatasi adanya nilai yang tidak terbatas yang diperoleh dari fungsi aktivasi *ReLU*. Selain itu, *cross channel normalization* digunakan menormalisasikan nilai tidak terbatas tersebut berdasarkan nilai elemen dari fitur terdekat baik berasal dari *feature map* yang sama ataupun tidak. Rumus yang digunakan pada operasi konvolusi ditunjukkan pada Persamaan dibawah(Bakti & Firdaus, 2023).

$$x_{ij}^p = \frac{a_{i,j}^p}{\left( \frac{\alpha \sum_{q=\max(1, p-\frac{n}{2})}^{\min(N-1, p+\frac{n}{2})} (a_{i,j}^q)}{n} \right)^\beta}$$

Gambar 2. 12 Cross Channel Normalization

Keterangan:

- $p$  : filter ke-p
- $x_{i,j}$  : piksel sesudah dinormalisasi
- $a_{i,j}$  : piksel sebelum dinormalisasi
- $k, \alpha, \beta, n$  : *hyperparameter*

#### 5. Dropout ij

Dropout merupakan suatu teknik dalam algoritma jaringan saraf yang digunakan untuk mengantisipasi adanya *overfitting* dengan menggabungkan beberapa jenis arsitektur. Jika semakin tinggi nilai bobot maka proses testing akan membutuhkan waktu yang semakin lama sehingga nantinya akan mempengaruhi kinerja CNN dalam melakukan proses *testing*. Ilustrasi *dropout* ditunjukkan oleh Gambar 2.12 dibawah(Bakti & Firdaus, 2023)

#### 6. Fully Connected Layer

*Fully connected layer* merupakan layer terakhir setelah *convolutional layer* dan *pooling layer* yang digunakan sebagai layer penyatuan akhir yang melakukan klasifikasi ke dalam kelas pada data *training*. Pada layer ini nilai-nilai piksel yang semula berupa matriks diubah kedalam bentuk 1 dimensi. Rumus yang digunakan pada operasi konvolusi ditunjukkan pada persamaan dibawah(Bakti & Firdaus, 2023).

Keterangan:

$y_i$  : output  
 $B_j$  : bias  
 $w_{i,j}$  : bobot  
 $x_i$  : input

## 7. Softmax layer

Softmax merupakan salah satu fungsi aktivasi yang digunakan untuk mengklasifikasikan suatu data dengan melihat nilai probabilitas tertinggi. Nilai softmax atau nilai probabilitas tiap data dengan interval 0 hingga 1. Rumus yang digunakan pada operasi konvolusi ditunjukkan pada persamaan dibawah(Bakti & Firdaus, 2023)

$$p_j(x) = \frac{e^{x_j}}{\sum_{i=1}^K p^{x_j}}$$

### 2.2.6 TensorFlow

Tensorflow Lite adalah evolusi dari Tensorflow Mobile, yang sudah mendukung penerapan perangkat seluler dan perangkat tertanam. Karena ada tren untuk menggabungkan machine learning dalam aplikasi seluler dan karena pengguna memiliki ekspektasi yang lebih tinggi pada aplikasi seluler dalam hal kamera dan suara, hal ini sangat untuk lebih mengoptimalkan tensorflow mobile untuk penggunaan mobile yang ringan . Beberapa optimalisasi yang termasuk dalam tensorflow lite adalah akselerasi perangkat keras melalui lapisan silikon, kerangka kerja seperti *Android Neural Network API* dan ANN yang dioptimalkan untuk seluler seperti *MobileNets* dan *SqueezeNet*. Model yang dilatih TF dikonversi ke format model tensorflow lite secara otomatis oleh tensorflow(Chicho & Sallow, 2021).

### 2.2.7 Keras

Keras adalah API jaringan saraf tingkat tinggi yang ditulis dalam bahasa Python yang dapat berjalan di atas TensorFlow dan kerangka kerja tingkat rendah lainnya. API ini dirancang dengan fokus untuk memungkinkan eksperimen cepat dan memungkinkan pembuatan prototipe yang sederhana dan cepat (melalui keramahan pengguna, modularitas, dan ekstensibilitas) (melalui keramahan pengguna, modularitas, dan ekstensibilitas). Keras menerima berbagai bentuk komponen jaringan syaraf, seperti lapisan tebal, lapisan konvolusi, lapisan

berulang, lapisan putus-putus, dan mendukung variasinya. Kode ini secara dinamis menangani alat-alat seperti *Central Processing Unit* (CPU) dan *Graphics Processing Unit* (GPU), sehingga memungkinkan penggunaan yang optimal. Kode ini juga memiliki implementasi fungsi aktivasi, pengoptimalisasi, rumus metrik, dan prosedur yang diperlukan untuk menangani sesi pelatihan dengan mudah. *Deep Learning plugin* mengintegrasikan fungsionalitas dari *library Keras*, yang pada gilirannya mengintegrasikan fungsi-fungsi dari TensorFlow ke dalam Python (Chicho & Sallow, 2021).

### **2.2.8 TensorFlow Lite**

Tensorflow Lite adalah evolusi dari Tensorflow Mobile, yang sudah mendukung penerapan perangkat seluler dan perangkat tertanam. Karena ada tren untuk menggabungkan machine learning dalam aplikasi seluler dan karena pengguna memiliki ekspektasi yang lebih tinggi pada aplikasi seluler dalam hal kamera dan suara, hal ini sangat penting untuk lebih mengoptimalkan tensorflow mobile untuk penggunaan mobile yang ringan. Beberapa optimalisasi yang termasuk dalam tensorflow lite adalah akselerasi perangkat keras melalui lapisan silikon, kerangka kerja seperti *Android Neural Network API* dan ANN yang dioptimalkan untuk seluler seperti *MobileNets* dan *SqueezeNet*. Model yang dilatih TF dikonversi ke format model tensorflow lite secara otomatis oleh tensorflow (Chicho & Sallow, 2021).



## BAB III METODE PENELITIAN

### 3.1 Metode Penelitian

Penelitian ini menggunakan metode Convolutional Neural Network dengan arsitektur Xception. Pada penelitian ini arsitektur akan dirancang menggunakan konfigurasi hyperparameter yang sekiranya di perlukan supaya model dapat mengenali dan bisa mengklasifikasi citra gambar dengan baik. Dibawah ini merupakan Langkah-langkah dalam pembuatan model :

#### 3.1.1 Studi *Literature*

Teori yang dipelajari dalam melakukan penelitian ini adalah mengenai bagaimana Convolutional Neural Network bisa mengklasifikasikan gambar dengan baik. Kemudian pemahaman tentang deep learning terutama bagaimana deep learning di implementasikan dalam pembuatan model. Serta penulis mempelajari bagaimana deployment mobile application dengan model machine learning yang sudah dilatih.

#### 3.1.2 Pengumpulan Data (*Data Collecting*)

*Data Collecting* merupakan proses pengumpulan dataset yang akan digunakan untuk proses klasifikasi citra gambar. Dataset yang digunakan pada penelitian ini memiliki sumber dari google dataset search dimana akuisisi datasetnya berasal dari Mendeley data Multi-format open-source sweet orange leaf dataset for disease detection, classification, and analysis (source : <https://data.mendeley.com/datasets/f7cr74mwpj/1>). Pada dataset yang sudah di download berisi 11 macam penyakit tanaman jeruk.

No	Nama Penyakit	Jumlah citra gambar
1.	Citrus Cancer	588
2.	Citrus Greening	431
3.	Citrus Mealybug	603
4.	Die Back	434
5.	Foliage Damaged	650
6.	Healty Leaf	594
7.	Powdery Mildew	598
8.	Shot Hole	560
9.	Spiny Whitefky	677

10	Yellow Dragon	407
11	Yellow Leaf	310

Tabel 3. 1 Dataset dan Kelas

Pada Tabel 3.1 merupakan banyak kelas yang terdapat dalam dataset yang akan digunakan penulis untuk melakukan klasifikasi gambar. Dari keseluruhan 11 kelas yang terdapat pada dataset, akan diambil 6 kelas sebagai sumber dataset yang akan dilakukan pengujian nantinya. enam (6 kelas) diantaranya yang akan digunakan adalah ‘Citrus Cancer’, ‘Citrus Greening’, ‘Citrus Healty’, ‘Citrus Mealybug’, ‘Powdery Mildew’, dan ‘Spiny Whitefly’. Pemilihan 6 kelas tersebut di karenakan perbedaan jumlah dataset.

### 3.1.3 Data Cleaning

Ketidaksamarataan dataset memberikan hasil yang buruk untuk prediksi model sehingga dilakukan penyamarataan jumlah citra gambar pada tiap kelas. Enam (6 Kelas) yang sudah dipilih pada pembahasan sebelumnya akan dilakukan pendistribusian jumlah citra gambar yang sama untuk setiap kelasnya. Pada tahap ini penulis melakukan pendistribusian jumlah citra gambar dengan rincian sebagai berikut :

No	Nama Kelas	Jumlah citra gambar
1.	cc	400
2.	cg	400
3.	ch	400
4.	cm	400
5.	pm	400
6.	sw	400

Tabel 3. 2 Distribusi dataset

Pada Tabel 3.2 penulis melakukan melakukan penyamarataan jumlah dataset pada setiap kelasnya menjadi 400 citra gambar. Hal ini bertujuan agar saat proses pengujian model arsitektur Xception meminimalkan terjadinya overfitting. Pada proses pengujian model overfitting sering terjadi dikarenakan banyak factor diantaranya model terlalu mempelajari data yang sudah ada sehingga sukar untuk

mempelajari data baru. Penyamaramaan ini tentunya hanya alternatif untuk mengurangi terjadinya overfitting.

#### **3.1.4 Data Preprocessing**

*Data preprocessing* adalah langkah penting dalam proses pengolahan data yang melibatkan persiapan dan pembersihan data mentah sebelum dianalisis atau diproses lebih lanjut. Tujuan utama dari data preprocessing adalah untuk memastikan bahwa data yang akan digunakan dalam analisis atau model pembelajaran mesin (machine learning) berada dalam format yang bersih, konsisten, dan siap untuk digunakan. Sebelum masuk proses *Training* atau Pelatihan Model. Dalam penelitian ini terdapat langkah *Data Preprocessing* yang harus dilalui diantaranya: Data preprocessing langkah esensial untuk mempersiapkan data mentah agar menjadi data yang berkualitas tinggi, siap untuk analisis atau pemodelan. Langkah ini mencakup pembersihan, transformasi, reduksi dimensi, dan pembagian data, yang semuanya bertujuan untuk meningkatkan keakuratan dan keandalan hasil analisis atau model yang dibangun.

##### **a. Split Data**

*Split data* adalah proses membagi dataset menjadi beberapa bagian terpisah untuk tujuan pelatihan dan pengujian model dalam pembelajaran mesin atau analisis statistik. Pembagian ini penting untuk memastikan bahwa model yang dibangun mampu melakukan generalisasi dengan baik terhadap data baru yang belum pernah dilihat sebelumnya. Tujuan utama dari split data adalah untuk menguji dan mengevaluasi performa model secara objektif. Dengan memisahkan data pelatihan (training data) dan data pengujian (testing data), kita dapat memastikan bahwa model tidak hanya belajar menghafal data, tetapi benar-benar memahami pola yang dapat diaplikasikan pada data baru.

b. Normalisasi

*Normalisasi data* adalah proses mentransformasikan data numerik ke dalam skala yang konsisten dan sebanding, biasanya dalam rentang tertentu, seperti antara 0 dan 1 atau -1 dan 1. Tujuan utama normalisasi adalah untuk memastikan bahwa semua fitur atau variabel memiliki bobot yang seimbang dalam model, terutama ketika fitur-fitur tersebut memiliki rentang nilai yang berbeda-beda. *Normalisasi data* merupakan langkah penting dalam preprocessing data yang memastikan fitur-fitur dalam dataset memiliki skala yang seimbang. Dengan menggunakan metode seperti Min-Max Scaling, Z-Score Normalization, atau lainnya, kita dapat meningkatkan kinerja dan akurasi model pembelajaran mesin serta mempercepat proses pelatihan.

c. *Image Augmentasi*

*Image augmentation* adalah teknik yang digunakan dalam pembelajaran mesin dan visi komputer untuk memperluas dataset gambar secara artifisial dengan membuat variasi gambar yang ada. Tujuan dari augmentasi gambar adalah untuk meningkatkan jumlah dan keragaman data pelatihan, yang dapat membantu model menjadi lebih umum (*generalized*) dan mengurangi *overfitting*, terutama ketika dataset asli kecil atau terbatas.

Pada penelitian ini variasi yang digunakan diantaranya adalah.

1. *Rotation Range*, Transformasi ini memutar gambar secara acak sehingga membantu model mempelajari fitur yang tidak bergantung pada orientasi objek dalam gambar.
2. *Zoom Range*, Transformasi ini secara acak memperbesar atau memperkecil gambar sehingga membantu model untuk mempelajari fitur yang relevan terlepas dari tingkat *zoom* gambar.
3. *Vertical Flip*, Transformasi ini membalik gambar secara vertikal secara acak (mencerminkan sepanjang sumbu atas-bawah). Ini membantu model mempelajari fitur yang tidak bergantung pada orientasi atas-bawah objek.
4. *Horizontal Flip*, Transformasi ini membalik gambar secara horizontal secara acak (mencerminkan sepanjang sumbu kiri-kanan). Ini membantu model mempelajari fitur yang tidak bergantung pada orientasi kiri-kanan objek.

5. *Brightness Range*, Transformasi ini secara acak menyesuaikan kecerahan gambar. Ini membantu model mempelajari fitur yang tidak bergantung pada variasi pencahayaan.

### 3.1.5 *Training*

Dataset yang telah dilakukan penyetaraan jumlah citra gambar kelas pada Tabel 3.2. kemudian dibagi menjadi 3 bagian antara lain data training, data testing, dan data validation. Rasion pembagian data training, testing, serta validation masing-masing secara berturut-turut sebanyak 80% : 10% : 10%.

No	Kelas	Train	Test	Validation
1.	cc	80%	10%	10%
2.	cg	80%	10%	10%
3.	ch	80%	10%	10%
4.	cm	80%	10%	10%
5.	pm	80%	10%	10%
6.	sw	80%	10%	10%
all	count	320	320	40

Tabel 3. 3 Rasio data

Keterangan :

- cc : Kelas Citrus Cancer
- cg : Kelas Citrus Greening
- ch : Kelas Citrus Healty
- cm : Kelas Citrus mealybug
- pm : Kelas Powdery mildew
- sw : Kelas Spiny Whitefly

Pembagian data dengan rasio seperti pada Tabel 3.3 dilakukan penulis untuk mendapatkan hasil prediksi yang bagus. Meskipun rasio data pada beberapa kasus bukan merupakan penentu hasil prediksi tapi pada tahap ini bisa mengurangi terjadinya overfitting.

### 3.1.6 Pengujian pada Arsitektur *Xception*

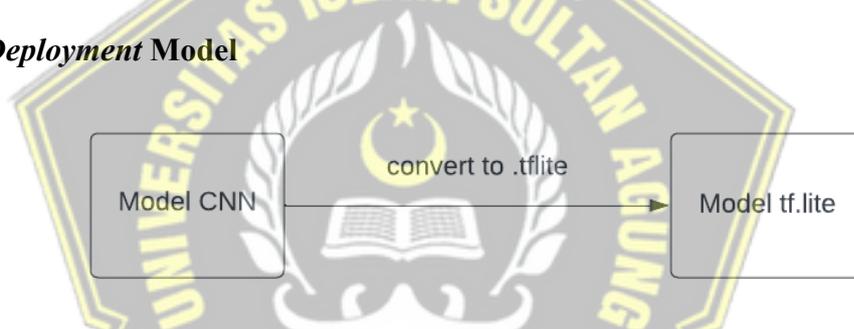
Untuk memperoleh hasil akurasi, *confusion matrix*, dan hasil *testing* yang baik pada model, Pada arsitektur *Xception* sendiri tidak adanya modifikasi pada

layernya karena merupakan hasil yang baik dari versi *Xception*. maka dari itu dibutuhkan beberapa pengujian yang harus dilakukan diantaranya. Pada *Data Preprocessing* dengan rasio 80 : 10 : 10 untuk data *train*, *test*, *valid*. Berikut merupakan detail pengujian pada model

Tabel 3. 4 Rangkaian Pengujian Arsitektur Xception

Model	<i>Preprocessing Data</i>	<i>Optimizer</i>
Xception	<ul style="list-style-type: none"> <li>- <i>Horizontal flip</i></li> <li>- <i>Vertical flip</i></li> <li>- <i>Brightness_range</i></li> <li>- <i>Zoom_range</i></li> <li>- <i>Rotation_range</i></li> </ul>	SGD

### 3.1.7 *Deployment Model*

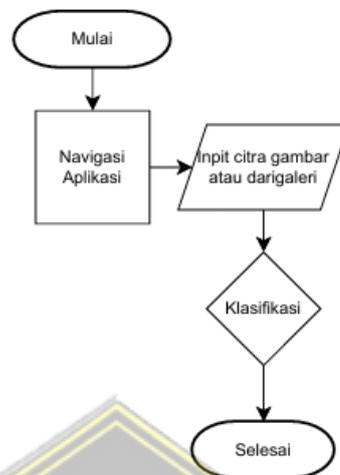


Gambar 3. 1 Alur *Deployment Model*

Setelah dilakukan serangkaian diatas kemudian data akan di compile kedalam model fit sehingga bisa diketahui nilai accuracy, loss, val loss, serta val accuracy. Hasil dari compile model kemudian akan di simpan kedalam format ‘.h5’ kemudian akan di convert kedalam format ‘.tflite’ menggunakan fungsi python yang nantinya akan digunakan untuk deployment aplikasi mobile.

### 3.1.8 *Pembuatan Aplikasi Mobile*

Setelah model sudah di-convert menjadi .tflite, beralih pada tahap pembuatan aplikasi mobile dengan menggunakan android studio.



Gambar 3. 2 flowchart aplikasi

Gambar 3.2 merupakan tahapan alur bagaimana proses klasifikasi terjadi dalam aplikasi android. Pada flowchart tersebut user bisa melakukan dua aksi dimana dapat melakukan pengambilan gambar dengan kamera ponsel serta mengambil gambar yang sudah ada pada galeri maupun internal ponsel. Kemudian akan muncul hasil akurasi dari gambar yang sudah dimasukkan model.

### 3.2 Analisis Kebutuhan

Pada tahap analisis kebutuhan ini, sistem yang sudah dirancang mulai dari hasil pelatihan model dengan arsitektur Xception sampai dengan proses klasifikasi yang terjadi dalam mobile application akan dilakukan modifikasi. Format akurasi kemudian akan ditampilkan sebagai informasi yang nantinya akan disajikan kepada pengguna aplikasi secara informatif dan mudah di pahami.

#### 3.2.1 Tips Penggunaan Aplikasi

Tips penggunaan aplikasi merupakan hal terpenting bagi user supaya tidak salah dalam *input* data.

1. Memiliki format PNG, JPG, JPEG (PNG diutamakan) karena pada model, gambar yang memiliki background putih atau warna lain kemungkinan bisa di

hitung sebagai value pada proses klasifikasi sehingga memengaruhi hasil prediksi.

2. Kualitas gambar citra yang bagus merupakan factor tingginya model akan bisa mengenali citra. Meskipun tidak pada semua kasus citra gambar harus jelas, namun ini bisa menjadikan model bekerja dengan baik dalam proses klasifikasi

### 3.2.2 Unggah Gambar

Pada fitur unggah gambar merupakan komponen sangat penting pada sistem ini, dimana user akan dihadapkan dalam dua pilihan yaitu dengan dari kamera atau dari galeri yang telah menyimpan beberapa gambar untuk diuji coba.

### 3.2.3 *Resize* Gambar

*Resize* gambar merupakan fitur yang esensial untuk mempersiapkan citra agar siap untuk diprediksi. Jika gambar tidak diubah ukurannya menjadi 299x299 pixel sebelum diprediksi oleh model, aplikasi dapat mengalami crash. Hal ini disebabkan karena *ByteBuffer* yang dihasilkan dari gambar yang tidak di-*resize* dapat melebihi kapasitas *ByteBuffer* yang dapat ditampung oleh model .tflite. Oleh karena itu, perlu untuk memastikan bahwa gambar yang digunakan telah di-*resize* agar sesuai dengan kebutuhan model sebelum melakukan prediksi.

### 3.2.4 Melakukan Prediksi

Fitur ini bertugas untuk melakukan klasifikasi pada citra gambar yang di masukan user. Proses klasifikasi dilakukan secara komputasi berdasarkan dari hasil model yang telah dibuat sebelumnya. Model dengan format '.tflite' yang telah ditambahkan kedalam pembuatan aplikasi akan melakukan prediksi kemudian hasil prediksinya berupa akurasi dari tiap input gambar.

### 3.2.5 Menampilkan Hasil Prediksi

Gambar yang telah di input oleh user baik menggunakan fitur kamera maupun dari internal ponsel kemudian akan dilakukan proses komputasi dan akan menyajikan nilai akurasi dalam pesentase nilai dalam bentuk text-view.

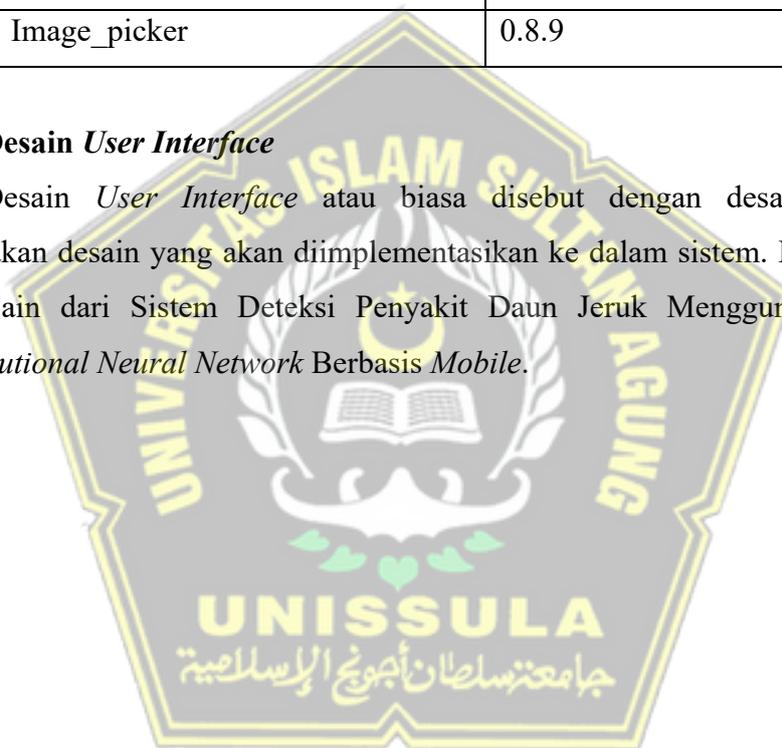
## 3.3 Analisis Sistem

Pada tahap analisis sistem, akan dianalisis *tools* apa saja yang digunakan dalam mengembangkan sistem, berikut merupakan tabel *tools* yang akan digunakan pada sistem:

<i>Tools</i>	<i>Version</i>
Laptop Acer Swift 3	Windows 11
Redmi 9T	Android 11
Google Colaboratory	T4 GPU
Android Studio Koala	1.87.2
Kotlin	3.16.7
Pixel 3 XL (Emulator)	API 30
kotlin_tflite	1.0.1
Image_picker	0.8.9

### 3.4 Desain *User Interface*

Desain *User Interface* atau biasa disebut dengan desain antarmuka merupakan desain yang akan diimplementasikan ke dalam sistem. Berikut adalah UI desain dari Sistem Deteksi Penyakit Daun Jeruk Menggunakan Metode *Convolutional Neural Network* Berbasis *Mobile*.



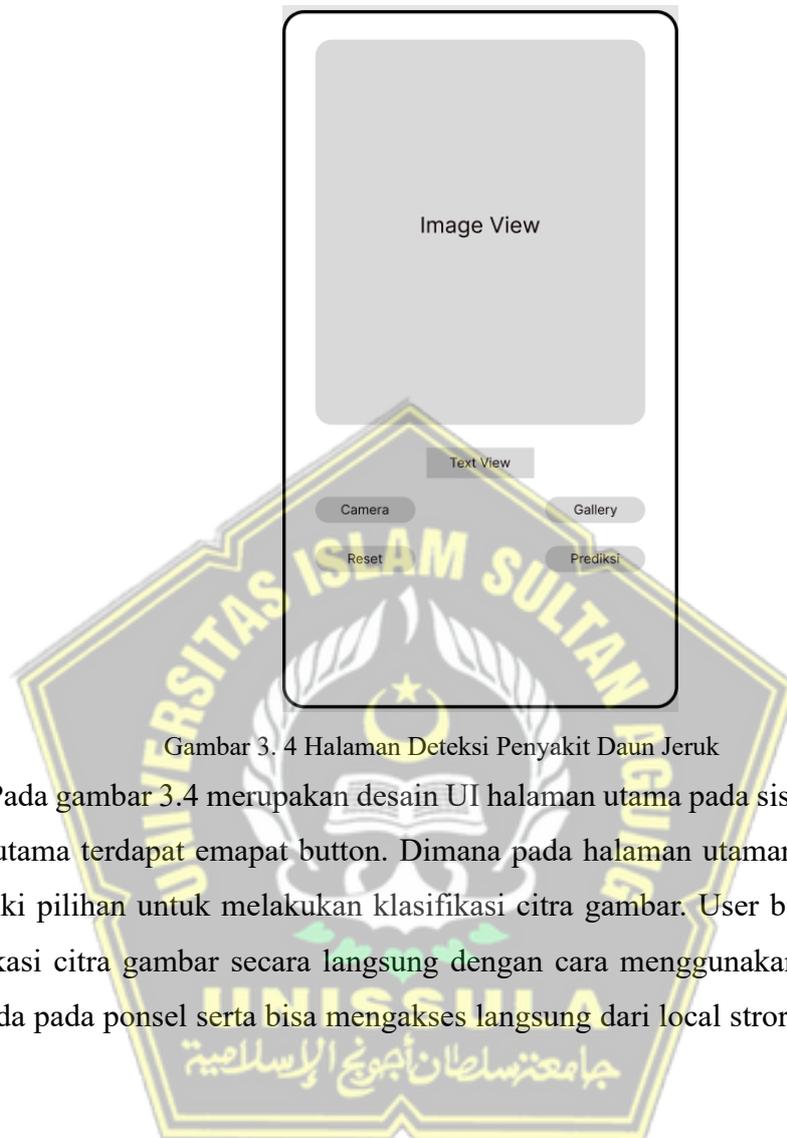
### 3.4.1 Halaman *Splash Screen*



Gambar 3. 3 Tampilan Ui Splash Screen

Gambar 3.3 merupakan tampilan splash screen yang ada pada aplikasi mobile. Tampilan tersebut merupakan tampilan awal saat aplikasi mobile dibuka oleh user. Pembuatan logo dilakukan sendiri, serta wireframe adalah tampilan yang di buat dengan figma.

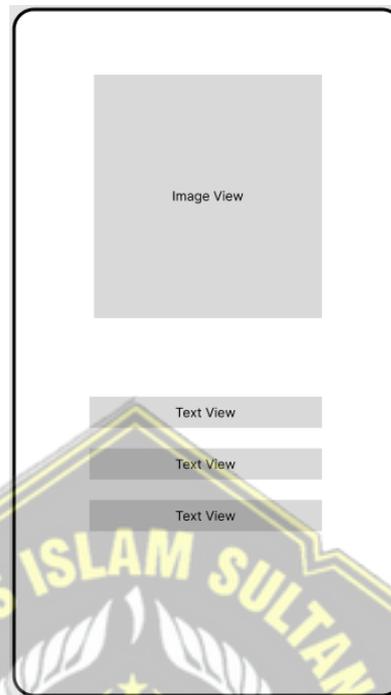
### 3.4.2 Halaman Klasifikasi Penyakit Daun Jeruk



Gambar 3. 4 Halaman Deteksi Penyakit Daun Jeruk

Pada gambar 3.4 merupakan desain UI halaman utama pada sistem ini, dalam menu utama terdapat empat button. Dimana pada halaman utaman tersebut user memiliki pilihan untuk melakukan klasifikasi citra gambar. User bisa melakukan klasifikasi citra gambar secara langsung dengan cara menggunakan fitur kamera yang ada pada ponsel serta bisa mengakses langsung dari local storage dari galeri ponsel.

### 3.4.3 Halaman Profil



Gambar 3. 5 Halamn Profil

Pada gambar 3.5 merupakan desain UI pada bagian profil. Halaman profil ini memuat informasi penulis yang berisikan informasi singkat tentang penulis.

## BAB IV

### HASIL DAN ANALISIS PENELITIAN

#### 4.1 Hasil dan Analisis Model CNN

Berdasarkan tabel 3.4 yang telah dilakukan pengujian pada arsitektur Xception: pemberian hyperparameter yang telah dilakukan pada tahap pre-processing bertujuan agar supaya citra gambar dapat di kenali oleh model. Sehingga model bisa mengklasifikasikan gambar dengan baik

##### 4.1.1 Pengujian Model Xception dengan Hyperparamater

Pengujian model Xception dengan hyperparameter pada tahap pre-processing seperti rangkaian yang dilakukan pada Tabel 3.4 pada bab sebelumnya. Penggunaan hyperparameter seperti rescale '1/255' dimana hal tersebut dilakukan bertujuan untuk membagi nilai dari 3 channel RGB sehingga di dapatkan nilai baru dengan range antara 0 sampai 1. Kemudian juga terdapat hyperparameter lainnya seperti flipping (Horizontal, Vertical) yang di terapkan sebagai berikut :

NO	Paramater	Value
1.	Rescale	1./255
2.	Zoom range	0.2
3.	Vertical flip	True
4.	Horizontal flip	True
5.	Brightness range	[0.8, 1.2]

Tabel 4. 1 Maccam Paramater

Pada gambar 4.1 merupakan pemberian parameter sebelum dilakukannya proses pelatihan dengan arsitektur Xception. Hyperparameter mengontrol perilaku pelatihan dan struktur model, seperti tingkat pembelajaran (learning rate), ukuran batch (batch size), jumlah lapisan (layers), dan sebagainya. Mereka berbeda dari parameter model, yang dipelajari secara langsung oleh model selama proses pelatihan.

#### 4.1.2 Pengujian dengan Optimizer SGD

**SGD** (Stochastic Gradient Descent) adalah salah satu algoritma optimasi paling dasar dan populer yang digunakan untuk melatih model pembelajaran mesin, terutama jaringan saraf (neural networks). SGD adalah varian dari algoritma Gradient Descent yang memperbarui parameter model berdasarkan satu contoh data atau mini-batch dari dataset, bukan seluruh dataset.

Gradient Descent adalah algoritma optimasi yang digunakan untuk meminimalkan fungsi kerugian (loss function). Tujuannya adalah menemukan nilai parameter model (misalnya, bobot dan bias) yang meminimalkan kesalahan antara prediksi model dan nilai target yang sebenarnya.

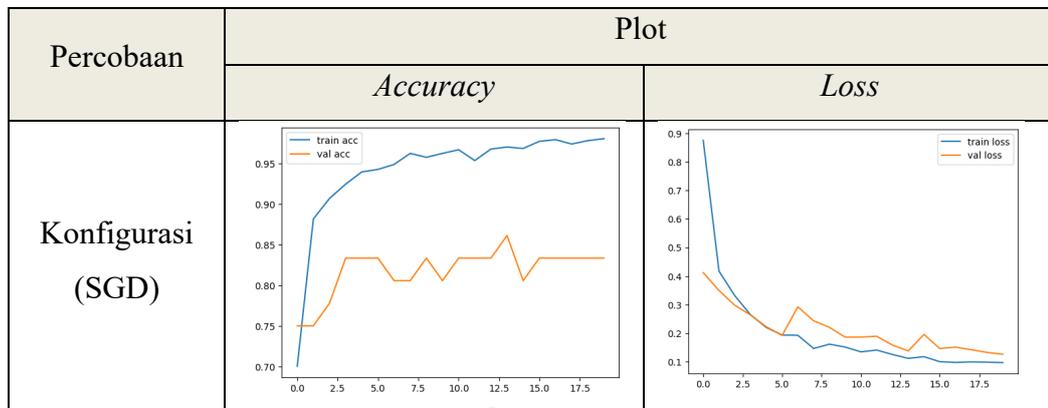
Proses ini dilakukan dengan menghitung gradien dari fungsi kerugian terhadap parameter model, kemudian memperbarui parameter tersebut ke arah yang berlawanan dengan gradien. Gradien menunjukkan arah peningkatan terbesar dalam fungsi kerugian, sehingga dengan bergerak berlawanan arah, kita dapat meminimalkan fungsi kerugian.

Penggunaan Optimizer yang berbeda bisa memberikan nilai akurasi yang berbeda juga. Penulis menggunakan optimizer SGD dengan learning rate '0.0001' yang nantinya akan digunakan saat meng-compile model dengan arsitektur Xception

#### 4.1.3 Hasil Akurasi

Setelah meng-compile model dengan arsitektur Xception serta penggunaan optimizer SGD, didapatkan hasil accuracy serta loss yang di visualisasikan sebagai berikut :

Tabel 4. 2 Konfigurasi Optimizer SGD



Tabel 4.2 merupakan grafik dari plot *accuracy* dan *loss* pada masing-masing konfigurasi. Semakin naik grafik pada plot *accuracy* dan semakin turun grafik pada plot *loss*, serta grafik *train* dan *val* yang rapat dan saling mendekati, model akan semakin baik dalam mengklasifikasi gambar.

Tabel 4. 3 Hasil Compile Optimizer Adam

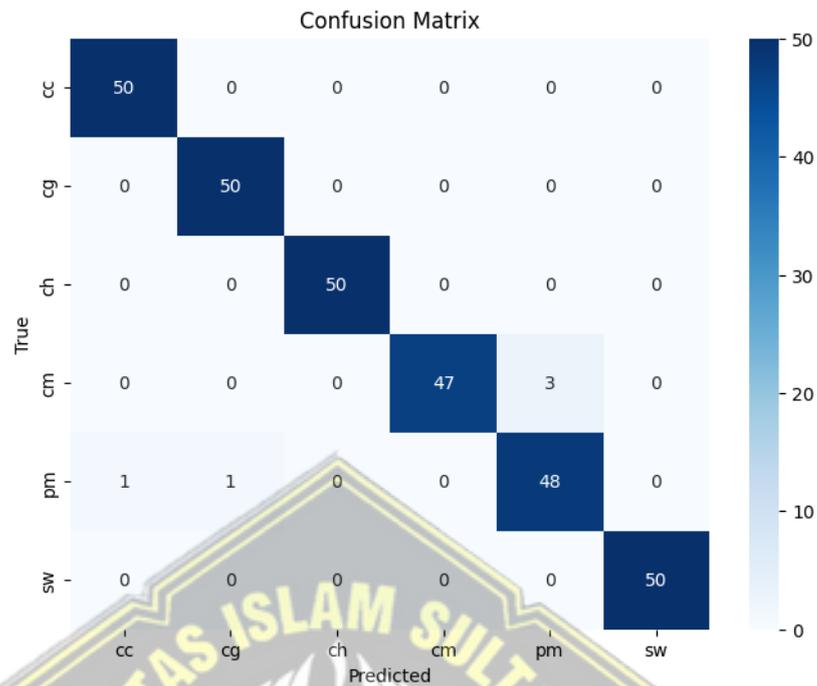
Percobaan	<i>epochs</i>	<i>acc</i>	<i>val-acc</i>	<i>loss</i>	<i>val- loss</i>
<b>Konfigurasi 1 (SGD)</b>	<b>20/20</b>	<b>0.9814</b>	<b>0.8333</b>	<b>0.0934</b>	<b>0.1260</b>

Dari tabel 4.2 menunjukkan bahwa pengujian menggunakan optimizer SGD dengan hasil akurasi yang tinggi yaitu 98,14% untuk *train accuracy* dan 83,33% untuk *test accuracy*. Pada model dapat disimpulkan bahwa indikasi overfitting tidak terjadi. Kesimpulan ini dapat diambil dari nilai *accuracy* dan *val accuracy* dimana hasil untuk *accuracy* lebih besar hamper 10% dari *val accuracy* yang berarti model mampu bekerja dengan baik pada data baru. Kemudian untuk hasil *loss* dengan *val loss* menunjukan bahwa nilai *loss* lebih rendah 3,5% dibandingkan dengan *val loss*.

#### 4.1.4 Hasil *Testing Model CNN*

Setelah melewati tahap uji akurasi dan mendapatkan hasilnya selanjutnya dilakukannya *Testing* pada model CNN. Tahap *Testing*, dilakukan untuk mengevaluasi keakuratan sistem dalam mengklasifikasikan data *testing* yang terdiri dari 6 kelas penyakit daun jeruk, maka diperlukan perhitungan *confusion matrix* digunakan untuk mencari nilai *accuracy*, *precision*, *recall*, dan *f1 score* yang mana dilakukan ke semua percobaan atau konfigurasi.

##### a. Konfigurasi (SGD)



Gambar 4. 1 Confusion Matriks Optimizer SGD

Konfigurasi	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
Konfigurasi (SGD)	0.98	0.98	0.98	0.98

Pada tabel 4.1 merupakan hasil *accuracy*, *precision*, *recall*, dan *f1-score*, jumlah data pada masing-masing kelas yang dilakukan *testing* menggunakan konfigurasi (SGD).

Tabel 4. 4 *Precision*, *Recall*, & *F1-score* Kelas ( Konfigurasi 1)

K	Jumlah Sampel Data	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
cc	40	0.98	1.00	0.99
cg	40	0.98	1.00	0.99
ch	40	1.00	1.00	1.00
cm	40	1.00	0.94	0.97
pm	40	0.94	0.96	0.95
sw	40	1.00	1.00	1.00

Sementara jika dihitung secara manual pada perhitungan *accuracy*, *precision*, *recall*, dan *f1-score* berdasarkan *confusion matrix* pada gambar 4.1, hasilnya hampir sama dengan perhitungan otomatis. Untuk hasil secara manual dapat dilihat pada tabel 4.5.

Tabel 4. 5 TP,FP, FN, dan TN

Kelas	TP	FP	FN	TN
cc	50	1	0	249
cg	50	1	0	249
ch	50	0	0	250
cm	47	0	3	250
pm	48	3	2	247
sw	50	0	0	250

Keterangan: cc = Citrus Cancer

cg = Citrus Greening

ch = Citrus Healthy

cm = Citrus Mealybug

pm = Powdery Mildew

sw = Spiny Whitefly

Pada tabel 4.6 merupakan perhitungan manual *precision*, recall, dan *f1-score* per kelas berdasarkan nilai TP, TN, FP, FN.

Tabel 4. 6 Perhitungan Manual

	Perhitungan	Hasil
	$Precision (P) = \frac{TP}{TP + FP}$	
$P_{cc}$	$= \frac{TP_{cc}}{TP_{cc} + FP_{cc}} = \frac{50}{50 + 1} = \frac{50}{51} = 0.98$	0.98
$P_{cg}$	$= \frac{TP_{cg}}{TP_{cg} + FP_{cg}} = \frac{50}{50 + 1} = \frac{50}{51} = 0.98$	0.98
$P_{ch}$	$= \frac{TP_{ch}}{TP_{ch} + FP_{ch}} = \frac{50}{50 + 0} = \frac{50}{50} = 1.00$	1.00
$P_{cm}$	$= \frac{TP_{cm}}{TP_{cm} + FP_{cm}} = \frac{47}{47 + 0} = \frac{47}{47} = 1.00$	1.00
$P_{pm}$	$= \frac{TP_{pm}}{TP_{pm} + FP_{pm}} = \frac{48}{48 + 3} = \frac{48}{51} = 0.94$	0.94
$P_{sw}$	$= \frac{TP_{sw}}{TP_{sw} + FP_{sw}} = \frac{50}{50 + 0} = \frac{50}{50} = 1.00$	1.00

$\mathbf{Recall (R)} = \frac{TP}{TP + FN}$		
$R_{cc}$	$= \frac{TP_{cc}}{TP_{cc} + FN_{cc}} = \frac{50}{50 + 0} = \frac{50}{50} = 1.00$	1.00
$R_{cg}$	$= \frac{TP_{cg}}{TP_{cg} + FN_{cg}} = \frac{50}{50 + 0} = \frac{50}{50} = 1.00$	1.00
$R_{ch}$	$= \frac{TP_{ch}}{TP_{ch} + FN_{ch}} = \frac{50}{50 + 0} = \frac{50}{50} = 1.00$	1.00
$R_{cm}$	$= \frac{TP_{cm}}{TP_{cm} + FN_{cm}} = \frac{47}{47 + 3} = \frac{47}{50} = 0.94$	0.94
$R_{pm}$	$= \frac{TP_{pm}}{TP_{pm} + FN_{pm}} = \frac{48}{48 + 2} = \frac{48}{50} = 0.96$	0.96
$R_{sw}$	$= \frac{TP_{sw}}{TP_{sw} + FN_{sw}} = \frac{50}{50 + 0} = \frac{50}{50} = 1.00$	1.00
$\mathbf{F1 - Score} = \frac{2 * Precision * Recall}{Precision + Recall}$		
$F1_{cc}$	$\frac{2 * P_{cc} * R_{cc}}{P_{cc} + R_{cc}} = \frac{2 * 0.98 * 1.00}{0.98 + 1.00} = \frac{1.96}{1.98} = 0.89$	0.99
$F1_{cg}$	$\frac{2 * P_{cg} * R_{cg}}{P_{cg} + R_{cg}} = \frac{2 * 0.98 * 1.00}{0.98 + 1.00} = \frac{1.96}{1.98} = 0.89$	0.99
$F1_{ch}$	$\frac{2 * P_{ch} * R_{ch}}{P_{ch} + R_{ch}} = \frac{2 * 1.00 * 1.00}{1.00 + 1.00} = \frac{2.00}{2.00} = 1.00$	1.00
$F1_{cm}$	$\frac{2 * P_{cm} * R_{cm}}{P_{cm} + R_{cm}} = \frac{2 * 1.00 * 0.94}{1.00 + 0.94} = \frac{1.88}{1.94} = 0.96$	0.97
$F1_{pm}$	$\frac{2 * P_{pm} * R_{pm}}{P_{pm} + R_{pm}} = \frac{2 * 0.94 * 0.96}{0.94 + 0.96} = \frac{1.9012}{1.95} = 0.949$	0.95
$F1_{sw}$	$\frac{2 * P_{sw} * R_{sw}}{P_{sw} + R_{sw}} = \frac{2 * 1.00 * 1.00}{1.00 + 1.00} = \frac{2.00}{2.00} = 1.00$	1.00

Pada tabel 4.6 ini merupakan perhitungan *accuracy*, *precision*, *recall*, dan *f1-score* pada confusion matriks dengan arsitektur Xception dan optimizer SGD. Dalam perhitungan tersebut dengan hasil komputasi memiliki nilai yang sama.

## **4.2 Cara Kerja Sistem Klasifikasi atau Deteksi**

### **4.2.1. Unggah Gambar**

Pada menu utama aplikasi mobile user dapat melakukan dua aksi. Pertama user dapat memasukan gambar langsung menggunakan fitur kamera yang tersedia dalam ponsel. Kemudian hasil dari citra yang diambil menggunakan kamera akan tersimpan di galeri untuk kemudian akan diklasifikasi. User juga bisa langsung mengambil gambar yang sudah tersimpan pada internal ponsel melalui galeri untuk nantinya akan dilakukan klasifikasi.

### **4.2.2. Konversi Gambar ke TensorImage**

Pada tahap ini, sistem akan mengambil gambar yang diunggah oleh pengguna dan menyimpannya sebagai sebuah gambar bitmap. Kemudian, gambar akan diubah menjadi format TensorImage. Proses konversi ini diperlukan agar gambar dapat disesuaikan dengan ukuran yang diharapkan oleh model, yaitu *resize* gambar menjadi 299 x 299 piksel. Ukuran akan disesuaikan dengan input layer pada arsitektur Xception sehingga bisa diproses oleh sistem.

### **4.2.3. Konversi TensorImage ke TensorBuffer**

Pengolahan gambar menggunakan Tensorflow perlu dilakukan konversi tensor image menjadi tensor buffer. Dalam proses machine learning diperlukan data gambar dalam format yang dapat digunakan oleh model. Dalam penelitian ini model menggunakan arsitektur Xception memerlukan input gambar berukuran 299 × 299 pixel. Proses konversi Tensor Image ke tensor Buffer akan dilakukan pengubahan input gambar yang telah diambil pada kamera kedalam Tensor Buffer yang memiliki dimensi '[height, width, channel]' untuk gambar tunggal.

### **4.2.4. Proses Klasifikasi**

Pada proses klasifikasi gambar yang mengharuskan konversi gambar menjadi TensorBuffer merupakan bagian yang penting dalam pipeline gambar.

Pembelajaran mesin yang di optimalkan untuk perangkat seluler atau edge menggunakan TensorFlow Lite mengharuskan untuk merubah dan memproses gambar menjadi tensor yang sesuai dengan input model Xception yaitu  $299 \times 299$  pixel. Kemudian melakukan konversi gambar kedalam TensorBuffer yang kompatibel dengan TensorFlow Lite. Setelah gambar diubah kedalam tensorbuffer kemudian melakukan inferensi menggunakan model yang di muat pada interpreter pada TensorFlow Lite. Langkah terakhir melakukan interpretasi hasil klasifikasi untuk bias memperoleh prediksi sesuai dengan kelas yang ada dalam model Xception.

#### **4.2.5. Hasil Klasifikasi**

Pada serangkaian proses diatas kemudian akan ditampilkan hasil klasifikasi dari model Xception. Hasil dari klasifikasi merupakan informasi jenis kelas penyakit pada daun tanaman jeruk serta persentase dari hasil prediksinya. Informasi tersebut akan ditampilkan pada TextView yang digunakan pada aplikasi model sehingga user bisa mengetahui hasil dari klasifikasinya.

### **4.3 Implementasi *User Interface* (UI)**

#### **4.3.1 Halaman *SplashScreen***

Pada halaman *SplashScreen* ini merupakan tampilan awal aplikasi, dimana terdapat logo daun.

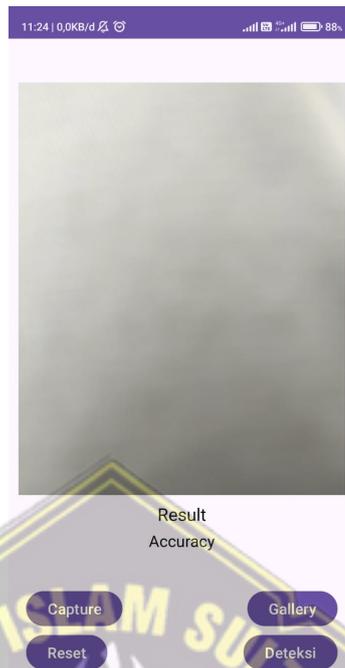


Gambar 4. 2 Halaman *SplashScreen*

Pada gambar 4.10 merupakan tampilan awal user ketika membuka aplikasi. Dibutuhkan waktu 3 detik untuk otomatis menuju halaman selanjutnya.

#### **4.3.2 Halaman *Home***

Pada halaman terdapat beberapa navigasi yang akan digunakan sebagai klasifikasi. Tampilan aplikasi mobile dibuat sedemikian rupa agar bisa mudah digunakan oleh user.



Gambar 4. 3 Halaman Utama

Pada gambar 4.7 aplikasi terdiri dari ImageView, TextView, serta Button. Image View dalam halaman utama memiliki fungsi untuk menampilkan gambar yang nantinya akan di input oleh user. Kemudian terdapat empat tombol yang memiliki fungsi untuk mengambil gambar dari fitur kamera pada ponsel. Terdapat pula tombol untuk mengunggah mengambil gambar dari galeri ponsel. Citra gambar yang sudah di input oleh user akan di lakukan klasifikasi oleh aplikasi sehingga memberikan output yang kemudian di tampilkan pada TextView.

#### 4.4 Pengujian Sistem

Pada tahap ini, sistem akan mengalami pengujian menggunakan metode *blackbox testing*. *Blackbox testing* adalah metode pengujian perangkat lunak yang bertujuan untuk menguji fungsionalitas sistem tanpa memerlukan pengetahuan tentang struktur kode sistem. Terdapat beberapa jenis *blackbox testing*, antara lain *functional testing*, *non-functional testing*, dan *regression testing*. Dalam penelitian ini, penulis menggunakan jenis *functional testing*, di mana pengujian dilakukan untuk menguji setiap fungsi yang ada dalam sistem. Tabel 4.1 menunjukkan hasil dari pengujian fungsional pada sistem ini.

Tabel 4. 7 Skenario aplikasi

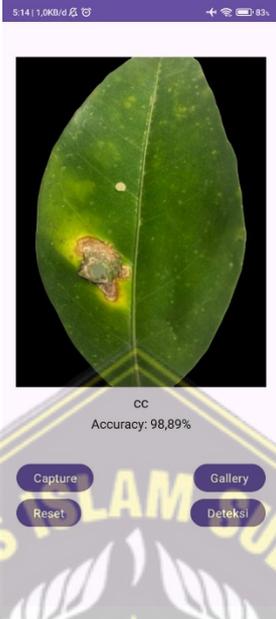
Skenario Pengujian	Kasus Pengujian	Hasil Pengujian	Kesimpulan
Mengambil foto dari kamera	Mengambil foto daun jeruk dari kamera	Sesuai	Nomal
Memilih foto dari galeri	Memilih gambar dari galeri	Sesuai	Normal
Upload Gambar	Melakukan <i>upload</i> gambar	Sesuai	Normal
Preview Gambar dari galeri atau kamera	Foto berhasil ditampilkan	Sesuai	Normal
<i>Testing</i> gambar penyakit daun Jeruk Citrus Cancer	Melakukan <i>testing</i> gambar penyakit daun Jeruk Citrus Cancer	Sesuai	Normal
<i>Testing</i> gambar penyakit daun Citrus Healty	Melakukan <i>testing</i> gambar penyakit daun	Sesuai	Normal

	Jeruk Citrus Healty		
<i>Testing</i> gambar daun Healthy	Melakukan <i>testing</i> gambar daun Healthy	Sesuai	Normal
<i>Testing</i> gambar penyakit daun Jeruk Citrus Mealybug	Melakukan <i>testing</i> gambar penyakit daun Jeruk Mealybug	Sesuai	Normal
<i>Testing</i> hasil Klasifikasi	Muncul hasil prediksi dan probabilitas	Sesuai	Normal

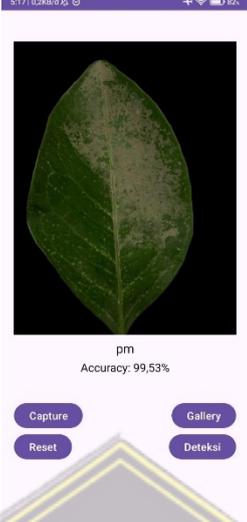
#### 4.5 Hasil *Testing* Aplikasi

Pada bagian ini akan menampilkan hasil *testing* dan analisis deteksi pada aplikasi dengan menggunakan metode *Convolutional Neural Network* serta arsitektur *Xception* yang telah melakukan uji coba dan beberapa konfigurasi untuk menentukan hasil yang tepat dan baik. Berikut hasil *testing* menggunakan aplikasi yang dapat dilihat pada tabel 4.24.

Tabel 4. 8 Hasil *Testing* Prediksi Aplikasi

No	Gambar	Keterangan
1.		Sesuai
2.		Sesuai

3.		Sesuai
4.		Sesuai

5.		Sesuai
6.		Sesuai

Pengujian aplikasi dilakukan dengan memasukkan 6 input gambar yang berasal dari kelas yang berbeda-beda. Dalam pengujian aplikasi ini output hasil klasifikasi sama dengan gambar dari kelasnya. Hal ini berarti aplikasi berjalan dengan baik dalam melakukan klasifikasi serta model Xception yang digunakan memiliki hasil yang baik

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

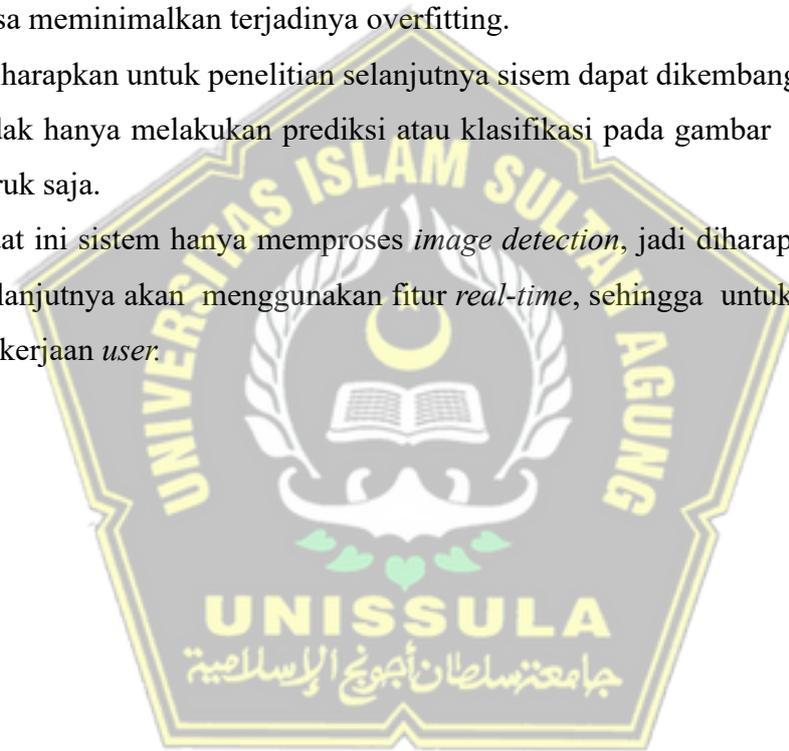
Berdasarkan hasil dari penelitian ini, dapat disimpulkan bahwa menggunakan algoritma *Convolutional Neural Network* dengan arsitektur *Xception* dapat memuat hasil akurasi yang tinggi dalam mengklasifikasikan 6 jenis penyakit daun tanaman jeruk. Hal itu telah teruji dengan perbandingan konfigurasi dan model tersebut diimplementasikan pada aplikasi mobile dengan menggunakan *Kotlin language*. Berikut beberapa poin kesimpulan penulis dapat selama penelitian ini:

1. Dengan menggunakan *Convolutional Neural Network* dan arsitektur *Xception* dapat diimplementasikan untuk deteksi penyakit daun jeruk dari *input-an* gambar dengan hasil yang cukup baik pada sistem berbasis *mobile*.
2. Pada lapisan-lapisan yang ada pada *Xception* merupakan hasil terbaik karena banyaknya lapisan tidak menjamin akurasi yang dihasilkan lebih tinggi pada penelitian ini.
3. Pada tahap *training*, konfigurasi terbaik terdapat pada konfigurasi dengan 20 epoch dengan *accuracy* sebesar  $5.6250e-04$ , *accuracy* 0.9814, *loss* 0.00934, *val-loss* 0.1260, dan *val-accuracy* 0.8333.

## 5.2 Saran

Berdasarkan penelitian yang sudah ada, penulis menyarankan untuk penelitian yang akan datang adalah :

1. Dari hasil penelitian ini, sistem membutuhkan waktu yang lama dalam meng-compile model pada epoch yang rendah. Diharapkan pada percobaan yang lain bisa memodifikasi ukuran batch size yang digunakan serta device yang memumpuni.
2. Penelitian selanjutnya dapat menambahkan jenis penyakit daun jeruk sehingga bisa meminimalkan terjadinya overfitting.
3. Diharapkan untuk penelitian selanjutnya sistem dapat dikembangkan, sehingga tidak hanya melakukan prediksi atau klasifikasi pada gambar penyakit daun jeruk saja.
4. Saat ini sistem hanya memproses *image detection*, jadi diharapkan penelitian selanjutnya akan menggunakan fitur *real-time*, sehingga untuk mempercepat pekerjaan *user*.



## DAFTAR PUSTAKA

- Ahmed, T., Muqit, A., Datta, J., Haque, M., & Haque, M. K. (2020). Prevalence and severity of different citrus diseases in Sylhet region. *Journal of Bioscience and Agriculture Research*, 23(2), 1957–1968. <https://doi.org/10.18801/jbar.230220.240>
- Akram, A., Fayakun, K., & Ramza, H. (2023). *Klasifikasi Hama Serangga pada Pertanian Menggunakan Metode Convolutional Neural Network*. 5(2), 397–406. <https://doi.org/10.47065/bits.v5i2.4063>
- Anugrah, I., Siregar, A. C., & Octariadi, B. C. (2024). Perbandingan Model Arsitektur CNN Dengan Metode Transfer Learning Untuk Klasifikasi Spesies Ikan Laut. *Progresif: Jurnal Ilmiah Komputer*, 20(1), 444. <https://doi.org/10.35889/progresif.v20i1.1834>
- Bakti, I., & Firdaus, M. (2023). Klasifikasi File Gambar Hasil X-Ray Paru -Paru Dengan Arsitektur Convolution Neural Network (CNN). *Jifotech (Journal of Information Technology)*, 3(1), 26–34.
- Cahyo, E. N., Susanti, E., & Ariyana, R. Y. (2023). Model Machine Learning Untuk Klasifikasi Kesegaran Daging Menggunakan Arsitektur Transfer Learning Xception. *Jurnal Sistem Dan Teknologi Informasi (JustIN)*, 11(2), 371. <https://doi.org/10.26418/justin.v11i2.57517>
- Chicho, B. T., & Sallow, A. B. (2021). A Comprehensive Survey of Deep Learning Models Based on Keras Framework. *Journal of Soft Computing and Data Mining*, 2(2), 49–62. <https://doi.org/10.30880/jscdm.2021.02.02.005>
- Fawwaz, M. A. A., Ramadhani, K. N., & Sthevani, F. (2020). *Klasifikasi Ras pada hewan peliharaan menggunakan Algoritma Convolutional Neural Network (CNN)*. 8(1), 715–730.
- Gaho, R. L., Ali, I. T., & Prakasa, E. (2024). Klasifikasi Kualitas Permukaan Jalan Raya Menggunakan Metode Cnn Berbasis Arsitektur Xception. *INOVTEK Polbeng - Seri Informatika*, 9(1), 354–365. <https://doi.org/10.35314/isi.v9i1.4213>
- Garofalo, P., Ventrella, D., Mastroilli, M., Palumbo, A. D., & Campi, P. (2020).

- Italian Journal of Agrometeorology climate change management. *Italian Journal of Agrometeorology*, 1, 49–62. <https://riviste.fupress.net/index.php/IJAm/issue/view/40/29>
- Hermawan, A. L. (2023). Klasifikasi Penyakit Daun Jagung Menggunakan Lightweight Convolutional Neural Network. *JIIFKOM (Jurnal Ilmiah Informatika Dan Komputer)*, 2(2), 1–7. <https://doi.org/10.51901/jiifkom.v2i2.347>
- Home, J., Picks, E., Health, P., & Health, P. (20 C.E.). *Home | Log In Citrus Huanglongbing: The Pathogen and Its Impact*. 8(June), 1–31. <https://doi.org/10.1094/PHP-2007-0906-01-RV.Introduction>
- Ibrahim, S., Gupta, R. K., War, A. R., Hussain, B., Kumar, A., Sofi, T., Noureldeen, A., & Darwish, H. (2021). Degradation of chlorpyrifos and polyethylene by endosymbiotic bacteria from citrus mealybug. *Saudi Journal of Biological Sciences*, 28(6), 3214–3224. <https://doi.org/10.1016/j.sjbs.2021.03.058>
- Kusumastuti, A. R., Kristian, Y., & Setyati, E. (2021). Klasifikasi Ketertarikan Belajar Anak PAUD Melalui Video Ekspresi Wajah Dan Gestur Menggunakan Convolutional Neural Network. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 10(2), 182–188. <https://doi.org/10.32736/sisfokom.v10i2.1146>
- Liu, S., Lou, Y., Li, Y., Zhang, J., Li, P., Yang, B., & Gu, Q. (2022). Review of phytochemical and nutritional characteristics and food applications of Citrus L. fruits. *Frontiers in Nutrition*, 9(1). <https://doi.org/10.3389/fnut.2022.968604>
- Merthaningsih, N. P., Sudiarta, I. P., & Wirya, G. N. A. S. (2020). Identification of Citrus Whitefly, Host of Entomopathogenic Fungi (*Aschersonia Placenta*) in Bali Indonesia. *International Journal of Biosciences and Biotechnology*, 7(2), 57. <https://doi.org/10.24843/ijbb.2020.v07.i02.p01>
- Naqvi, S. A. H., Wang, J., Malik, M. T., Umar, U. U. D., Ateeq-Ur-rehman, Hasnain, A., Sohail, M. A., Shakeel, M. T., Nauman, M., Hafeez-Ur-rehman, Hassan, M. Z., Fatima, M., & Datta, R. (2022). Citrus Canker—Distribution, Taxonomy, Epidemiology, Disease Cycle, Pathogen Biology, Detection, and Management: A Critical Review and Future Research Agenda. *Agronomy*,

12(5). <https://doi.org/10.3390/agronomy12051075>

- Pandia, M. (2024). Kajian Literatur Multimedia Retrieval : Machine Learning Untuk Pengenalan Wajah. *Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI)*, 7(1), 161–166. <https://doi.org/10.55338/jikomsi.v7i1.2758>
- Rasyid, M. I., & Wisudawati, L. M. (n.d.). *Klasifikasi Hama Ulat Pada Citra Daun Sawi Berbasis Convolutional Neural Network Dengan Model Xception*.
- Rinaldi, A., Mukti, N., & Prasetyaningrum, P. T. (2024). *Klasifikasi Daun Teh Klon Seri GMB Menggunakan Convolutional Neural Network dengan Arsitektur VGG16 dan Xception*. 5(1), 100–116. <https://doi.org/10.51519/journalita.v5i1.540>
- Sanjaya, D., & Budi, S. (2020). Prediksi Pencapaian Target Kerja Menggunakan Metode Deep Learning dan Data Envelopment Analysis. *Jurnal Teknik Informatika Dan Sistem Informasi*, 6(2), 288–300. <https://doi.org/10.28932/jutisi.v6i2.2678>
- Sirait, J., Pardede, J., & Informatika, P. S. (2022). Xception Dan Gated Recurrent Unit Pada Image Captioning. *Fti*, 10(4), 1–13. <https://eproceeding.itenas.ac.id/index.php/fti/article/view/968%0Ahttps://eprceeding.itenas.ac.id/index.php/fti/article/download/968/944>
- Song, Q., Li, L., & Huang, X. (2024). LELBC: A low energy lightweight block cipher for smart agriculture. *Internet of Things (Netherlands)*, 25(August 2023), 101022. <https://doi.org/10.1016/j.iot.2023.101022>
- Tuwo, M., Kuswinanti, T., Nasruddin, A., & Tambaru, E. (2024). Uncovering the presence of CVPD disease in citrus varieties of South Sulawesi, Indonesia: A molecular approach. *Journal of Genetic Engineering and Biotechnology*, 22(1), 100332. <https://doi.org/10.1016/j.jgeb.2023.100332>
- Vgg, M. A., Nasnetmobile, X., & Chandra, A. Y. (2024). *Analisis Performa Akurasi Klasifikasi Citra Jenis Sayur Salada*. 8, 1240–1249. <https://doi.org/10.30865/mib.v8i3.7661>
- Yang, X., & Yu, Z. (2024). Interplay of network information dissemination in the era of big data on environmental sustainable development and agricultural consumers' purchase decisions. *Journal of King Saud University - Science*,

36(4), 103117. <https://doi.org/10.1016/j.jksus.2024.103117>

Ahmed, T., Muqit, A., Datta, J., Haque, M., & Haque, M. K. (2020). Prevalence and severity of different citrus diseases in Sylhet region. *Journal of Bioscience and Agriculture Research*, 23(2), 1957–1968. <https://doi.org/10.18801/jbar.230220.240>

Akram, A., Fayakun, K., & Ramza, H. (2023). *Klasifikasi Hama Serangga pada Pertanian Menggunakan Metode Convolutional Neural Network*. 5(2), 397–406. <https://doi.org/10.47065/bits.v5i2.4063>

Anugrah, I., Siregar, A. C., & Octariadi, B. C. (2024). Perbandingan Model Arsitektur CNN Dengan Metode Transfer Learning Untuk Klasifikasi Spesies Ikan Laut. *Progresif: Jurnal Ilmiah Komputer*, 20(1), 444. <https://doi.org/10.35889/progresif.v20i1.1834>

Bakti, I., & Firdaus, M. (2023). Klasifikasi File Gambar Hasil X-Ray Paru -Paru Dengan Arsitektur Convolution Neural Network (CNN). *Jifotech (Journal of Information Technology)*, 3(1), 26–34.

Cahyo, E. N., Susanti, E., & Ariyana, R. Y. (2023). Model Machine Learning Untuk Klasifikasi Kesegaran Daging Menggunakan Arsitektur Transfer Learning Xception. *Jurnal Sistem Dan Teknologi Informasi (JustIN)*, 11(2), 371. <https://doi.org/10.26418/justin.v11i2.57517>

Chicho, B. T., & Sallow, A. B. (2021). A Comprehensive Survey of Deep Learning Models Based on Keras Framework. *Journal of Soft Computing and Data Mining*, 2(2), 49–62. <https://doi.org/10.30880/jscdm.2021.02.02.005>

Fawwaz, M. A. A., Ramadhani, K. N., & Sthevani, F. (2020). *Klasifikasi Ras pada hewan peliharaan menggunakan Algoritma Convolutional Neural Network (CNN)*. 8(1), 715–730.

Gaho, R. L., Ali, I. T., & Prakasa, E. (2024). Klasifikasi Kualitas Permukaan Jalan Raya Menggunakan Metode Cnn Berbasis Arsitektur Xception. *INOVTEK Polbeng - Seri Informatika*, 9(1), 354–365. <https://doi.org/10.35314/isi.v9i1.4213>

Garofalo, P., Ventrella, D., Mastroilli, M., Palumbo, A. D., & Campi, P. (2020). Italian Journal of Agrometeorology climate change management. *Italian*

*Journal of Agrometeorology*, 1, 49–62.  
<https://riviste.fupress.net/index.php/IJAm/issue/view/40/29>

Hermawan, A. L. (2023). Klasifikasi Penyakit Daun Jagung Menggunakan Lightweight Convolutional Neural Network. *JIIFKOM (Jurnal Ilmiah Informatika Dan Komputer)*, 2(2), 1–7.  
<https://doi.org/10.51901/jiifkom.v2i2.347>

Home, J., Picks, E., Health, P., & Health, P. (20 C.E.). *Home | Log In Citrus Huanglongbing: The Pathogen and Its Impact*. 8(June), 1–31.  
<https://doi.org/10.1094/PHP-2007-0906-01-RV.Introduction>

Ibrahim, S., Gupta, R. K., War, A. R., Hussain, B., Kumar, A., Sofi, T., Noureldeen, A., & Darwish, H. (2021). Degradation of chlorpyrifos and polyethylene by endosymbiotic bacteria from citrus mealybug. *Saudi Journal of Biological Sciences*, 28(6), 3214–3224. <https://doi.org/10.1016/j.sjbs.2021.03.058>

Kusumastuti, A. R., Kristian, Y., & Setyati, E. (2021). Klasifikasi Ketertarikan Belajar Anak PAUD Melalui Video Ekspresi Wajah Dan Gestur Menggunakan Convolutional Neural Network. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 10(2), 182–188. <https://doi.org/10.32736/sisfokom.v10i2.1146>

Liu, S., Lou, Y., Li, Y., Zhang, J., Li, P., Yang, B., & Gu, Q. (2022). Review of phytochemical and nutritional characteristics and food applications of Citrus L. fruits. *Frontiers in Nutrition*, 9(1).  
<https://doi.org/10.3389/fnut.2022.968604>

Merthaningsih, N. P., Sudiarta, I. P., & Wirya, G. N. A. S. (2020). Identification of Citrus Whitefly, Host of Entomopathogenic Fungi (*Aschersonia Placenta*) in Bali Indonesia. *International Journal of Biosciences and Biotechnology*, 7(2), 57. <https://doi.org/10.24843/ijbb.2020.v07.i02.p01>

Naqvi, S. A. H., Wang, J., Malik, M. T., Umar, U. U. D., Ateeq-Ur-rehman, Hasnain, A., Sohail, M. A., Shakeel, M. T., Nauman, M., Hafeez-Ur-rehman, Hassan, M. Z., Fatima, M., & Datta, R. (2022). Citrus Canker—Distribution, Taxonomy, Epidemiology, Disease Cycle, Pathogen Biology, Detection, and Management: A Critical Review and Future Research Agenda. *Agronomy*, 12(5). <https://doi.org/10.3390/agronomy12051075>

- Pandia, M. (2024). Kajian Literatur Multimedia Retrieval : Machine Learning Untuk Pengenalan Wajah. *Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI)*, 7(1), 161–166. <https://doi.org/10.55338/jikomsi.v7i1.2758>
- Rasyid, M. I., & Wisudawati, L. M. (n.d.). *Klasifikasi Hama Ulat Pada Citra Daun Sawi Berbasis Convolutional Neural Network Dengan Model Xception*.
- Rinaldi, A., Mukti, N., & Prasetyaningrum, P. T. (2024). *Klasifikasi Daun Teh Klon Seri GMB Menggunakan Convolutional Neural Network dengan Arsitektur VGG16 dan Xception*. 5(1), 100–116. <https://doi.org/10.51519/journalita.v5i1.540>
- Sanjaya, D., & Budi, S. (2020). Prediksi Pencapaian Target Kerja Menggunakan Metode Deep Learning dan Data Envelopment Analysis. *Jurnal Teknik Informatika Dan Sistem Informasi*, 6(2), 288–300. <https://doi.org/10.28932/jutisi.v6i2.2678>
- Sirait, J., Pardede, J., & Informatika, P. S. (2022). Xception Dan Gated Recurrent Unit Pada Image Captioning. *Fti*, 10(4), 1–13. <https://eproceeding.itenas.ac.id/index.php/fti/article/view/968%0Ahttps://eproceeding.itenas.ac.id/index.php/fti/article/download/968/944>
- Song, Q., Li, L., & Huang, X. (2024). LELBC: A low energy lightweight block cipher for smart agriculture. *Internet of Things (Netherlands)*, 25(August 2023), 101022. <https://doi.org/10.1016/j.iot.2023.101022>
- Tuwo, M., Kuswinanti, T., Nasruddin, A., & Tambaru, E. (2024). Uncovering the presence of CVPD disease in citrus varieties of South Sulawesi, Indonesia: A molecular approach. *Journal of Genetic Engineering and Biotechnology*, 22(1), 100332. <https://doi.org/10.1016/j.jgeb.2023.100332>
- Vgg, M. A., Nasnetmobile, X., & Chandra, A. Y. (2024). *Analisis Performa Akurasi Klasifikasi Citra Jenis Sayur Salada*. 8, 1240–1249. <https://doi.org/10.30865/mib.v8i3.7661>
- Yang, X., & Yu, Z. (2024). Interplay of network information dissemination in the era of big data on environmental sustainable development and agricultural consumers' purchase decisions. *Journal of King Saud University - Science*, 36(4), 103117. <https://doi.org/10.1016/j.jksus.2024.103117>