

**DETEKSI JENIS PENYAKIT TANAMAN JAMBU BIJI  
BERDASARKAN CITRA DAUN DAN BUAH  
MENGGUNAKAN YOLO V-8**

**SKRIPSI**

**CHRISTOPHER MIANDO IMANUEL MANURUNG**

**191402117**



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA**

**2024**

**DETEKSI JENIS PENYAKIT TANAMAN JAMBU BIJI BERDASARKAN  
CITRA DAUN DAN BUAH MENGGUNAKAN  
YOLO V-8**

**SKRIPSI**

**Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah  
Sarjana Teknologi Informasi**

**CHRISTOPHER MIANDO IMANUEL MANURUNG**

**191402117**



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA**

**2024**

## PERSETUJUAN

Judul : DETEKSI JENIS PENYAKIT TANAMAN JAMBU BIJI BERDASARKAN CITRA DAUN DAN BUAH MENGGUNAKAN YOLO V-8

Kategori : SKRIPSI

Nama : CHRISTOPHER MIANDO IMANUEL MANURUNG

Nomor Induk Mahasiswa : 191402117

Program Studi : SARJANA (S1) TEKNOLOGI INFORMASI

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

Medan, 12 Juli 2024

Komisi Pembimbing :

Pembimbing 2,

Pembimbing 1,

Ade Sarah Huzaifah S.Kom., M.Kom.  
NIP. 198506302018032001

Ivan Jaya S.Si., M.Kom.  
NIP. 198407072015041001

Diketahui/disetujui oleh  
Program Studi S1 Teknologi Informasi  
Ketua,

Dedy Arisandi S.T., M.Kom.  
NIP. 197908312009121002



## **PERNYATAAN**

DETEKSI JENIS PENYAKIT TANAMAN JAMBU BIJI BERDASARKAN  
CITRA DAUN DAN BUAH MENGGUNAKAN  
YOLO V-8

## **SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 12 Juli 2024

Christopher Miando Imanuel Manurung  
NIM. 191402117

## KATA PENGANTAR

Puji dan Syukur kepada Tuhan Yang Maha Esa, yang senantiasa memberikan berkat, rahmat, dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi ini sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi S1 Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.

Dalam proses menyelesaikan penulisan tugas akhir ini, penulis telah menerima banyak bimbingan, dukungan, bantuan, serta doa dari berbagai pihak. Pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada:

1. Keluarga penulis, Bapak Saut Maruli Manurung dan Ibu Eva Yohanna Sitorus selaku orang tua penulis yang telah mendidik dan membesarkan penulis, selalu memberikan semangat dan doa terbaik untuk penulis, begitu juga dengan abang penulis Yohanes Minardo Halomoan Manurung, serta adik penulis yang terkasih Abraham Mindo Hamonangan Manurung yang senantiasa memberikan doa dan dukungan dalam kondisi apapun.
2. Bapak Ivan Jaya S.Si., M.Kom. selaku Dosen Pembimbing 1 yang telah banyak membimbing dan mengarahkan, serta memberikan saran pada penulis dalam proses penggerjaan skripsi ini.
3. Ibu Ade Sarah Huzaifah S.Kom., M.Kom. selaku Dosen Pembimbing 2 yang juga sangat banyak membantu penulis dalam membimbing serta memberi saran dan arahan dalam proses penggerjaan skripsi.
4. Ibu Dr. Maya Silvi Lydia, B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
5. Bapak Dedy Arisandi ST., M.Kom. selaku Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
6. Seluruh Dosen Program Studi S1 Teknologi Informasi Sumatera Utara yang telah mengajar dan memberikan ilmu yang berguna selama masa perkuliahan penulis.
7. Seluruh Staff dan Pegawai Fasilkom-TI Universitas Sumatera Utara yang telah membantu segala urusan administrasi selama masa perkuliahan dan menyelesaikan skripsi penulis.
8. Teman-teman penulis yang turut langsung membantu penulis dalam menyelesaikan tugas akhir yaitu Vincent Sirait, Daniel Situmeang, Jhuan Sitorus,

Samuel Malau, Aulia Rahman, Geylfedra Panggabean, Anri Marpaung, Nhakchok Gemilang, Grace Ogestin, Monang Limbong, Muhammad Zikri, Brian Tarihoran, Anggi Pardede, Timothy Agalliasis, Sarah Sinurat, Vania Siahaan, Dhea Sitompul, Tommy Parlindungan, dan masih banyak lagi yang tidak bisa penulis sebutkan satu persatu.

9. Teman-teman Teknologi Informasi angkatan 19, yang telah membersamai penulis selama masa perkuliahan.
10. Kepada Senior, Junior, Kerabat, teman-teman dan pihak lainnya yang juga berperan dalam memberikan semangat dan dukungan serta doa selama masa perkuliahan dan dalam menyelesaikan penulisan skripsi.

Medan, 12 Juli 2024  
Penulis

Christopher Miando Imanuel Manurung  
191402117

## ABSTRAK

Jambu biji (*Psidium guajava*) adalah buah yang populer di wilayah tropis dan subtropis karena rasa manisnya. Meskipun memiliki potensi pasar yang baik, penanaman intensif jambu biji di Indonesia masih jarang, sehingga produksinya rendah. Salah satu hambatan utama dalam meningkatkan produktivitas adalah serangan penyakit, yang biasanya didiagnosis berdasarkan gejala morfologi, namun sering kurang akurat jika hanya menggunakan pengamatan mata telanjang. Oleh karena itu, diperlukan sistem yang dapat mendeteksi penyakit pada daun dan buah jambu biji secara lebih tepat. Penelitian ini mengimplementasikan algoritma *You Only Look Once* (YOLO) versi 8 untuk mendeteksi empat jenis penyakit pada tanaman jambu biji: *antraknosa*, *phytophthora* (jamur air), *red rust* (karat merah), dan *scab* (bercak buah) dengan jumlah data yang digunakan sebanyak 1.920 data yang terdiri dari 1.536 data sebagai data train, 192 data sebagai data validation, 192. Hasil menunjukkan bahwa algoritma YOLOv8 mampu mendeteksi penyakit secara *real-time* dengan nilai *accuracy* 94,27%, *precision* 94,60%, *recall* 94,26%, dan F1-Score 94,31%. Model ini juga mampu memprediksi lebih dari satu objek penyakit dalam satu frame dengan catatan objek tersebut berada dalam jarak optimal di depan kamera. Dari hasil tersebut menunjukkan bahwa sistem yang dibuat menggunakan algoritma YOLO V-8 telah berhasil dengan baik dalam mendeteksi penyakit pada tanaman jambu biji.

**Kata Kunci:** Penyakit Tanaman Jambu Biji, YOLOv8, Deteksi Objek, *Real-time*, Citra Digital

## ***Guava Plant Disease Detection Based on Leaf and Fruit Images Using Yolo-V8***

### ***ABSTRACT***

*Guava (*Psidium guajava*) is a popular fruit in tropical and subtropical regions due to its sweet flavour. Despite its good market potential, intensive planting of guava in Indonesia is still rare, resulting in low production. One of the main obstacles in increasing productivity is disease infestation, which is usually diagnosed based on morphological symptoms, but is often inaccurate if only using naked eye observation. Therefore, a system is needed that can detect diseases on guava leaves and fruits more precisely. This research implements the You Only Look Once (YOLO) algorithm version 8 to detect four types of diseases on guava plants: anthracnose, phytophthora (water mould), red rust, and scab (fruit spot) with a total of 1,920 data used consisting of 1,536 data as train data, 192 data as validation data, 192. The results show that the YOLOv8 algorithm is able to detect diseases in real-time with an accuracy value of 94.27%, precision 94.60%, recall 94.26%, and F1-Score 94.31%. This model is also able to predict more than one disease object in one frame provided that the object is within the optimal distance in front of the camera. These results show that the system created using the YOLO V-8 algorithm has succeeded well in detecting diseases in guava plants.*

***Keywords:*** ***Guava Plant Disease, YOLOv8, Object Detection, Real-time, Digital Imagery***

## DAFTAR ISI

<b>PERSETUJUAN</b>	<b>Error! Bookmark not defined.</b>
<b>PERNYATAAN</b>	ii
<b>KATA PENGANTAR</b>	iii
<b>ABSTRAK</b>	v
<b>ABSTRACT</b>	vi
<b>DAFTAR ISI</b>	vii
<b>DAFTAR GAMBAR</b>	x
<b>DAFTAR TABEL</b>	xii
<b>BAB I PENDAHULUAN</b>	1
<b>1.1 Latar Belakang</b>	1
<b>1.2 Rumusan Masalah</b>	4
<b>1.3 Batasan Masalah</b>	5
<b>1.4 Tujuan Penelitian</b>	5
<b>1.5 Manfaat Penelitian</b>	5
<b>1.6 Metodologi Penelitian</b>	6
<b>1.7 Sistematika Penulisan dan Dokumentasi</b>	7
<b>BAB II LANDASAN TEORI</b>	8
<b>2.1 Jambu Biji</b>	8
<b>2.2 Penyakit Tanaman Jambu Biji</b>	9
<b>2.2.1 Antraknosa</b>	9
<b>2.2.2 Red Rust (Karat Merah)</b>	10
<b>2.2.3 Scab (Bercak Buah)</b>	11
<b>2.2.4 Phytophthora (Jamur Air)</b>	12
<b>2.3 Citra Digital</b>	13
<b>2.3.1 Citra Biner</b>	13

2.3.2 Citra Warna	14
<b>2.4 Realtime Object Detection</b>	14
<b>2.5 Convolutional Neural Network (CNN)</b>	15
<b>2.6 Algoritma You Only Look Once (YOLO)</b>	17
2.6.1 YOLO v5	17
2.6.2 YOLO v7	18
2.6.3 YOLO v8	19
<b>2.7 Perbandingan YOLOv5, YOLOv7, YOLOv8</b>	20
<b>2.8 Confusion Matrix</b>	21
2.8.1 Accuracy	22
2.8.2 Precision	23
2.8.3 Recall	23
2.8.4 F1-Score	23
<b>2.9 Penelitian Terdahulu</b>	23
<b>BAB III ANALISIS DAN PERANCANGAN SISTEM</b>	29
<b>3.1 Dataset</b>	29
<b>3.2 Analisis Sistem</b>	30
3.2.1 <i>Image Acquisition</i>	31
3.2.2 <i>Pre-Processing</i>	31
3.2.3 <i>Image Classification</i>	34
3.2.4 <i>Proses Training</i>	36
3.2.5 <i>Learned Model</i>	36
3.2.6 <i>TF-Lite Model</i>	37
3.2.7 <i>Proses Pengujian</i>	37
<b>3.3 Rancangan Antarmuka Sistem</b>	37
3.3.1 Tampilan splash screen	38
3.3.2 Tampilan Beranda	38
3.3.3 Tampilan Halaman Artikel	39
3.3.4 Tampilan Halaman Deteksi	40
3.3.5 Tampilan halaman <i>about</i>	40
<b>BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM</b>	41

<b>4.1 Implementasi Sistem</b>	41
<b>4.2 Implementasi Data</b>	42
<b>4.3 Implementasi Perancangan Antarmuka</b>	43
4.3.1 Tampilan Halaman <i>Splash Screen</i>	43
4.3.2 Tampilan Halaman Beranda	44
4.3.3 Tampilan Halaman Artikel	45
4.3.4 Tampilan Halaman Deteksi	46
4.3.5 Tampilan Halaman Tentang	47
<b>4.4 Pelatihan Sistem</b>	48
<b>4.5 Pengujian Sistem</b>	50
4.5.1 Pengujian Akurasi	50
4.5.2 Pengujian Dengan Jarak	61
4.5.3 Pengujian <i>Angle</i>	63
<b>BAB V KESIMPULAN DAN SARAN</b>	66
<b>5.1 Kesimpulan</b>	66
<b>5.2 Saran</b>	67
<b>DAFTAR PUSTAKA</b>	68

## DAFTAR GAMBAR

Gambar 2. 1 Tanaman Jambu Biji	9
Gambar 2. 2 Penyakit Antraknosa	10
Gambar 2. 3 Penyakit Red Rust	11
Gambar 2. 4 Penyakit <i>Scab</i>	12
Gambar 2. 5 Penyakit <i>Phytophthora</i>	13
Gambar 2. 6 Citra Biner (Kusumanto & Tompunu, 2011)	14
Gambar 2. 7 Citra Warna RGB (Kusumanto & Tompunu, 2011)	14
Gambar 2. 8 Arsitektur Umum CNN (Sumber: <a href="https://nafizshahriar.medium.com">https://nafizshahriar.medium.com</a> )	16
Gambar 2. 9 Arsitektur Umum YOLO (Sumber: <a href="https://rubikscode.net">https://rubikscode.net</a> )	17
Gambar 2. 10 Arsitektur YOLO v8 (Solawetz, J., & Francesco, 2023)	20
Gambar 2. 11 Rata-rata YOLO mAP@.50 dibandingkan kategori RF100 (Solawetz, J., & Francesco, 2023)	21
Gambar 3. 1 Contoh Data Citra Buah Jambu Biji yang Terserang Penyakit	29
Gambar 3. 2 Arsitektur Umum	30
Gambar 3. 3 Data yang Diberi Label	31
Gambar 3. 4 Pemberian Label dan Bounding Box Pada Citra	32
Gambar 3. 5 Isi File Txt Yang Diberi Label	32
Gambar 3. 6 Contoh Data Sebelum dan Setelah <i>Resizing</i>	33
Gambar 3. 7 Penerapan <i>Resize</i> Pada Platform <i>Roboflow</i>	33
Gambar 3. 8 Contoh Nilai RGB Pada Citra	34
Gambar 3. 9 Piramida Fitur hasil SPPF	35
Gambar 3. 10 Proses YOLOv8 hingga pemberian <i>bounding box</i> dan <i>class</i>	36
Gambar 3. 11 Tampilan <i>Splash Screen</i>	38
Gambar 3. 12 Rancangan Tampilan Halaman Beranda	39
Gambar 3. 13 Rancangan Tampilan Halaman Artikel	39
Gambar 3. 14 Rancangan Tampilan Halaman Deteksi	40
Gambar 3. 15 Rancangan Tampilan Halaman <i>About</i>	40
Gambar 4. 1 Beberapa Contoh Data Citra Penyakit <i>Antraknosa</i>	42
Gambar 4. 2 Beberapa Contoh Data Citra Penyakit <i>Phytophthora</i>	42
Gambar 4. 3 Beberapa Contoh Data Citra Penyakit <i>Red Rust</i>	43

Gambar 4. 4 Beberapa Contoh Data Citra Penyakit <i>Scab</i>	43
Gambar 4. 5 Tampilan <i>Splash Screen</i>	44
Gambar 4. 6 Tampilan <i>Home</i>	45
Gambar 4. 7 Tampilan Artikel	46
Gambar 4. 8 Tampilan Deteksi	47
Gambar 4. 9 Tampilan <i>About</i>	47
Gambar 4. 10 <i>Google Colab</i>	48
Gambar 4. 11 Grafik Pelatihan	49

**DAFTAR TABEL**

Tabel 2. 1 Taksonomi Jambu Biji	8
Tabel 2. 2 <i>Confusion Matrix</i>	22
Tabel 2. 3 Penelitian Terdahulu	26
Tabel 3. 1 Split Data	28
Tabel 4. 1 Hasil Percobaan <i>Training</i>	49
Tabel 4. 2 Hasil Uji Sistem	51
Tabel 4. 3 <i>Confusion Matrix</i>	58
Tabel 4. 4 Nilai Pengujian TP, FP, dan FN	58
Tabel 4. 5 Nilai Precision, Recall, dan F1-Score	60
Tabel 4. 6 Hasil Pengujian Pada Jarak	61
Tabel 4. 7 Hasil Pengujian <i>Angle</i>	63
Tabel 4. 8 Pengujian <i>Angle</i> 70 Sampai 90 Derajat Dari Permukaan Daun	65

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Jambu biji adalah sejenis buah-buahan yang dikenal karena rasa manis dan segarnya. Buah ini tumbuh di berbagai wilayah tropis dan subtropis di seluruh dunia dan memiliki banyak varietas yang berbeda. Tanaman jambu biji ditanam hampir di seluruh wilayah Nusantara. Namun masyarakat Indonesia masih sedikit yang menanam jambu biji secara intensif sehingga produksi jambu berkualitas rendah dan harganya pun menjadi rendah. Padahal, jambu biji merupakan salah satu komoditas buah yang memiliki pasaran prospektif, baik untuk pasaran di dalam negeri maupun pasaran di luar negeri (Cahyono, 2010).

Tanaman jambu biji merupakan salah satu tanaman buah yang cukup mudah untuk dipelihara oleh masyarakat. Pada umumnya, tingkat permintaan pasar produk jambu biji relatif tinggi sehingga produk tanaman jambu biji sering dijadikan sebagai peluang usaha dan terus dikembangkan. Terdapat beberapa permasalahan dalam usaha perkebunan tanaman jambu biji, salah satunya adalah serangan penyakit pada tanaman jambu biji yang merupakan menjadi faktor penghambat dalam meningkatkan produktivitas pertanian, serta mengganggu fisiologi dan pertumbuhan tanaman. Diagnosis penyakit pada tanaman jambu kristal biasanya dilakukan oleh seorang peneliti tanaman yang didasarkan pada gejala yang timbul dan teramat perubahan morfologinya (Raharjo, 2014). Namun, perbedaan fisik jambu biji tidak bisa dilihat secara mendetail oleh kebanyakan orang dengan mata telanjang, sehingga perlu dibangun sebuah sistem yang dapat membantu dalam mendekripsi daun tanaman jambu sekaligus buahnya karena diagnosis penyakit berdasarkan pengamatan optik pada buah dan daun tanaman jambu yang dilakukan biasanya mengalami kekeliruan atau kurang tepat.

Telah dilakukan beberapa penelitian terdahulu terhadap penyakit pada daun jambu biji menggunakan metode D-CNN, salah satunya adalah penelitian yang dilakukan oleh Md. Rasel Howlader (2019) mengenai Pengenalan Otomatis Penyakit Daun Jambu Biji dimana pada penelitian ini dilakukan pendekatan

berbasis Deep Convolution Neural Network (D-CNN) yang sesuai untuk mengidentifikasi penyakit daun jambu biji secara otomatis. Model yang diusulkan berlaku untuk mengklasifikasikan penyakit utama pada daun jambu biji seperti Bercak Daun Alga, Kutu Kebul, dan Karat. Berdasarkan pernyataan pada jurnal terkait, ini adalah pertama kalinya menggunakan model berbasis D-CNN untuk mengenali penyakit daun jambu biji. Selain itu, peneliti membuat dataset sendiri yaitu BU\_Guava\_Leaf (BUGL2018) dengan empat kategori berbeda. Pada kumpulan data ini, peneliti melatih pendekatan identifikasi penyakit yang diusulkan dan mengevaluasi hasil eksperimen yang menunjukkan akurasi rata-rata 98,74% pada set pengujian.

Penelitian terdahulu yang lainnya pernah dilakukan oleh Rajasekaran Thangaraj, et al (2023) terkait dengan perbandingan Model Deep Learning Deteksi Penyakit Daun Jambu Biji dimana penelitian ini menyatakan bahwa beberapa Model Deep Learning yang disajikan dapat digunakan untuk mengkategorikan penyakit utama daun jambu biji antara lain Canker, Dot, Healthy, Mummification, dan Rust. Dataset *kaggle* digunakan untuk mengumpulkan gambar jambu biji yang mengidap penyakit *Canker, Dot, Healthy, Mummification* dan *Rust*. Dalam prosesnya, model CNN seperti *DenseNet121*, *DenseNet169*, *InceptionV3*, dan *Xception* digunakan untuk mengkategorikan penyakit yang menyerang daun jambu biji. Temuan eksperimental menunjukkan bahwa, dengan akurasi 96,12%, model *DenseNet169* mengungguli model lainnya. Ketika efektivitas model dinilai menggunakan metrik kinerja seperti presisi, perolehan, dan skor F1, hasilnya menunjukkan bahwa *DenseNet169* memberikan hasil yang sangat baik.

Telah dilakukan beberapa penelitian terdahulu terhadap daun teh menggunakan algoritma YOLO (*You Only Look Once*), salah satunya adalah penelitian yang dilakukan oleh Md. Janibul Alam Soeb, et al (2023) mengenai Penyakit daun teh dideteksi secara manual, menambah waktu dan mempengaruhi kualitas hasil dan produktivitas. Penelitian ini bertujuan untuk menyajikan solusi berbasis kecerdasan buatan terhadap permasalahan tersebut deteksi penyakit daun teh dengan melatih model deteksi objek satu tahap tercepat, YOLOv7, aktif kumpulan data daun teh berpenyakit yang dikumpulkan dari empat kebun teh terkemuka di Bangladesh. 4000 digital gambar lima jenis penyakit daun dikumpulkan dari

kebun teh ini, dibuat secara manual kumpulan data gambar penyakit daun yang ditambah dengan data dan diberi anotasi. Penelitian ini menggabungkan augmentasi data pendekatan untuk memecahkan masalah ukuran sampel yang tidak mencukupi. Hasil deteksi dan identifikasi untuk pendekatan YOLOv7 divalidasi oleh metrik statistik terkemuka seperti akurasi deteksi, presisi, perolehan, nilai mAP, dan skor F1, yang menghasilkan 97,3%, 96,7%, 96,4%, 98,2%, dan 0,965, masing-masing. Hasil percobaan menunjukkan bahwa YOLOv7 untuk penyakit daun teh di alam gambar lebih unggul dari jaringan deteksi dan identifikasi target yang ada, termasuk CNN, Deep CNN, DNN, AX-Retina Net, peningkatan DCNN, YOLOv5, dan segmentasi gambar multi-objektif. Oleh karena itu, ini Penelitian ini diharapkan dapat meminimalkan beban kerja ahli entomologi dan membantu dalam identifikasi cepat dan deteksi penyakit daun teh, sehingga meminimalkan kerugian ekonomi.

Pada 10 Januari 2023 yang lalu Glenn Jocher memperkenalkan model YOLOv8 yang merupakan model YOLO tercanggih terbaru yang dapat digunakan untuk tugas deteksi objek, klasifikasi gambar, dan segmentasi instan. Dari semua varian YOLO sebelumnya, YOLOv5 merupakan model terbaik dalam melakukan deteksi objek. Berdasarkan pembaharuan yang ada YOLOv5 memiliki kemudahan dalam penggunaannya, namun YOLOv8 memiliki keunggulan dalam hal kecepatan dan ketepatan dalam mendekripsi objek dibandingkan YOLO versi sebelumnya. Dalam mendekripsi penyakit tanaman menggunakan YOLOv8, telah dilakukan penelitian oleh Syed Asif Ahmad Qadri, et al (2023) terkait pendekripsi dan segmentasi penyakit tanaman dimana penelitian ini menyatakan bahwa model YOLOv8, yang merupakan kemajuan dari seri YOLO, telah dirancang untuk meningkatkan kecepatan deteksi tanpa mengorbankan akurasi. Arsitekturnya yang rumit, terdiri dari beberapa lapisan konvolusional, memungkinkan ekstraksi fitur kompleks dari gambar, sehingga menghasilkan identifikasi penyakit daun tanaman secara tepat. Ketika model ini dilatih secara end-to-end, model tersebut dapat secara efektif mempelajari dan menggeneralisasi data masukan, sehingga meningkatkan kinerja prediktifnya untuk kejadian penyakit daun yang belum pernah terjadi sebelumnya atau yang baru. Hasil evaluasi untuk pendekatan YOLOv8 divalidasi oleh metrik statistik yang menonjol seperti presisi, perolehan, nilai mAP50 dan mAP50-95, serta skor F1, yang

menghasilkan 99,8%, 99,3%, 99,5%, 96,5%, dan 0,999 untuk kotak pembatas dan 99,1%, 99,3%, 99,3%, 98,5% dan 0,992 masing-masing untuk masker segmentasi. Hasilnya menunjukkan performa model yang kuat dalam mendeteksi dan mensegmentasi wilayah yang sakit secara akurat, seperti yang ditunjukkan oleh nilai presisi, perolehan, dan peta peta yang tinggi. Temuan ini menyoroti efektivitas model YOLOv8 dalam deteksi penyakit tanaman, dan menunjukkan potensinya dalam penerapan pertanian presisi dan pengelolaan tanaman.

Berdasarkan latar belakang di atas beserta dengan hasil dari penelitian-penelitian terdahulu, maka peneliti melakukan penelitian dalam Tugas Akhir dengan judul **“Deteksi Jenis Penyakit Tanaman Jambu Biji Berdasarkan Citra Daun dan Buah Menggunakan YOLO V-8”** dimana objek dalam penelitian ini adalah Pertanian Jambu Biji Acun yang mana pada pertanian tersebut para petani tidak menggunakan istilah ilmiah dari jenis-jenis penyakit yang ada pada tanaman jambu biji. Para petani tersebut tidak mengetahui keseluruhan gejala penyakit yang ada pada tanaman jambu biji dan menganggap gejala tersebut adalah hal yang biasa yang terjadi pada tanaman jambu biji dan menganggap hal itu bukanlah sebuah penyakit yang menjangkit tanaman jambu biji padahal hal tersebut sudah menjadi indikasi bahwasanya tanaman jambu biji dengan gejala tersebut sudah dikatakan terjangkit penyakit.

## 1.2 Rumusan Masalah

Jambu biji merupakan salah satu dari jenis buah-buahan yang saat ini banyak dikonsumsi oleh masyarakat karena khasiatnya yang cukup tinggi. Sejak semakin banyaknya peminat dalam mengkonsumsi jambu biji, produk tanaman jambu biji sering dijadikan sebagai peluang usaha dan terus dikembangkan. Dikarenakan banyaknya yang menjadikan jambu sebagai peluang usaha, banyak masyarakat yang belum mengetahui cara membedakan tanaman jambu biji yang terkena penyakit. Dalam kasus ini, penyakit pada tanaman jambu biji bisa dilihat dari citra buah maupun daun tanaman. Namun, ada kalanya terdapat kekeliruan dalam mendeteksi dan mendiagnosa jenis penyakit yang terdapat pada buah dan daun tanaman jambu apabila hanya dilihat dengan mata telanjang. Oleh karena itu, dibutuhkan sebuah sistem yang dapat membantu para petani di Pertanian Jambu Biji Acun maupun orang-orang yang membutuhkan dalam mendeteksi dan

mengenali jenis penyakit pada tanaman jambu biji berdasarkan citra buah dan daunnya.

### **1.3 Batasan Masalah**

Agar penelitian ini tetap di arah yang benar, maka diperlukan batasan masalah. Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penyakit yang dapat dideteksi hanya jenis penyakit *antraknosa*, *red rust* (karat merah), *Phytophthora* (jamur air), dan *scab*
2. *Output* berupa nama *class* penyakit jambu biji yang dideteksi dari citra daun atau pun buah beserta nilai confidencenya
3. Implementasi model hanya dalam bentuk aplikasi berbasis mobile
4. Data yang digunakan berekstensi .jpg/.jpeg

### **1.4 Tujuan Penelitian**

Tujuan penelitian ini adalah untuk mengimplementasikan metode *You Only Look Once* (YOLO) versi 8 dalam mendeteksi jenis penyakit jambu biji berdasarkan citra daun dan buah.

### **1.5 Manfaat Penelitian**

Adapun manfaat penelitian yang diperoleh dari penelitian ini adalah sebagai berikut:

1. Membantu para petani jambu biji dalam mengidentifikasi jenis penyakit yang terletak pada buah maupun daun tanaman jambu biji.
2. Bagi perguruan tinggi, dapat dijadikan sebagai sumber referensi untuk pengembangan penelitian-penelitian selanjutnya mengenai pengolahan citra digital dan metode YOLO versi 8.
3. Bagi masyarakat, dapat membantu dalam mengenali serta dapat membantu dalam tindak penanganan terhadap tanaman yang terkena penyakit. Serta langkah dasar untuk memberikan ilmu pengetahuan guna menarik minat masyarakat akan kemajuan teknologi di Indonesia.

## 1.6 Metodologi Penelitian

Tahapan-tahapan yang akan dilakukan dalam penelitian ini adalah sebagai berikut:

### 1. Studi Literatur

Pada tahap ini merupakan proses mengumpulkan data daun dan buah jambu biji dan mempelajari referensi yang berkaitan dengan penelitian mengenai *image processing* dan metode *You Only Look Once version8* yang diperoleh dari berbagai jurnal, artikel, buku, skripsi dan sumber referensi lainnya yang berkaitan dengan penelitian ini.

### 2. Analisis Permasalahan

Pada tahapan ini dilakukan analisis studi literatur yang telah dikumpulkan pada tahap sebelumnya untuk mendapatkan pemahaman tentang metode yang akan diterapkan dalam penelitian ini yaitu melakukan deteksi jenis penyakit pada tanaman jambu biji menggunakan *You Only Look Once version8*.

### 3. Perancangan Sistem

Langkah-langkah dalam tahapan perancangan sistem meliputi tahapan perancangan keseluruhan, seperti pembuatan arsitektur, pengumpulan data, pembagian data *training* dan data *testing*, serta perancangan antarmuka berdasarkan hasil pada tahap sebelumnya.

### 4. Implementasi

Pada tahap ini dilakukannya tindakan pengujian terhadap model *You Only Look Once* yang sudah dibangun dengan menggunakan testing data yang sudah dipisahkan pada pembuatan arsitektur untuk menghasilkan output yang sesuai dengan tujuan penelitian.

### 5. Pengujian Sistem

Pada bagian ini, sistem yang sudah dibangun sebelumnya akan dilakukan tahap pengujian yang bertujuan untuk memastikan bahwa sistem yang telah dibangun dapat mendeteksi jenis penyakit tanaman jambu biji dapat digunakan dengan akurat dan sesuai dengan tujuan.

### 6. Dokumentasi dan Penyusunan Laporan

Pada tahap ini dilakukan dokumentasi dan penyusunan laporan yang menjabarkan hasil dari penelitian yang telah dilakukan.

## **1.7 Sistematika Penulisan dan Dokumentasi**

Sistem penulisan skripsi ini terdiri dari lima bagian, yaitu:

### **BAB I : Pendahuluan**

Bab ini berisi tentang latar belakang, rumusan masalah, Batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

### **BAB II : Landasan Teori**

Bab ini berisi tentang penjelasan dari teori-teori yang relevan dengan sistem dan permasalahan yang ditangkap dalam penelitian ini.

### **BAB III : Analisis dan Perancangan Sistem**

Bab ini berisi tentang arsitektur umum dan metode yang diterapkan dalam pembuatan sistem.

### **BAB IV : Implementasi dan Pengujian**

Bab ini berisikan implementasi dari hasil analisis dan perancangan sistem yang telah dibahas pada bab sebelumnya dan menampilkan hasil pengujian terhadap sistem yang telah dibangun.

### **BAB V : Kesimpulan dan Saran**

Bab ini berisi tentang kesimpulan dari hasil penelitian dan saran untuk penelitian-penelitian selanjutnya.

## BAB II

### LANDASAN TEORI

#### **2.1 Jambu Biji**

Jambu biji (*Psidium guajava*) adalah salah satu buah tropis yang paling dikenal dan dikonsumsi di seluruh dunia. Buah ini berasal dari Amerika Tengah dan Amerika Selatan dan telah menyebar luas ke berbagai wilayah tropis di seluruh dunia. Secara botani, jambu biji adalah anggota suku Myrtaceae dan memiliki karakteristik daun yang biasanya berwarna hijau mengkilap dan berbentuk oval. Buahnya, yang bisa berwarna hijau atau kuning saat matang, memiliki rasa manis dan sedikit asam serta aroma yang khas. Jambu biji dikenal memiliki kandungan vitamin C yang tinggi, serat, dan senyawa fitokimia yang berpotensi memberikan manfaat kesehatan. Selain itu, jambu biji juga mengandung nutrisi lain, seperti serat, zat besi, protein, magnesium, serta folat, meski dalam jumlah yang sedikit. Selain itu jambu biji juga memiliki kandungan antioksidan yang terdapat pada buah dan daunnya (Bakara, 2020).

**Tabel 2. 1** Taksonomi Jambu Biji

Kingdom	Plantae
Sub Kingdom	Tracheobionta
Divisi	Magnoliophyta
Sub Divisi	Spermatophyta
Kelas	Magnoliopsida
Sub Kelas	Rosidae
Ordo	Myrales
Famili	Myrtaceae
Genus	<i>Psidium</i>
Spesies	<i>Psidium guajava L.</i>

Menurut Badan Pusat Statistik (2023), produksi jambu biji di Indonesia tahun 2021 – 2022 mengalami fluktuasi. Dimana pada tahun 2021 produksi jambu biji sebesar 422.491 ton. Pada tahun 2022 mengalami peningkatan produksi sebesar kurang lebih 50.000 ton menjadi 472.686 ton. Terjadinya fluktuasi produksi jambu biji diakibatkan oleh beberapa faktor seperti perubahan cuaca dan iklim yang tidak

menentu. Bukan hanya faktor alam saja, tingkat serangan hama dan penyakit juga dapat mempengaruhi produksi jambu biji karena dapat menyebabkan kehilangan hasil karena adanya penurunan kualitas yang terjadi pada tanaman jambu biji yang terserang penyakit maupun hama. Kurangnya ketersediaan bibit atau benih unggul juga bisa mengakibatkan terbatasnya jumlah tanaman dan berkurangnya produksi jambu biji.



**Gambar 2. 1 Tanaman Jambu Biji**

## **2.2 Penyakit Tanaman Jambu Biji**

Jambu biji merupakan salah satu buah yang diminati oleh masyarakat, sehingga banyak petani yang melihat potensi keuntungan dalam usaha berkebunnya. Namun, kualitas dari hasil panen tanaman jambu biji dapat mengalami penurunan akibat serangan penyakit. Bagian dari tanaman jambu biji yang sering kali terdampak adalah daun dan buah. Terdapat beberapa jenis penyakit yang kerap menyerang tanaman jambu biji, diantaranya adalah penyakit *antraknosa*, karat merah, jamur air (*Phytophthora*), dan *scab*.

### **2.2.1 Antraknosa**

Penyakit *antraknosa* pada jambu biji disebabkan oleh cendawan *Collectrichum gleosporioides*, dimulai dengan munculnya bercak hitam pada buah muda dan berkembang ke seluruh permukaan buah sehingga buah berwarna hitam dan menjadi busuk. Buah jambu muda yang terserang menunjukkan gejala bercak-bercak secara cepat dan terjadi mumifikasi (Amusa *et al.*, 2006). Penyakit ini menyerang tunas muda, daun, dan buah pada tanaman. Gejalanya terlihat pada tunas muda dengan adanya sel tumbuhan yang mati (nekrosis) berwarna hitam.

Nekrosis ini juga menyebar ke ujung tunas, yang kemudian berubah menjadi coklat tua dan dapat merambat ke pangkal tunas, akhirnya menyebabkan kematian ujung.

Penyakit ini juga bisa dideteksi lewat daunnya yang mengalami kerutan dan mengembangkan jaringan mati pada tepi dan ujungnya yang berwarna coklat muda sampai hitam. Dalam kondisi serangan yang berat, daun-daun ini bahkan dapat gugur. Pada buah muda yang terinfeksi, gejalanya terlihat dalam bentuk bercak-bercak kecil seukuran kepala jarum, yang kemudian bersatu membentuk bulatan besar. Akibatnya, buah tersebut akan menjadi keras dan memiliki tekstur yang kasar. Penyakit ini seringkali pertama kali muncul pada buah yang sudah tua dan mulai matang. Penyakit antraknosa sering mengakibatkan bercak-bercak pada kulit luar buah, dan pada tingkat serangan yang parah, dapat menyebabkan buah gugur. Pengendalian penyakit ini dapat dilakukan dengan menjaga kebersihan sekitar kebun, termasuk lingkungan sekitar tanaman, serta mengurangi kelembaban melalui pemangkasan atau peningkatan jarak antar tanaman. Selain itu, untuk mengatasi penyakit ini, dapat digunakan fungisida alami seperti ekstrak kulit buah mahoni, sereh, dan lengkuas, atau fungisida sintetik yang dibuat oleh pabrik dengan bahan aktif tembaga (Cu), seperti Mankozeb, Propineb, dan Difenokonazo (Hanik et al, 2023).



**Gambar 2. 2** Penyakit Antraknosa

#### 2.2.2 Red Rust (*Karat Merah*)

Penyakit karat merah pada tanaman jambu biji disebabkan oleh alga *Chephaleuros virescens* yang tumbuh pada daun, bunga, buah, ranting, dan batang tanaman jambu kristal. Karat merah memiliki gejala berupa pertumbuhan talus yang menonjol pada permukaan atas daun, dengan filamen berwarna kuning atau merah

kecoklatan. Biasanya, daun yang terinfeksi terletak di pinggiran atau meluas hingga menutupi sebagian besar permukaan daun. Awalnya, bercak-bercak kecil muncul pada daun, lalu berkembang menjadi bercak yang lebih besar, kemudian bergabung dan terkadang menyebar ke seluruh bagian. (Hanik et al, 2023).

Pada tanaman jambu biji yang terinfeksi penyakit ini, akan terbentuk bercak-bintik berwarna coklat kemerahan berbentuk bulat. Dalam kasus karat merah, terlihat benang-benang yang menembus jaringan tanaman yang berdekatan, menyebabkan daun memiliki bercak yang menyerupai beludru. Pengendalian penyakit karat merah pada tanaman jambu biji dapat dilakukan dengan cara penyemprotan tembaga oksiklorida sebanyak 3 hingga 4 kali dengan jarak 15 hari sampai kondisi tanaman membaik. Selain itu, menjaga jarak tanam yang tidak terlalu rapat juga dapat membantu mengurangi penyebaran karat merah.



**Gambar 2. 3** Penyakit Red Rust

#### 2.2.3 *Scab (Bercak Buah)*

Penyakit scab pada tanaman jambu biji adalah penyakit yang disebabkan oleh jamur, biasanya jenis jamur *Venturia spp.* Penyakit ini adalah masalah umum yang dapat mempengaruhi tanaman jambu biji dan dapat menyebabkan kerusakan pada buah, daun, dan batang tanaman. Penyakit bercak buah ini biasanya ditandai oleh bercak-bercak atau bercak-bintik berwarna berbeda pada kulit buah. Bercak-bintik ini bisa berwarna coklat, hitam, atau bahkan berwarna lain tergantung pada penyebabnya. Bercak ini dapat membuat buah tidak menarik secara visual dan mengurangi kualitasnya.

Pengendalian bercak buah pada jambu biji melibatkan tindakan seperti pemangkasan buah yang terinfeksi guna membantu mencegah penyebaran infeksi

ke buah-buah yang sehat, dan penggunaan fungisida atau pestisida jika diperlukan, tergantung pada penyebabnya.



**Gambar 2. 4 Penyakit Scab**

#### 2.2.4 *Phytophthora* (*Jamur Air*)

*Phytophthora* adalah genus dari organisme eukariota uniselular yang termasuk dalam kelompok protista atau *Stramenopiles*. Organisme dalam genus *Phytophthora* dikenal sebagai *oomycetes* atau *air mold*. *Phytophthora* merupakan salah satu penyakit yang paling sering menyerang tanaman jambu biji. *Phytophthora* disebabkan oleh jamur air dan bermanifestasi sebagai bercak berwarna hitam keabu-abuan yang terletak pada permukaan buah jambu biji. Gejala dapat dikenali dengan munculnya miselium atau jamur air pertama kali pada pucuk buah dan tumbuh ke arah tangkai. Di kondisi lembab, seluruh permukaan buah dapat tertutup oleh pertumbuhan miselium dalam 2 hingga 3 hari. Akibatnya, kulit buah menjadi lunak dan dalam beberapa kasus, buah bahkan retak. Tangkai buah jambu biji yang terserang juga menjadi lunak dan akhirnya buah rontok.

Pengendalian *Phytophthora* pada tanaman jambu biji melibatkan praktik-praktik budidaya yang baik, seperti menjaga drainase yang baik, menghindari genangan air, dan mengurangi kelembaban di sekitar tanaman. Penggunaan fungisida atau bahan kimia yang sesuai juga dapat membantu mengendalikan penyakit ini agar jamur yang melekat pada tanaman yang satu tidak menyebar luas ke tanaman jambu biji yang lain.



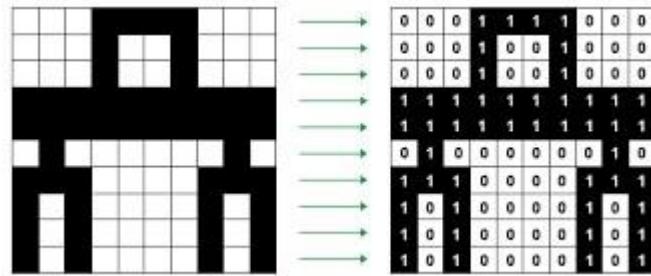
**Gambar 2. 5 Penyakit *Phytophthora***

### 2.3 Citra Digital

Citra digital dapat diartikan sebagai suatu fungsi dua dimensi  $f(x,y)$ , berukuran  $M$  baris dan  $N$  kolom sedangkan  $x$  dan  $y$  adalah posisi koordinat spasial dan amplitudo  $f$  di titik koordinat  $(x,y)$  yang dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut (Putra, 2013). Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari webcam). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer. Citra digital adalah representasi gambar yang dibuat oleh perangkat berdasarkan proses sampling dan kuantisasi. Proses sampling menentukan jumlah kotak yang tersusun dalam baris dan kolom dalam citra, yang pada dasarnya menggambarkan ukuran piksel (titik) pada citra. Sementara itu, kuantisasi mengacu pada kisaran nilai tingkat kecerahan yang diungkapkan sebagai tingkat keabuan (*grayscale*), sesuai dengan jumlah bit biner yang digunakan oleh perangkat. Dengan kata lain, kuantisasi citra menentukan seberapa banyak warna yang dapat direpresentasikan dalam citra (Kusumanto & Tompunu, 2011).

#### 2.3.1 Citra Biner

Citra biner merupakan citra yang setiap pikselnya memiliki 2 warna yaitu warna hitam atau warna putih, oleh karena hanya terdapat dua warna pada setiap piksel, maka hanya diperlukan 1 bit atau 8 bit per piksel, sehingga penyimpanan dapat dilakukan secara efisien (Laksono et al, 2022).

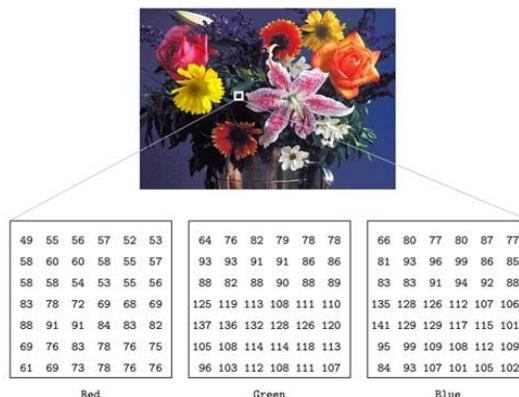


**Gambar 2. 6** Citra Biner (Kusumanto & Tompunu, 2011)

### 2.3.2 Citra Warna

Citra warna atau yang sering juga disebut dengan citra RGB merupakan jenis citra yang terdiri dari komponen warna merah (R), hijau (G) dan biru (B). Apabila warna-warna tersebut dikombinasikan, maka akan terbentuk warna citra asli. Oleh karena itu RGB sering disebut juga sebagai *additive color*.

Pengaturan warna RGB mempunyai skala 0-255 (Kusumanto & Tompunu, 2011). Warna dari tiap piksel ditentukan dengan kombinasi intensitas warna merah, biru dan hijau. Format file grafik menyimpan citra RGB sebagai 24-bit dengan komponen merah, hijau dan biru dengan masing-masing komponen 8 bit.



**Gambar 2. 7** Citra Warna RGB (Kusumanto & Tompunu, 2011)

### 2.4 Realtime Object Detection

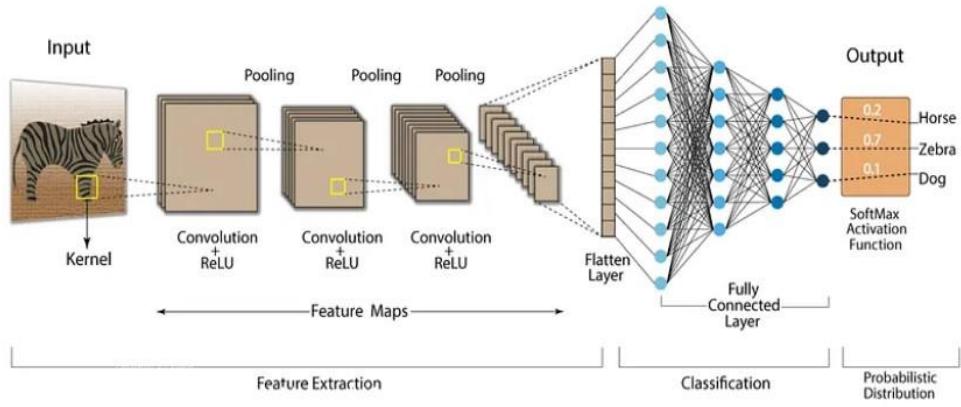
*Object Detection* atau deteksi objek merupakan sebuah teknologi komputer yang terkait dengan *computer vision* dan *image processing*, yang bertujuan untuk mengidentifikasi keberadaan suatu objek dalam gambar digital. Metode deteksi objek bertanggung jawab atas penemuan dan penandaan koordinat objek dalam gambar yang menjadi input. Pendekslsian objek bermanfaat dalam mengenali dan

mengidentifikasi objek dalam citra digital berdasarkan warna, bentuk, dan dataset yang digunakan.

*Real-time Object Detection* adalah proses deteksi objek yang dilakukan secara langsung menggunakan kamera atau ponsel pintar sebagai sumber input. Data yang diperoleh langsung dari perangkat tersebut kemudian diproses untuk klasifikasi. Kecepatan sangat diutamakan dalam *real-time Object Detection* karena objek harus terdeteksi secara instan, dengan frame per second (FPS) menjadi salah satu ukuran kinerja yang penting. Perkembangan teknologi perangkat keras juga berdampak pada kemampuan deteksi dan pelacakan objek secara *real-time*. Implementasi real-time memerlukan sistem pelacakan yang cepat, efisien energi, dan manajemen memori yang optimal agar dapat memenuhi persyaratan ketat real-time (Naeem et al, 2013).

## **2.5 Convolutional Neural Network (CNN)**

Convolutional neural network merupakan jenis dari jaringan syaraf buatan multilapisan yang didesain khusus untuk memproses data dua dimensi (McCann, Jin & Unser, 2017). *Convolutional Neural Network* dapat mengurangi jumlah parameter pada jaringan untuk meningkatkan algoritma backpropagation sebagai jaringan propagasi maju. *Convolutional Neural Network (CNN)* merupakan pendekatan komputasional yang mengikuti prinsip-prinsip jaringan syaraf tiruan, dirancang khusus untuk pengolahan citra dan pengenalan pola. CNN didasarkan pada konsep konvolusi, lapisan-lapisan yang menguraikan fitur-fitur penting dari citra dan mendalamkan pemahaman representasi citra. Pada algoritma *Convolutional Neural Network (CNN)*, area kecil yang juga disebut sebagai area local sensing diambil sebagai data input. Melalui propagasi maju, informasi melewati berbagai lapisan pada jaringan. Masing-masing lapisan terdiri dari kernel sehingga memperoleh beberapa fitur penting dari data yang diamati. Metode ini juga dapat memberikan bentuk bervariasi seperti stretching dan rotation (Nissa, N.F. et al 2021).



**Gambar 2.8** Arsitektur Umum CNN  
(Sumber: <https://nafizshahriar.medium.com>)

Struktur *Convolutional Neural Network (CNN)* terdiri dari beberapa tahap atau *layer* yang berbeda, tetapi ada beberapa *layer* utama yang umumnya ada dalam arsitektur CNN. Berikut adalah beberapa lapisan yang biasanya ada dalam CNN:

#### 1. Convolutional Layer

*Convolutional layer* merupakan lapisan yang bertanggung jawab untuk mengekstraksi fitur-fitur dari citra dengan menggunakan operasi konvolusi. Fitur-fitur ini dapat berupa tepi, tekstur, dan pola-pola lainnya.

#### 2. Pooling Layer

*Pooling layer* merupakan lapisan yang berfungsi untuk mengurangi dimensi data dengan mereduksi resolusi citra. Ini membantu mengurangi jumlah parameter yang perlu dipelajari oleh jaringan dan membuat representasi citra lebih tahan terhadap pergeseran dan perubahan kecil.

#### 3. Activation Layer

*Activation layer* merupakan lapisan yang berfungsi untuk aktivasi, seperti ReLU (*Rectified Linear Unit*), untuk memperkenalkan non-linearitas ke dalam jaringan. Ini memungkinkan jaringan untuk memahami hubungan yang lebih kompleks dalam data.

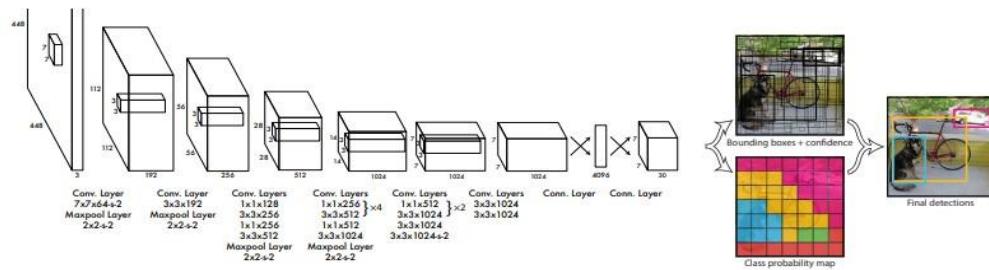
#### 4. Fully Connected Layer

*Fully connected layer* merupakan lapisan yang bertanggung jawab untuk menggabungkan fitur-fitur yang diekstraksi dari lapisan sebelumnya dan menghasilkan keluaran akhir. Lapisan ini sering digunakan dalam tugas klasifikasi dan dapat terdiri dari beberapa neuron yang menghasilkan prediksi.

Selain lapisan-lapisan di atas, CNN juga dapat memiliki lapisan tambahan seperti lapisan normalisasi, lapisan dropout, dan sebagainya, tergantung pada arsitektur spesifik dan tugas yang dihadapi. Jumlah lapisan dan konfigurasi CNN dapat bervariasi tergantung pada tujuan dan kompleksitas model yang dibutuhkan.

## 2.6 Algoritma *You Only Look Once* (YOLO)

Algoritma *You Only Look Once* (YOLO) adalah sebuah metode yang digunakan dalam pengolahan citra dan pengenalan objek dalam visi komputer. Algoritma YOLO menerapkan neural network pada sebuah gambar dan membagi input gambar menjadi sejumlah grid, lalu memprediksi *bounding box* serta probabilitas untuk masing-masing grid. YOLO bekerja dengan membagi citra menjadi grid yang lebih kecil dan kemudian memprediksi kotak pembatas dan kelas objek dalam setiap grid. Pendekatan ini memungkinkan YOLO untuk mendeteksi objek dalam satu proses pengolahan, menghasilkan deteksi yang cepat dan akurat.



**Gambar 2.9** Arsitektur Umum YOLO  
(Sumber: <https://rubikscode.net>)

Berikut adalah versi YOLO yang pada umumnya digunakan dalam pendekripsi objek:

### 2.6.1 YOLO v5

Pada tanggal 9 Juni 2020, Glenn Jocher, seorang peneliti dan CEO *Ultralytics* mengumumkan perilisan YOLOv5. Namun, pengumuman ini tidak disertai dengan penulisan formal, melainkan hanya merilis implementasi sumber terbuka di GitHub. Ultralytics adalah sebuah perusahaan yang mengonversi versi YOLO ke dalam kerangka kerja *PyTorch* yang ditulis dalam bahasa *Python*. YOLOv5 hadir dalam empat versi utama: nano (n), kecil (s), sedang (M), dan besar (L). Setiap varian memiliki durasi pelatihan yang berbeda.

Struktur arsitektur YOLOv5 menunjukkan kesamaan yang mencolok dengan arsitektur YOLO versi sebelumnya. Seperti pendahulunya, YOLOv5 juga menerapkan pendekripsi tahap tunggal yang terdiri dari tiga komponen utama: *backbone*, *neck*, dan *head*. YOLOv5 menggunakan jaringan saraf yang sama dengan versi sebelumnya, yaitu *CSPDarknet53* untuk *backbone*, PANet untuk *neck*, dan versi *head* yang serupa dengan YOLO versi sebelumnya.

Rancangan YOLOv5 menggunakan bahasa pemrograman *Python* dan *framework PyTorch*. Keputusan ini memungkinkan fleksibilitas yang lebih besar dalam penggunaan YOLOv5, mulai dari model kecil yang mampu memberikan kecepatan frame per detik (FPS) *real-time* pada perangkat edge hingga model besar dan akurat yang dapat diimplementasikan pada GPU *cloud*. Integrasi dengan *PyTorch* membuat modifikasi terhadap arsitektur YOLOv5 lebih mudah dilakukan oleh pengembang dan dapat diterapkan secara langsung.

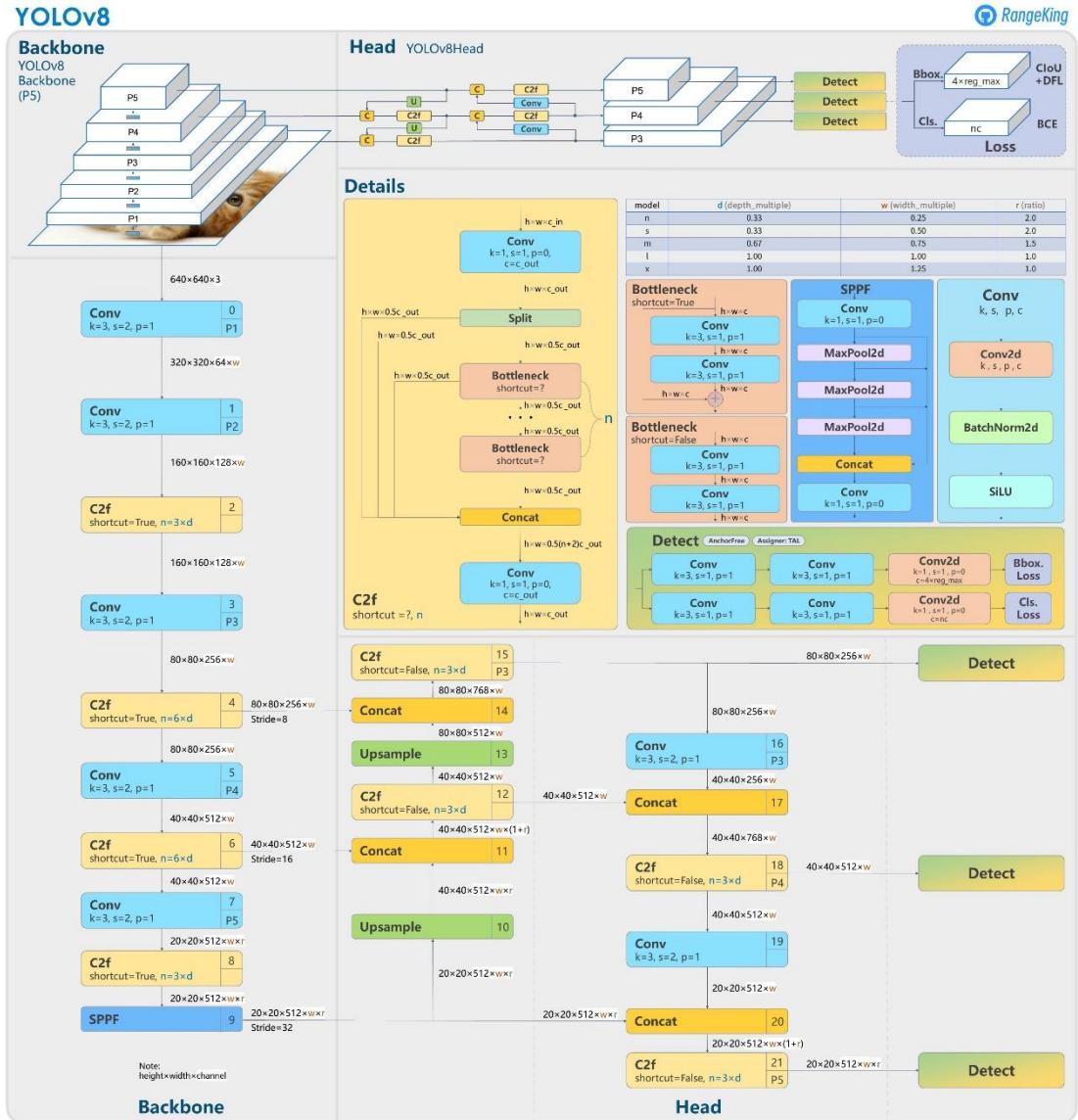
### 2.6.2 YOLO v7

Pada tahun 2022 detektor objek YOLOv7 menciptakan perubahan signifikan dalam pasar visi komputer dengan fitur-fiturnya yang mengesankan. YOLOv7 secara resmi menampilkan peningkatan kecepatan dan akurasi yang luar biasa dibandingkan dengan versi sebelumnya. Berbeda dengan pendekatan sebelumnya yang menggunakan model yang telah dilatih sebelumnya, YOLOv7 melatih modelnya menggunakan kumpulan data COCO *Microsoft*. Salah satu fokus utama dari arsitektur YOLOv7 adalah penggunaan *Extended-Efficient Layer Aggregation Network* (EELAN) sebagai *backbone*, yang mengutamakan efisiensi komputasi dan pemodelan karakteristik arsitektur. Dengan pengaturan rute gradien, EELAN memberikan manfaat utama berupa peningkatan kemampuan pembelajaran dan konvergensi jaringan yang lebih dalam. YOLOv7 juga memperkenalkan model gabungan penskalaan untuk model berbasis penggabungan, yang memungkinkan pelestarian atribut awal model dan meningkatkan struktur secara efektif. Model tersebut kemudian difokuskan pada modul yang dapat dilatih dan teknik optimasi yang dikenal sebagai "*Bag-of-Freebies*" (BoF), strategi yang meningkatkan kinerja model tanpa biaya tambahan pelatihan. Meskipun masih dalam tahap pengembangan, YOLOv7 menjanjikan potensi besar dalam menangani berbagai tantangan visi komputer. Dengan terus mengembangkan solusi untuk kesulitan

yang dihadapi oleh para pengembang, algoritma ini diharapkan akan menjadi alat yang sangat berharga dalam menyelesaikan berbagai masalah komputasi visual di masa mendatang.

#### 2.6.3 YOLO v8

YOLOv8 adalah model YOLO terbaru yang diperkenalkan oleh Glenn Jocher pada Januari 2023 yang lalu yang dapat digunakan untuk tugas deteksi objek, klasifikasi gambar, dan segmentasi instan. YOLOv8 masih menggunakan arsitektur dasar yang sama dengan YOLOv5, yakni *CSPDarknet53*. *CSPDarknet53* adalah varian arsitektur model dari *Darknet53* yang telah ditingkatkan dalam hal akurasi dan efisiensi dengan penerapan metode *CSPNet* (*Cross Stage Partial Network*). *CSPNet* sendiri adalah teknik yang memisahkan jaringan menjadi dua jalur paralel pada setiap tahapnya. Dimana jalur pertama menjalankan proses konvolusi lengkap (*full convolution*), sementara jalur kedua melakukan proses konvolusi parsial (*partial convolution*). YOLOv8 dioptimalkan dengan menggunakan pengoptimalan ADAM (*Adaptive Moment Estimation*) dalam proses *model training*. Dimana pengoptimalan ADAM dikenal memiliki kemampuan untuk membuat *learning rate* yang lebih adaptif dan dapat meningkatkan efisiensi kinerja algoritma.



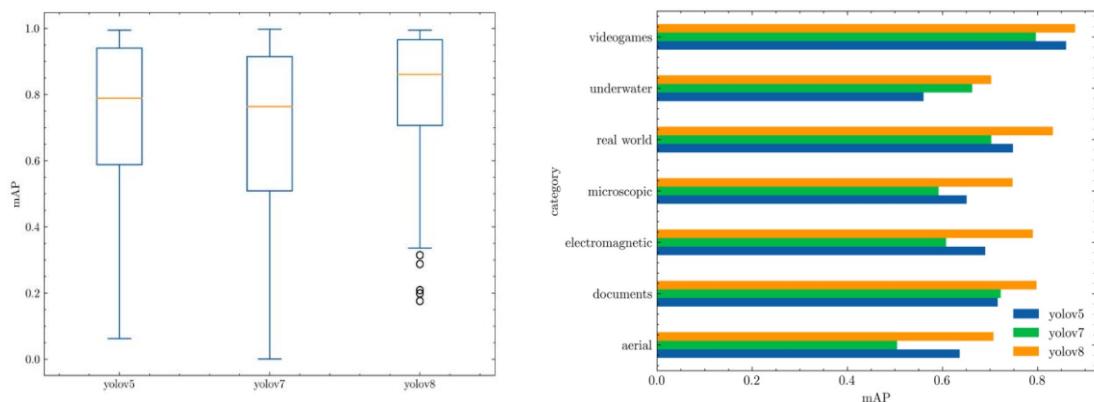
**Gambar 2. 10** Arsitektur YOLO v8 (Solawetz, J., & Francesco, 2023)

## 2.7 Perbandingan YOLOv5, YOLOv7, YOLOv8

Dibandingkan dengan model-model utama sebelumnya dalam seri YOLO, seperti YOLOv5 dan YOLOv7, YOLOv8 menonjol sebagai model yang lebih canggih dan up-to-date, menawarkan tingkat akurasi dan kecepatan deteksi yang lebih tinggi. Penelitian YOLOv8 didasarkan pada evaluasi empiris yang kuat menggunakan *Common Objects in Context* (COCO) sebagai standar evaluasi utama. Ketika setiap aspek jaringan dan rutinitas pelatihan mengalami perubahan, serangkaian eksperimen baru dijalankan untuk mengonfirmasi dampak perubahan tersebut dalam pemodelan COCO.

COCO dikenal sebagai tolak ukur industri yang digunakan untuk mengevaluasi kinerja model deteksi objek. Saat membandingkan berbagai model pada dataset COCO, *roboflow* memperhatikan nilai mean *Average Precision* (mAP) dan mengukur frame per detik (FPS) untuk mengukur kecepatan inferensi. Perbandingan model-model harus dilakukan pada tingkat kecepatan inferensi yang sebanding (Solawetz, J., & Francesco, 2023).

Roboflow telah menggunakan 100 sampel dari berbagai dataset yang tersedia di Roboflow Universe, yang merupakan repositori dengan lebih dari 100.000 dataset, untuk mengevaluasi kemampuan model dalam konteks generalisasi ke domain yang baru. Evaluasi model YOLOv8 dilakukan sejalan dengan model-model YOLOv5 dan YOLOv7 pada benchmark RF100. Grafik yang disajikan menunjukkan bahwa YOLOv8 memiliki jumlah outlier yang lebih sedikit dan kinerja keseluruhan yang lebih baik jika diukur dengan standar benchmark Roboflow 100.



**Gambar 2. 11** Rata-rata YOLO mAP@.50 dibandingkan kategori RF100  
(Solawetz, J., & Francesco, 2023)

## 2.8 Confusion Matrix

*Confusion matrix* merupakan sebuah metode yang digunakan untuk mengevaluasi kinerja model atau algoritma dalam memprediksi kategori atau kelas data yang diberikan. Kinerja klasifikasi pada pembelajaran mesin dapat diukur dengan menggunakan *confusion matrix* saat *output* memiliki dua atau lebih kelas. *Confusion matrix* yang dihasilkan oleh pengklasifikasi pada set validasi dapat

digunakan untuk mengidentifikasi kelas yang menyebabkan kebingungan dalam klasifikasi, sehingga struktur klasifikasi yang lebih spesifik dapat dikembangkan (Cavalin & Oliveira, 2019).

**Tabel 2. 2 Confusion Matrix**

		<b>Prediksi</b>	
		<b>Positif</b>	<b>Negatif</b>
<b>Aktual</b>	<b>Positif</b>	TP	FN
	<b>Negatif</b>	FP	TN

Dari tabel diatas, terdapat empat nilai yang dibentuk dari *confusion matrix*, yaitu:

- *True Positive* (TP): Berarti data memiliki nilai positif dengan hasil prediksi positif juga.
- *False Negative* (FN): Berarti data memiliki nilai positif, namun hasil prediksi negatif.
- *False Positive* (FP): Berarti data memiliki nilai negatif, namun hasil prediksi positif.
- *True Negative* (TN): berarti data memiliki nilai negatif dengan hasil prediksi negatif juga.

Dengan menggunakan empat istilah tersebut, *confusion matrix* dapat digunakan dalam menghitung berbagai *performance matrix*, seperti:

### 2.8.1 Accuracy

*Accuracy* merupakan persentase kebenaran dari nilai prediksi dibanding dengan nilai sebenarnya dari sebuah data. Nilai akurasi yang semakin tinggi menghasilkan model yang semakin akurat sehingga dapat mengklasifikasi data dengan benar.

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

### 2.8.2 *Precision*

*Precision* adalah proporsi hasil yang diprediksi positif dengan hasil yang diprediksi positif secara keseluruhan.

$$\text{Precision} = \frac{TP}{TP + FP}$$

(2)

### 2.8.3 *Recall*

*Recall* adalah persentase keberhasilan model dalam memperoleh informasi yang diminta. Oleh karena itu, *recall* adalah proporsi data dengan kategori positif yang dengan tepat diprediksi positif.

$$\text{Recall} = \frac{TP}{TP + FN}$$

(3)

### 2.8.4 *F1-Score*

*F1-Score* merupakan fungsi dari *precision* dan *recall* yang merupakan persentase tingkat keseimbangan antara *precision* dan *recall*. *F1-Score* bisa juga disebut sebagai perbandingan rata-rata dari *precision* dan *recall*.

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

(4)

## 2.9 Penelitian Terdahulu

Terdapat beberapa penelitian mengenai deteksi penyakit daun jambu biji yang sebelumnya sudah pernah dilakukan, diantaranya adalah penelitian oleh (Howlader et al, 2019) yang berjudul “Automatic Recognition of Guava Leaf Diseases using Deep Convolution Neural Network”. Penelitian ini dilakukan dengan dilakukan pendekatan berbasis *Deep Convolution Neural Network* (D-CNN) yang sesuai untuk mengidentifikasi penyakit daun jambu biji secara otomatis. Model yang diusulkan berlaku untuk mengklasifikasikan penyakit utama pada daun jambu biji seperti Bercak Daun Alga, Kutu Kebul, dan Karat. Berdasarkan pernyataan pada jurnal terkait, ini adalah pertama kalinya

menggunakan model berbasis D-CNN untuk mengenali penyakit daun jambu biji. Selain itu, peneliti membuat dataset sendiri yaitu BU\_Guava\_Leaf (BUGL2018) dengan empat kategori berbeda. Pada kumpulan data ini, peneliti melatih pendekatan identifikasi penyakit yang diusulkan dan mengevaluasi hasil eksperimen yang menunjukkan akurasi rata-rata 98,74% pada set pengujian.

Penelitian lainnya tentang deteksi penyakit daun jambu biji pernah dilakukan oleh (Thangaraj, R. et al, 2023) dengan judul “*A Comparative Study of Deep Learning Models for Guava Leaf Disease Detection*”. Penelitian dini dimulai dengan proses mengkategorikan penyakit utama daun jambu biji antara lain *Canker*, *Dot*, *Healthy*, *Mummification* dan *Rust* menggunakan model yang disajikan. Data yang digunakan adalah gambar dari daun jambu biji yang dikumpulkan dari dataset kaggle. Dalam pemrosesan data, model CNN seperti *DenseNet121*, *DenseNet169*, *InceptionV3*, dan *Xception* digunakan untuk mengkategorikan penyakit yang menyerang daun jambu biji. Dalam penelitian ini dibuktikan bahwa model DenseNet169 mengungguli model lainnya dengan akurasi 96,12%. Ketika efektivitas model dinilai menggunakan metrik kinerja seperti presisi, perolehan, dan skor F1, hasilnya menunjukkan bahwa *DenseNet169* memberikan hasil yang sangat baik.

Penelitian tentang deteksi penyakit pada daun jambu lainnya juga pernah dilakukan oleh (Ghone, U. et al, 2021) dengan judul “*Deteksi Penyakit Daun Pada Citra Daun Jambu Biji Menggunakan Segmentasi Warna*”. Penelitian ini menggunakan data gambar dari website google dengan ukuran dimensi dan pencahayaan yang berbeda. Pada penelitian ini digunakan variabel berupa dimensi citra daun jambu biji. Dilakukan proses pengaturan variabel mulai dari nilai variabel kecil hingga ke variabel yang besar untuk mengetahui pengaruh variabel dimensi citra daun jambu biji terhadap kemampuan mengidentifikasi daerah daun yang terkena penyakit. Kemudian, dilakukan proses *pre-processing* yang berupa pengambilan nilai elemen A, dan B pada ruang warna cielab. Kedua elemen ini digunakan sebagai input pada algoritma *k-means*. Algoritma *k-means* memetakan setiap titik citra daun berdasarkan komponen ruang warna A, B dari ruang warna cielab tanpa memasukkan komponen yang lainnya yang menunjukkan *lightness*. Peneliti menarik kesimpulan bahwa ruang warna cielab handal digunakan sebagai

input dari algoritma *k-means* untuk mengenali area daun jambu biji yang terserang penyakit dalam pencahayaan yang berbeda. Algoritma *k-means* handal dalam mengenali warna pada citra daun jambu biji yang terserang penyakit meskipun citra sudah dilakukan reduksi ukuran dimensi citra.

Terdapat sebuah penelitian yang dilakukan oleh (Md. Janibul et al, 2023) dengan judul “*Tea leaf disease detection and identification based on YOLOv7 (YOLO-T)*”. Penelitian ini menggunakan 4000 gambar digital yang mewakili lima jenis penyakit daun yaitu *Tea Rust* (*Hemileia vastatrix*), *Tea Blight* (*Exobasidium vexans*), *Tea Anthracnose* (*Colletotrichum gloeosporioides*), *Tea Wither Tip* (*Fusarium spp.*), dan *Root Rot* (*Phytophthora spp.*). Gambar-gambar ini dikumpulkan dari empat kebun teh terkemuka di Bangladesh dan kemudian dibuat kumpulan data gambar penyakit daun yang diberi anotasi secara manual. Untuk mengatasi masalah ukuran sampel yang tidak mencukupi, penelitian ini menggunakan pendekatan augmentasi data. Hasil deteksi dan identifikasi dengan menggunakan YOLOv7 divalidasi dengan metrik statistik terkemuka seperti akurasi deteksi, presisi, perolehan, nilai mAP, dan skor F1. Hasil validasi tersebut menunjukkan angka-angka yang sangat baik, yaitu 97,3% untuk akurasi deteksi, 96,7% untuk presisi, 96,4% untuk perolehan, 98,2% untuk nilai mAP, dan 0,965 untuk skor F1. Berdasarkan hasil percobaan, YOLOv7 terbukti lebih unggul dalam mendeteksi dan mengidentifikasi penyakit daun teh pada gambar alam jika dibandingkan dengan berbagai jaringan deteksi dan identifikasi target yang sudah ada, seperti CNN, Deep CNN, DNN, AX-Retina Net, peningkatan DCNN, YOLOv5, dan segmentasi gambar multi-objektif.

Selanjutnya adalah penelitian yang dilakukan oleh (Qadri, S. A. et al, 2023) dengan judul “*Plant Disease Detection and Segmentation using End-to-End YOLOv8: A Comprehensive Approach*”. Penelitian ini menggunakan versi terbaru dari algoritma YOLO, yang memiliki arsitektur yang rumit dengan berbagai lapisan konvolusional. Arsitektur ini memungkinkan ekstraksi fitur yang kompleks dari gambar, sehingga dapat mengidentifikasi penyakit daun tanaman dengan akurat. Model ini dilatih secara menyeluruh (end-to-end), memungkinkan model untuk efektif mempelajari dan menggeneralisasi data masukan, yang pada gilirannya meningkatkan kinerja prediktifnya untuk mendeteksi penyakit daun

yang belum pernah terjadi sebelumnya atau yang baru. Hasil evaluasi untuk pendekatan YOLOv8 divalidasi dengan menggunakan metrik statistik terkemuka, termasuk presisi, perolehan, nilai mAP50, mAP50-95, dan skor F1, yang menghasilkan 99,8%, 99,3%, 99,5%, 96,5%, dan 0,999 untuk *bounding box* dan 99,1%, 99,3%, 99,3%, 98,5% dan 0,992 untuk masing-masing *segmentation mask*.

Penelitian lain yang menggunakan algoritma YOLO juga dilakukan oleh (Agbulos, K. et al, 2021) dengan judul “*Identification of Leaf Blast and Brown Spot Diseases on Rice Leaf with YOLO Algorithm*”. Dalam penelitian ini, digunakan algoritma YOLO yang telah dilatih dengan dataset yang terdiri dari 200 gambar daun padi. Daun padi diklasifikasikan ke dalam tiga kelas yang berbeda. Hasil penelitian menunjukkan bahwa akurasi deteksi penyakit leaf blast pada kelas pertama mencapai 90,00%, sementara akurasi deteksi penyakit brown spot pada kelas kedua sebesar 70,00%. Untuk kelas ketiga, yaitu penyakit yang tidak dapat diidentifikasi, perangkat mencapai akurasi 100,00%. Dengan demikian, dapat disimpulkan bahwa akurasi keseluruhan perangkat adalah sebesar 73,33%, sementara tingkat kesalahan komisi hanya mencapai 26,67%.

**Tabel 2. 3** Penelitian Terdahulu

No.	Penulis	Tahun	Metode	Keterangan
1.	Howlader, M. R., Habiba, U., Faisal, R. H., & Rahman, M.	2019	<i>Deep Convolutional Neural Network</i>	Penelitian ini membahas tentang pengenalan otomatis daun jambu biji. Akurasi rata-rata yang dihasilkan dari penelitian ini adalah 98,74%.
2.	Thangaraj, R., Mohan, M., et al	2023	<i>Convolutional Neural Network</i>	Penelitian ini membahas tentang pendekripsi penyakit jambu biji yang dapat dilihat dari daunnya. Penelitian ini mendapatkan akurasi rata-rata yang cukup tinggi sebesar 96,12%.

<b>3.</b>	Ghoni, U., Fatkhunnajah, E., et al	2021	<i>kmeans</i>	Penelitian ini membahas tentang deteksi penyakit daun jambu biji. Penelitian ini menyatakan bahwa algoritma kmeans cukup efektif dalam mengenali warna citra daun walaupun dilakukan di dimensi dengan ukuran yang berbeda.
<b>4.</b>	Md. Janibul Alam Soeb, Md. Fahad Jubayer, et al	2023	<i>YOLOv7</i>	Penelitian ini membahas tentang identifikasi penyakit yang terletak pada daun teh menggunakan model deteksi objek satu tahap <i>YOLOv7</i> . Penelitian ini memiliki nilai akurasi deteksi sebesar 97,3%.
<b>5.</b>	Qadri, S. A., Huang, N.-F., Wani, T. M., & Bhat, S. A.	2023	<i>YOLOv8</i>	Penelitian ini membahas tentang pendekripsi dan segmentasi penyakit tanaman. Penelitian ini menyoroti efektivitas model <i>YOLOv8</i> dalam deteksi penyakit tanaman dengan nilai akurasi presisi sebesar 99,8%.
<b>6.</b>	Agbulos, K., Sarmiento, Y., Villaverde, J.	2021	<i>YOLOv5</i>	Penelitian ini membahas tentang identifikasi penyakit ledakan daun padi dan penyakit bercak coklat pada daun tanaman padi. Penelitian ini mendapatkan akurasi keseluruhan sebesar 73,33%.

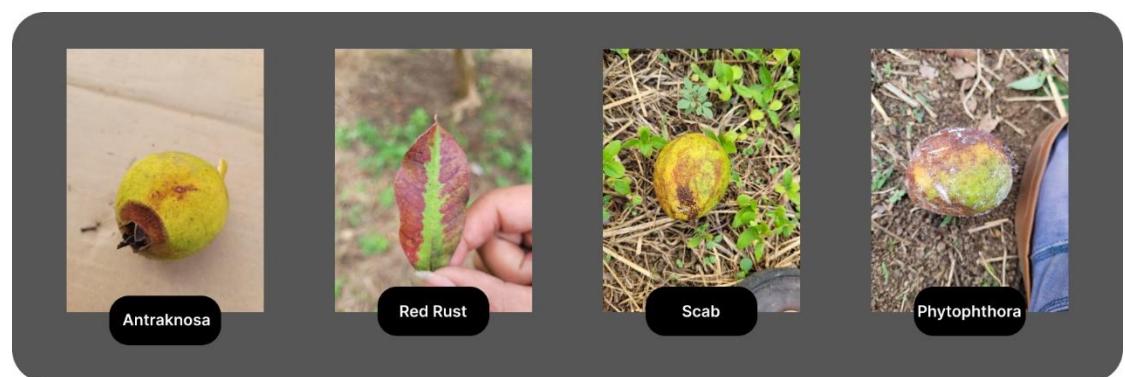
Adapun perbedaan penelitian ini dengan penelitian sebelumnya seperti (Howlader et al, 2019) adalah penelitian tersebut mengidentifikasi penyakit pada daun jambu biji secara otomatis dengan menggunakan pendekatan berbasis *Deep Convolutional Neural Network* (D-CNN). Kemudian penelitian yang dilakukan oleh (Qadri, S. A. et al, 2023) meneliti penyakit pada daun tanaman secara general menggunakan proses segmentasi dari hasil deteksi jenis penyakit yang umum terjadi pada daun tanaman dengan pendekatan YOLOv8. Sedangkan penelitian ini adalah deteksi jenis penyakit tanaman jambu biji menggunakan YOLO V-8. Sistem ini akan mendeteksi jenis penyakit yang ada pada tanaman jambu biji lewat citra buah dan juga citra daun yang dapat diakses menggunakan *android smartphone*.

## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1 Dataset

Data input yang digunakan berupa citra buah jambu biji yang terserang penyakit yang didapat dari lahan pertanian jambu biji milik seorang petani di Kabupaten Deli Serdang - Tanjung Anom, Sumatera Utara. Pengumpulan data diambil langsung menggunakan kamera *smartphone*. Seluruh data yang dikumpulkan berekstensi JPG dengan jumlah total data sebanyak 1926 citra. Contoh data citra jambu biji yang digunakan pada penelitian ini dapat dilihat pada Gambar 3.1.



**Gambar 3. 1** Contoh Data Citra Buah Jambu Biji yang Terserang Penyakit

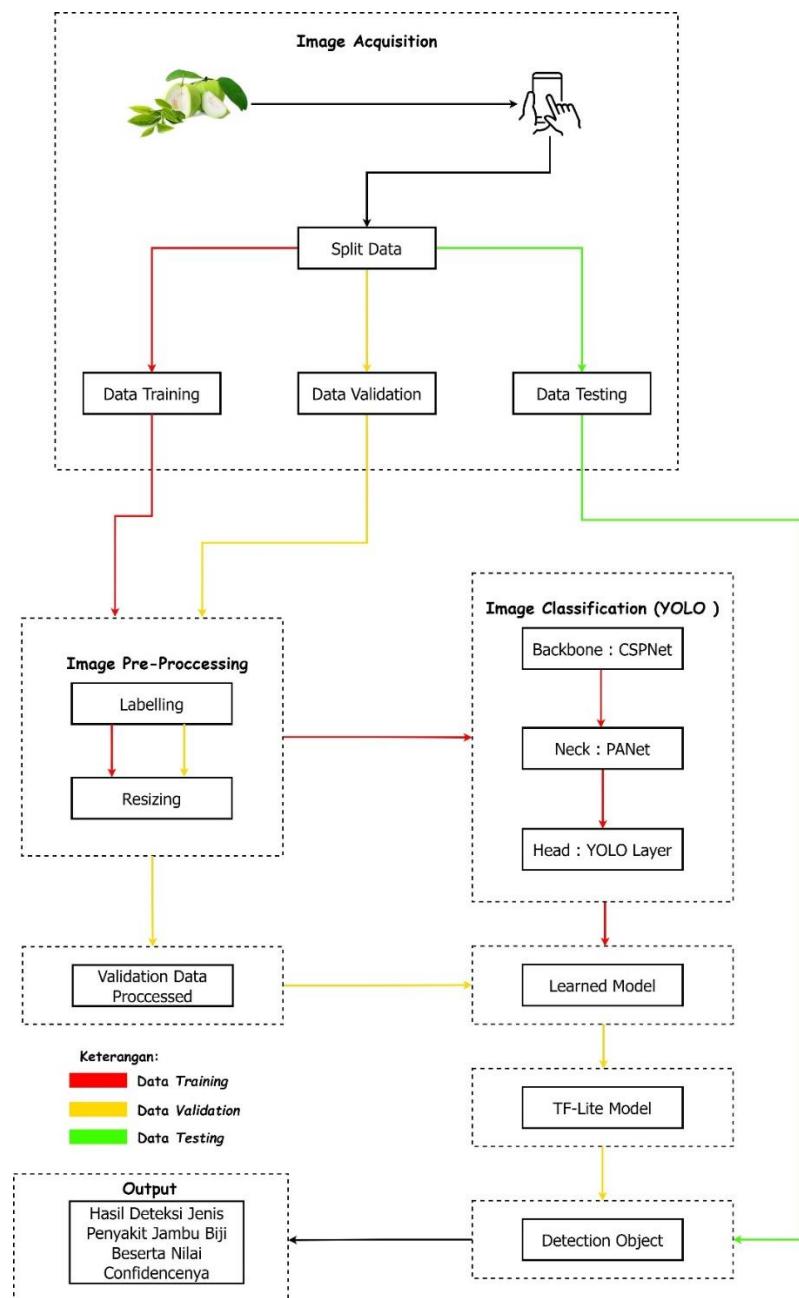
Dalam penelitian ini, digunakan empat jenis penyakit tanaman jambu biji, yaitu *Antraknosa*, *Red Rust*, *Scab*, dan *Phytophthora*. Seluruhnya terdiri dari 1926 data gambar. Data tersebut kemudian dibagi menggunakan rasio 80/10/10 yang merupakan rasio data split dengan performa yang terbaik (Golchubian et al, 2021), menjadi 1542 data untuk keperluan pelatihan, 192 data untuk data validasi dan 192 data untuk pengujian.

**Tabel 3. 1** Split Data

	<b>Train</b>	<b>Validation</b>	<b>Test</b>	<b>Total</b>
<b><i>Antraknosa</i></b>	385	48	48	481
<b><i>Phytophthora</i></b>	384	48	48	480
<b><i>Red Rust</i></b>	387	48	48	483
<b><i>Scab</i></b>	386	48	48	482
<b><i>TOTAL</i></b>	1542	192	192	1926

### 3.2 Analisis Sistem

Penelitian ini mengusulkan suatu metode yang digunakan untuk mengidentifikasi penyakit tanaman jambu biji berdasarkan citra buah maupun daun tanaman. Gambar 3.2 merupakan alur yang merepresentasikan proses yang akan dilakukan pada penelitian ini dimana hasil akhirnya diharapkan mendapat akurasi yang baik dalam mendeteksi jenis penyakit tanaman jambu air berdasarkan citra daun maupun buahnya.



Gambar 3. 2 Arsitektur Umum

Berikut merupakan penjelasan mengenai proses arsitektur pada gambar 3.2

### 3.2.1 *Image Acquisition*

Tahap ini merupakan tahap awal yaitu tahap memasukkan data sistem dalam pengumpulan data citra yang digunakan. Penelitian ini menggunakan data citra daun dan buah tanaman jambu biji yang diambil dari dua sumber yaitu dataset yang berasal dari *kaggle* dan data yang diambil dengan kamera *smartphone*. Kemudian data citra dibagi ke dalam tiga bagian, yaitu *data training*, *data validation* dan *data testing*.

### 3.2.2 *Pre-Processing*

Tahapan ini merupakan proses pengolahan awal citra daun dan buah jambu biji agar mendapatkan hasil yang lebih baik untuk diproses ke tahap selanjutnya. Adapun tahapan pada *pre-processing* yang akan dilakukan pada penelitian ini adalah *labelling*, *auto-orient*, dan *resizing*.

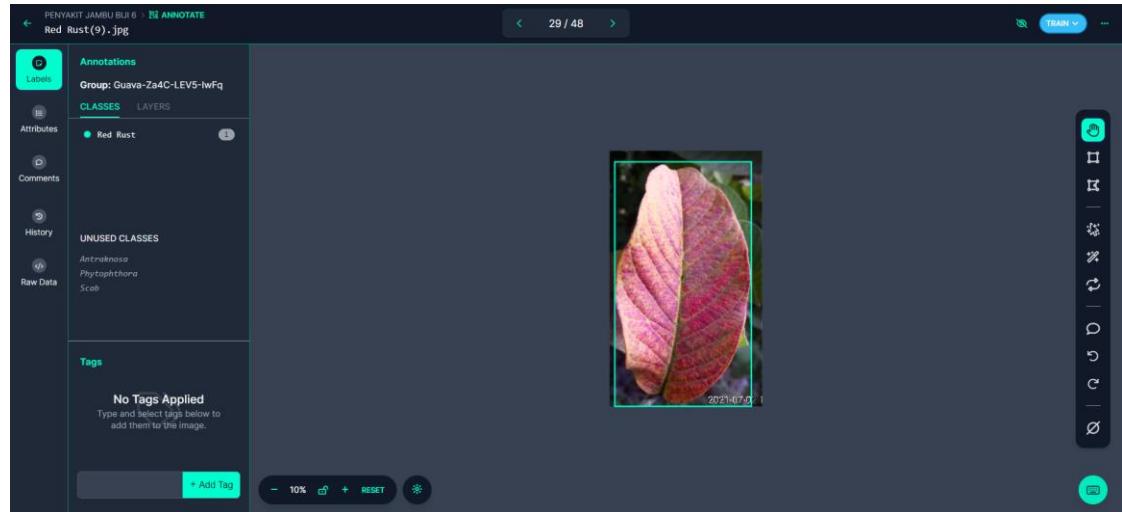
#### 1. *Labelling*

*Labelling* merupakan tahapan proses untuk memberikan label atau nama pada data citra yang akan digunakan untuk pengujian sistem. Proses *labelling* ini dilakukan agar dapat ditandai kelas citra tersebut sehingga dapat dibedakan jenis penyakit tanaman jambu biji yang akan digunakan.



**Gambar 3. 3** Data yang Diberi Label

Data citra akan diberi label dengan menganotasi pada setiap citra, terdiri dari empat kelas yang sudah ditetapkan, yaitu *Antraknosa*, *Red Rust*, *Scab*, *Phytophthora*. Setelah proses pelabelan, hasilnya akan disimpan dalam bentuk folder yang nantinya digunakan dalam tahap pelatihan data.



**Gambar 3. 4** Pemberian Label dan Bounding Box Pada Citra

Citra yang telah diberi label akan tersimpan dalam format text. File ini berisi kelas dari objek tersebut, nilai *center point x* label, nilai *center point y* label, lebar label, dan tinggi label. Contoh format teks hasil pelabelan citra terlihat seperti gambar 3.5.

```
Red-Rust-9-.jpg.rf.911f8e7a4c71002c362f6ea...
File Edit Format View Help
2 0.93125 0.04218750000000006 0.034375 0.0421875
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

The screenshot shows a text editor window with the title 'Red-Rust-9-.jpg.rf.911f8e7a4c71002c362f6ea...'. The content of the file is displayed in the main area, showing a single line of text: '2 0.93125 0.04218750000000006 0.034375 0.0421875'. At the bottom of the editor, the status bar shows 'Ln 1, Col 1', '100%', 'Windows (CRLF)', and 'UTF-8'.

**Gambar 3. 5** Isi File Txt Yang Diberi Label

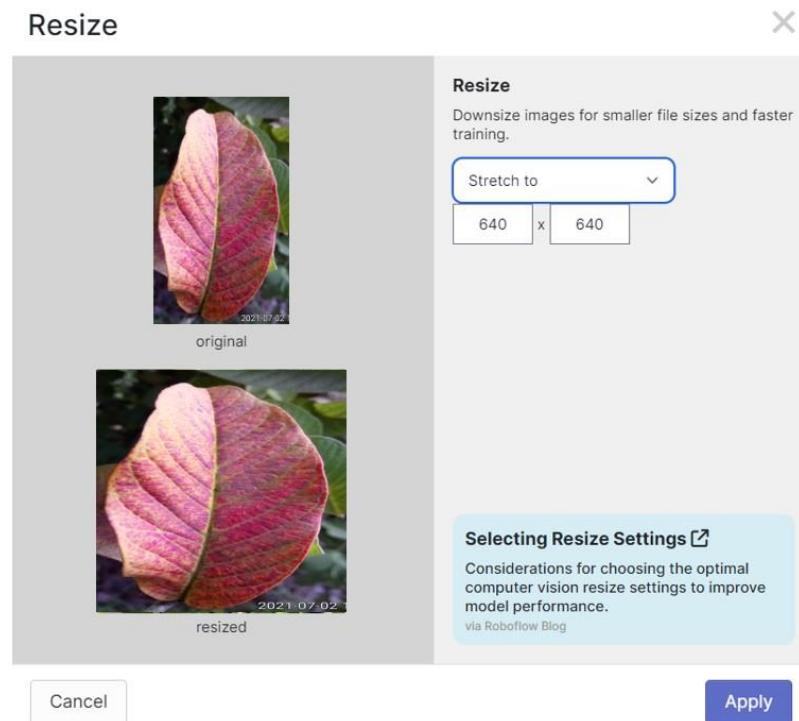
## 2. Resizing

*Resizing* pada *pre-processing* adalah proses mengubah ukuran dimensi gambar menjadi ukuran yang diinginkan sebelum data tersebut dimasukkan ke dalam model atau algoritma pengenalan objek. Pada penelitian ini, data citra memiliki ukuran dimensi sebesar 2106 x 3520. Oleh karena itu, untuk mengurangi jumlah piksel yang perlu diproses oleh model, mengurangi beban komputasi, dan mempercepat waktu inferensi, maka perlu dilakukan proses *resize* gambar ke ukuran dimensi yang lebih kecil.



**Gambar 3. 6** Contoh Data Sebelum dan Setelah *Resizing*

Proses *resize* yang digunakan pada penelitian ini adalah jenis *resize stretch to*. Ukuran *input* gambar model *default* yang digunakan untuk melatih model YOLOv8 adalah 640. Dikarenakan metode yang digunakan untuk penelitian ini adalah YOLOv8, maka ukuran *resize* yang ditetapkan untuk penelitian ini adalah 640x640.



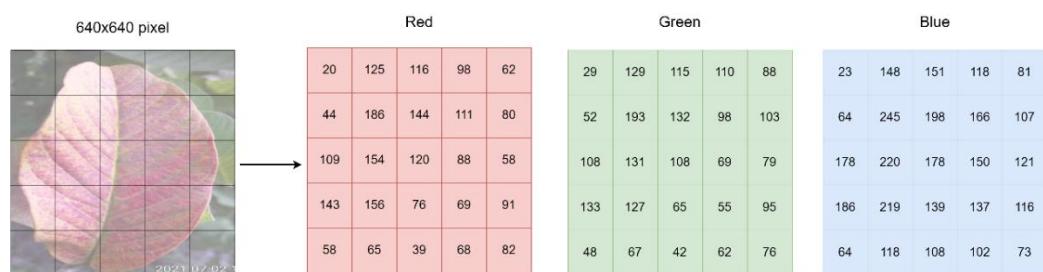
**Gambar 3. 7** Penerapan *Resize* Pada Platform *Roboflow*

### 3.2.3 Image Classification

Pada tahap ini citra daun dan buah jambu biji akan diklasifikasi pada kelasnya menggunakan algoritma YOLO versi 8. YOLOv8 merupakan salah satu algoritma yang mampu mendeteksi objek berdasarkan kemungkinan kelas yang ditetapkan. Adapun proses yang ada pada tahapan ini adalah:

#### 1. Backbone: CSPDarknet53

*Backbone network* merupakan tahapan yang mengacu pada *feature extraction*. Pada tahapan ini *Backbone network* yang digunakan untuk mengekstraksi fitur adalah *CSPDarknet53* yang telah dimodifikasi dan ditingkatkan dengan nama *C2f* (*cross-stage partial bottleneck with two convolutions*) yang terinspirasi oleh modul ELAN. Modul *C2f* ini mengintegrasikan dua cabang aliran gradien paralel, memfasilitasi aliran informasi gradien yang lebih kuat untuk meningkatkan akurasi deteksi. Modul *C2f* menggantikan *CSPLayer* yang digunakan dalam YOLOv5 (Terven & Esparaza, 2023). Gambar yang digunakan pada penelitian ini adalah gambar dengan dimensi 640x640x3, dengan kata lain masing-masing gambar memiliki nilai *red*, *green*, dan *blue*. Contoh nilai warna pada gambar yang akan digunakan dapat dilihat pada gambar 3.8 berikut.



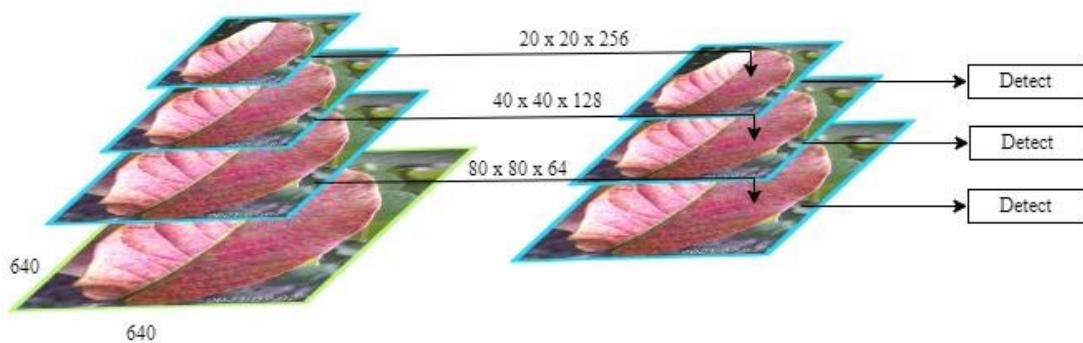
**Gambar 3.8** Contoh Nilai RGB Pada Citra

#### 2. Neck: PAN-FPN

Terinspirasi oleh *PANet*, YOLOv8 dirancang dengan struktur PAN-FPN (*Path Aggregation Network with Feature Pyramid Network*) di bagian *Neck*. PAN-FPN membangun struktur jaringan *top-down* dan *bottom-up* guna saling melengkapi informasi posisi dangkal dan informasi semantik yang mendalam

melalui perpaduan fitur, sehingga menghasilkan keragaman dan kelengkapan fitur. Pada tahap ini ada penambahan *layer* dan nantinya akan dibentuk *pyramid feature* yang berguna untuk menghasilkan prediksi deteksi pada piksel yang kecil sehingga meningkatkan proses segmentasi. Dibandingkan dengan struktur *Neck* pada model YOLOv5 dan YOLOv7, YOLOv8 menghapus operasi konvolusi setelah pengambilan sampel di struktur PAN, yang mempertahankan performa aslinya sekaligus mencapai model yang ringan. YOLOv8 menggunakan *neck* yang sama dengan versi sebelumnya yaitu *PANet (Path Aggregation Network)* yang telah dimodifikasi dengan menambahkannya ke FPN agar mendapatkan akurasi yang lebih baik lagi (Wang, G. et al., 2023). Di bagian ini terdapat modul SPPF (*Spatial Pyramid Pooling-Fast*), beberapa blok C2f, dan blok konvolusi yang digunakan untuk mengekstrak fitur yang didapat dari backbone lalu disampaikan kepada bagian head sehingga dapat dilakukan pendekstrian.

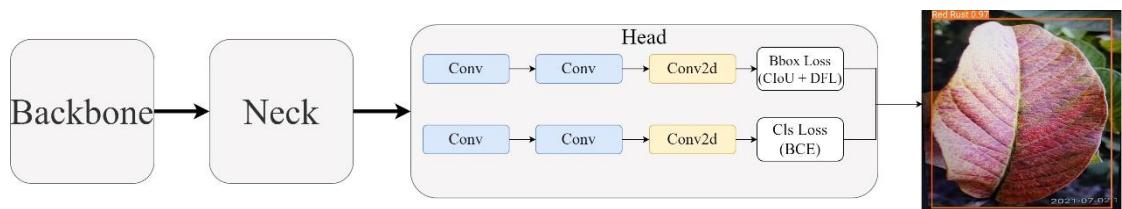
SPPF (*Spatial Pyramid Pooling-Fast*) mengumpulkan fitur-fitur pada berbagai skala spasial, memungkinkan jaringan menangkap informasi konteks pada resolusi berbeda. Dalam hal ini masukan dengan dimensi 640x640x3 akan diproses terlebih dahulu pada blok konvolusi dan C2f hingga dimensi *feature map* menjadi 80x80x256w. Nilai ‘w’ yang dimaksud pada dimensi tersebut merujuk kepada nilai ‘*widen\_factor*’ yang berbeda-beda nilainya disesuaikan pada jenis model YOLOv8 yang digunakan yang dapat dilihat pada gambar 2.10. Pada penelitian ini, model YOLOv8 yang digunakan adalah YOLOv8n.



**Gambar 3. 9** Piramida Fitur hasil SPPF

### 3. Head

Pada tahap ini dihasilkan kotak deteksi yang digunakan untuk mendapatkan koordinat atau bagian dari *bounding box* objek dan menentukan *class probabilities* pada suatu kelas. Pada tahap ini nantinya akan ada tujuh vektor dimana masing-masing vektor mewakili satu koordinat tertentu. Pada tahap ini juga nantinya akan diimplementasikan fungsi *IOU (Intersection Over Union)* dan *non max suppression* untuk menghindari hasil kotak yang tumpang tindih.



**Gambar 3. 10** Proses YOLOv8 hingga pemberian *bounding box* dan *class*

#### 3.2.4 Proses Training

Pada tahap ini seluruh data yang telah dipisah menjadi data *train* dan *validation* akan melalui proses *training*. Tahap training pada penelitian ini menggunakan algoritma YOLOv8 dengan bantuan model *pre-trained* yolov8n.pt yang telah diberikan parameter *epoch*, *batchsize* dan *imgsize*. Tahap training akan memberikan output berupa *bounding box* dan *confidence* pada setiap objek yang telah terdeteksi.

#### 3.2.5 Learned Model

Setelah proses pelatihan model menggunakan dataset selesai, akan dihasilkan sebuah *learned model* dalam bentuk file dengan ekstensi '.pt'. *Learned model* ini merupakan hasil dari proses pelatihan data, di mana model tersebut akan digunakan untuk menguji keakuratan deteksi program menggunakan data validasi.

Pada tahap validasi, data yang digunakan adalah data *validation* dimana data yang digunakan untuk memvalidasi atau mengevaluasi model setelah proses *training* dilakukan. Data validasi yang digunakan pada penelitian ini adalah 10% dari total dataset, yaitu 192. Proses validasi berfungsi untuk memastikan bahwa hasil model yang dilatih sudah memenuhi kriteria yang diharapkan dan merupakan model terbaik.

### 3.2.6 *TF-Lite Model*

Tahapan ini merupakan tahapan dimana *learned model* yang nantinya digunakan akan diubah kedalam *Tensorflow Lite Model* dengan ekstensi file ‘.pt’. Hal ini digunakan agar model yang dibuat dapat digunakan pada perangkat *android*. *Learned Model* nantinya akan terkompresi menjadi ukuran yang lebih kecil sehingga bisa digunakan kedalam perangkat seluler. Dengan menggunakan perintah YOLO, file ‘.pt’ yang diperoleh dapat dieksport menjadi *TF-Lite Model*.

### 3.2.7 *Proses Pengujian*

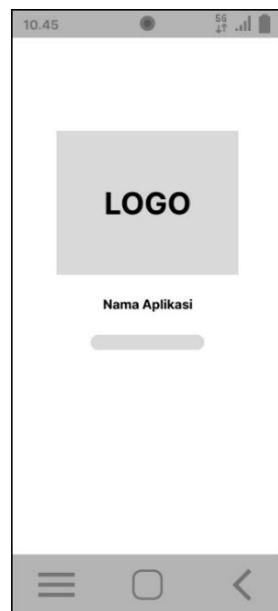
*TF-Lite Model* yang dihasilkan akan diuji dalam tahap pengujian untuk mengevaluasi kinerjanya dalam mendeteksi objek pada citra yang belum pernah dilihat sebelumnya. Pada tahap ini, data yang digunakan adalah data uji. Data uji merupakan data yang digunakan untuk menguji model setelah model terbaik diperoleh dari proses validasi. Data ini berfungsi untuk mengukur sejauh mana model dapat bekerja dengan baik pada data yang belum pernah dikenalnya. Dalam penelitian ini, data uji yang digunakan berjumlah 10% dari total data citra, yaitu sebanyak 192 citra. Pengujian ini juga bertujuan untuk memastikan apakah model dapat memberikan output yang diharapkan, yaitu mendeteksi penyakit pada kategori yang benar seperti *antraknosa*, *phytophthora*, *red rust*, *scab*.

## 3.3 Rancangan Antarmuka Sistem

Perancangan antarmuka sistem melibatkan penyusunan dan penjelasan desain layar dari aplikasi deteksi penyakit tanaman jambu biji yang dikenal sebagai "Jambotec". Pengembangan sistem ini menggunakan *Tensorflow Lite* untuk integrasi dengan platform *Android*. Aplikasi ini terdiri dari satu tampilan yang berjalan di perangkat *Android*, yang menampilkan hasil deteksi penyakit pada tanaman jambu biji. Tujuan dari proses ini adalah untuk memberikan gambaran yang jelas tentang desain aplikasi dan mempermudah pemahaman serta pembangunan aplikasi bagi pengguna. Berikut ini adalah rancangan tampilan yang telah dibentuk untuk aplikasi Jambotec.

### 3.3.1 Tampilan splash screen

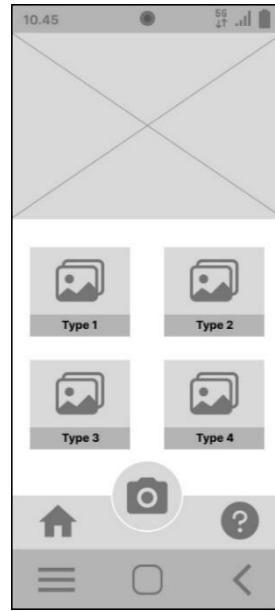
*Splash screen* adalah layar pertama yang muncul saat membuka sebuah aplikasi mobile. Pada platform *Android*, *splash screen* biasanya menampilkan logo atau elemen grafis dari aplikasi. Fungsinya adalah sebagai antarmuka pengguna awal yang memperkenalkan aplikasi kepada pengguna sebelum mereka masuk ke layar utama atau aktivitas dalam aplikasi tersebut.



**Gambar 3. 11** Tampilan *Splash Screen*

### 3.3.2 Tampilan Beranda

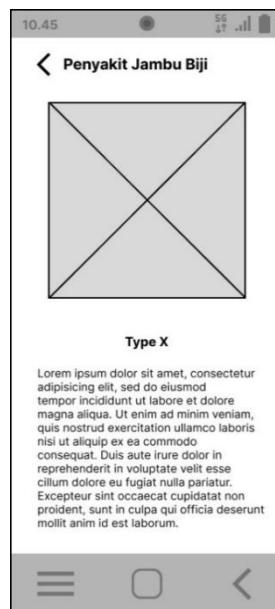
Tampilan ini adalah halaman beranda yang mana merupakan pratinjau dari aplikasi yang sedang berjalan, yang memuat 4 artikel mengenai informasi lebih lanjut terkait dari jenis-jenis penyakit yang paling sering menjangkit tanaman jambu biji. Selain itu, di navbar aplikasi juga terdapat tiga pilihan menu yang mengarahkan pengguna ke halaman-halaman yang sesuai dengan setiap menu.



**Gambar 3. 12** Rancangan Tampilan Halaman Beranda

### 3.3.3 Tampilan Halaman Artikel

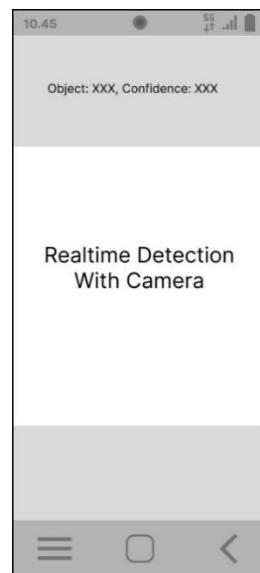
Tampilan ini akan memberi informasi terkait yang bersangkutan dengan artikel jenis penyakit tanaman jambu biji yang ingin diketahui oleh pengguna. Artikel ini juga memuat ciri dan cara pengendalian dari tanaman jambu biji yang terserang penyakit.



**Gambar 3. 13** Rancangan Tampilan Halaman Artikel

### 3.3.4 Tampilan Halaman Deteksi

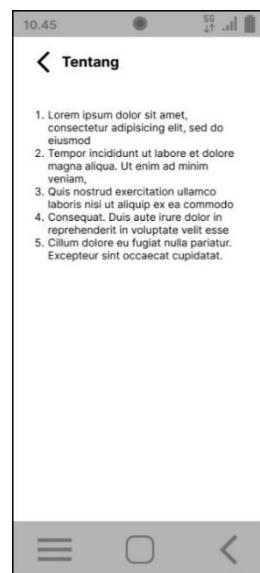
Di halaman ini, tersedia sebuah fungsi yang mengizinkan pengguna untuk mendeteksi jenis penyakit pada buah atau daun jambu biji dengan mengarahkan kamera ke bagian buah atau daun tanaman.



**Gambar 3. 14** Rancangan Tampilan Halaman Deteksi

### 3.3.5 Tampilan halaman *about*

Pada halaman ini, terdapat panduan yang memberikan informasi kepada pengguna tentang cara menggunakan aplikasi.



**Gambar 3. 15** Rancangan Tampilan Halaman *About*

## **BAB IV**

### **IMPLEMENTASI DAN PENGUJIAN SISTEM**

Dalam bab ini, akan dibahas proses implementasi deteksi jenis penyakit pada tanaman jambu biji menggunakan YOLOv8 sesuai dengan rancangan yang telah disusun sebelumnya. Selain itu, bab ini juga akan memuat analisis dan pemaparan hasil dari pengujian sistem.

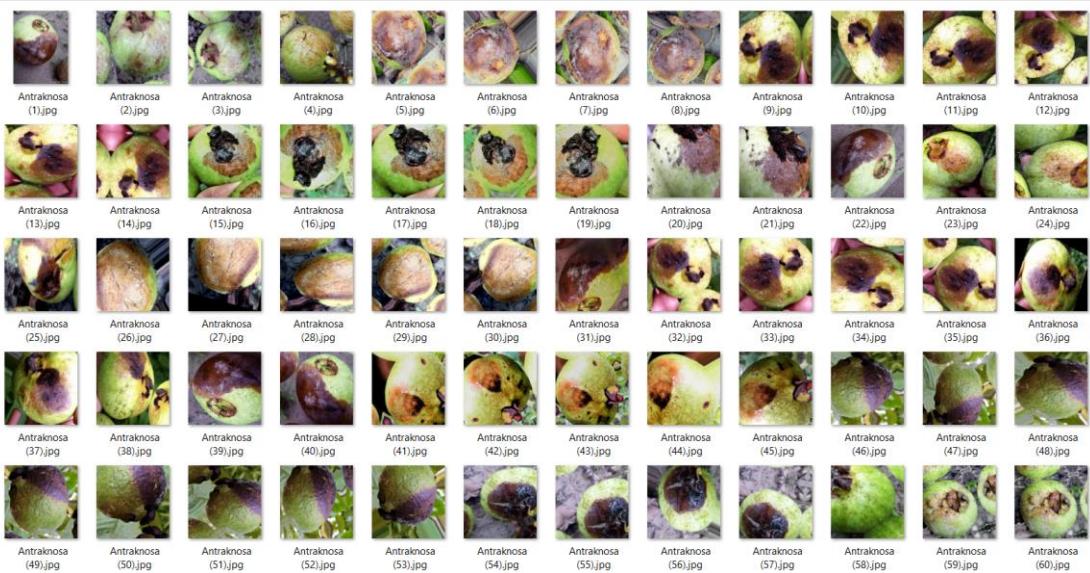
#### **4.1 Implementasi Sistem**

Pada tahap ini, metode YOLOv8 akan diterapkan untuk mendeteksi jenis penyakit pada tanaman jambu biji menggunakan perangkat *Android*. Untuk melaksanakan hal ini, diperlukan sistem perangkat keras dan perangkat lunak yang mendukung. Penelitian ini mengimplementasikan model sebagai sebuah aplikasi, yang memerlukan perangkat keras dan perangkat lunak untuk membangun sistem pendekripsi. Spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk mengembangkan aplikasi ini adalah sebagai berikut:

- Spesifikasi *Hardware* dan *Software* pada laptop:
  1. *HP Pavilion Gaming Laptop 15-dk1xxx*
  2. *Processor Intel Core I5-10300H @ 2,50GHz*
  3. *GPU Nvidia GeForce GTX 1650*
  4. *Random Access Memory (RAM) 8 GB*
  5. *Penyimpanan SSD 512 GB*
  6. *Android Studio*
  7. *Google Colab*
- Spesifikasi *Hardware* dan *Software* pada *smartphone*:
  1. *Samsung Galaxy S20 FE series*
  2. *Random Access Memory (RAM) 8 GB*
  3. *Screen Resolution 1080 x 2400.*
  4. *128 GB internal storage*
  5. *Camera 12MP/2160p*

## 4.2 Implementasi Data

Data gambar yang digunakan dalam penelitian ini adalah gambar tanaman jambu biji yang terkena penyakit, yang terdiri dari empat jenis, yaitu *Antraknosa*, *Red Rust*, *Phytophthora*, dan *Scab*. Data ini diperoleh dari salah satu ladang jambu biji di Medan dan juga menggunakan dataset yang tersedia di *Kaggle*, dengan total sebanyak 1926 gambar yang dibagi menjadi 481 citra setiap kategori penyakit. Contoh citra gambar yang digunakan dapat dilihat pada gambar-gambar di bawah ini:



**Gambar 4. 1 Beberapa Contoh Data Citra Penyakit *Antraknosa***



**Gambar 4. 2 Beberapa Contoh Data Citra Penyakit *Phytophthora***



**Gambar 4. 3 Beberapa Contoh Data Citra Penyakit *Red Rust***



**Gambar 4. 4 Beberapa Contoh Data Citra Penyakit *Scab***

### 4.3 Implementasi Perancangan Antarmuka

#### 4.3.1 Tampilan Halaman *Splash Screen*

Tampilan *splash screen* dirancang untuk memperkenalkan identitas peneliti, lengkap dengan nama aplikasi yang bertajuk “Jambotec”, logo aplikasi, dan instansi yang terkait. Tampilan ini muncul saat aplikasi pertama kali dibuka sebagai perantara sebelum pengguna diarahkan ke halaman utama.



**Gambar 4. 5** Tampilan *Splash Screen*

#### 4.3.2 Tampilan Halaman Beranda

Halaman *home* atau beranda merupakan tampilan utama dari aplikasi. Halaman *home* ini berisi fitur-fitur lain dari aplikasi yang didalamnya memuat 4 artikel mengenai informasi lebih lanjut terkait dari jenis-jenis penyakit yang paling sering menjangkit tanaman jambu biji. Selain itu, pada halaman beranda juga terdapat navbar aplikasi yang berisi 3 menu yang mengarahkan pengguna ke halaman lain. Adapun fitur menu pada halaman ini antara lain adalah *home*, *detect*, dan *about*.



**Gambar 4. 6 Tampilan *Home***

#### 4.3.3 Tampilan Halaman Artikel

Pada halaman artikel, pengguna dapat mengakses dan membaca mengenai informasi lebih lanjut tentang jenis penyakit tanaman jambu biji. Setiap artikel berisi informasi yang bermanfaat mengenai penyakit pada tanaman jambu biji. Artikel ini juga memuat serangkaian ciri dan cara pengendalian dari tanaman jambu biji yang terindikasi terkena penyakit yang bertujuan agar pengguna dapat mengenal ciri tanaman jambu biji yang terserang penyakit dan bagaimana cara penanganan yang tepat dan benar.



**Gambar 4. 7 Tampilan Artikel**

#### 4.3.4 Tampilan Halaman Deteksi

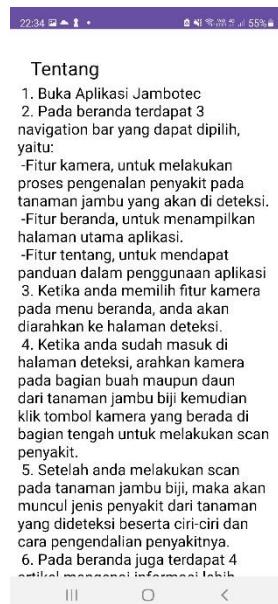
Apabila pengguna menekan icon *detect* pada halaman *home*, maka pengguna akan diarahkan tampilan halaman deteksi dimana fitur ini menggunakan kamera *smartphone android* sebagai alat untuk mendeteksi jenis penyakit tanaman jambu biji. Pada halaman ini pengguna dapat mengarahkan kamera pada bagian tanaman jambu biji yang terserang penyakit baik di bagian buah maupun daun dari tanaman. Kemudian dari objek tersebut akan muncul bounding box yang menampilkan jenis penyakit yang menyerang beserta nilai *confidence* dari objek tersebut.



**Gambar 4. 8** Tampilan Deteksi

#### 4.3.5 Tampilan Halaman Tentang

Halaman ini berisi penjelasan mengenai cara penggunaan aplikasi “Jambotec” dengan tujuan untuk membantu si pengguna dalam memahami cara kerja dari aplikasi beserta fitur-fitur yang ada didalamnya.



**Gambar 4. 9** Tampilan *About*

#### **4.4 Pelatihan Sistem**

Pelatihan sistem adalah langkah penting dalam melatih model agar bisa mengenali objek. Data yang digunakan dalam pelatihan ini telah melalui tahap *pre-processing*. Proses pelatihan sistem dilakukan menggunakan Google Colab sebagai alat bantu dan model *pre-trained* yolov8n.pt, yang merupakan model menengah dari YOLOv8.

The screenshot shows a Jupyter Notebook interface with the following details:

- File Path:** SkripsiJambu.ipynb
- Cells:** Code + Text
- Code Content:**

```
#model.train(data = data_path, epochs=100, batch = 8, device=0, patience=30, resume = True) # proses training
...
Ultralytics YOLOv8.2.35 Python-3.10.12 torch-2.3.0+cu121 CUDA10 (Tesla T4, 15102MB)
engine/trainer: task=detect, model_type=yolov8n.yaml, data=/content/drive/MyDrive/dataset (split) 6/data.yaml, epochs=100, time=None, p,
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
100% |██████████| 755K/755K [00:00<00:00, 26.2M/s]
Overriding model.yaml nc=nc with nc=4

from n import params module
arguments
0   -1  1   464 ultralytics.nn.modules.conv.Conv    [3, 16, 3, 2]
1   -1  1   4672 ultralytics.nn.modules.conv.Conv   [16, 32, 3, 2]
2   -1  1   7360 ultralytics.nn.modules.block.C2f  [32, 32, 1, True]
3   -1  1   18560 ultralytics.nn.modules.conv.Conv  [32, 64, 3, 2]
4   -1  2   49664 ultralytics.nn.modules.block.C2f [64, 64, 2, True]
5   -1  1   73984 ultralytics.nn.modules.conv.Conv  [64, 128, 3, 2]
6   -1  1   107920 ultralytics.nn.modules.block.C2f [128, 128, 2, True]
7   -1  1   195424 ultralytics.nn.modules.conv.Conv [128, 256, 1, 2]
8   -1  1   460288 ultralytics.nn.modules.block.C2f [256, 256, 1, True]
9   -1  1   164688 ultralytics.nn.modules.block.SPPF [256, 256, 1, True]
10  -1  1   0 torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
11  [-1, 6] 1   0 ultralytics.nn.modules.concat.Concat [1]
12  -1  1   148224 ultralytics.nn.modules.block.C2f [384, 128, 1]
13  -1  1   0 torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
14  [-1, 4] 1   0 ultralytics.nn.modules.concat.Concat [1]
15  -1  1   37248 ultralytics.nn.modules.block.C2f [192, 64, 1]
16  -1  1   36992 ultralytics.nn.modules.conv.Conv  [64, 64, 3, 2]
17  [-1, 12] 1   0 ultralytics.nn.modules.concat.Concat [1]
18  -1  1   123648 ultralytics.nn.modules.block.C2f [192, 128, 1]
19  -1  1   147712 ultralytics.nn.modules.conv.Conv  [128, 128, 3, 2]
20  [-1, 9] 1   0 ultralytics.nn.modules.concat.Concat [1]
21  -1  1   493056 ultralytics.nn.modules.block.C2f [384, 256, 1]
22  [15, 18, 21] 1   752092 ultralytics.nn.modules.head.Detect [4, [64, 128, 256]]
```

YOLOv8 summary: 225 layers, 3011628 parameters, 3011612 gradients, 8.2 GFLOPs

Executing (In 0m 29s) <cell line: 2> [train] > [train] > [do\_train] > [backward] > [backward] > [engine\_run\_backward()]

#### **Gambar 4. 10 Google Colab**

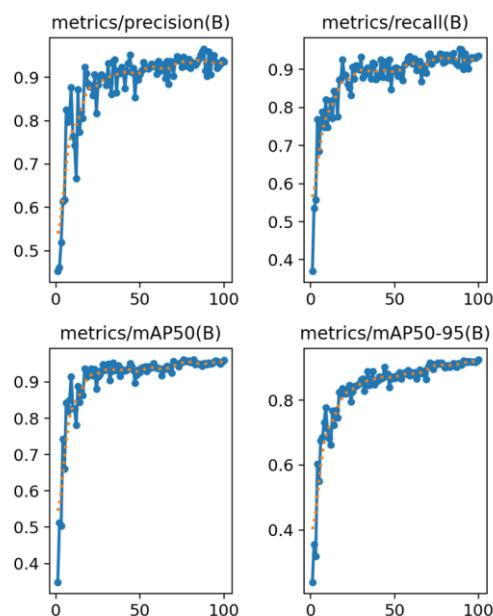
Dapat dilihat pada Gambar 4.10 *Google Colab* menyediakan *framework PyTorch* dan *runtime GPU NVIDIA* dengan dukungan CUDA secara gratis, dengan batas waktu penggunaan 12 jam per hari, *google colab* juga menyediakan penggunaan *Tesla T4 runtime type* secara gratis dengan interval waktu 24 jam.

Proses pelatihan dilakukan hingga sebanyak sembilan kali dengan berbagai konfigurasi nilai *epoch* dan *batch size* guna mendapat nilai akurasi yang optimal. Nilai *epoch* yang digunakan pada pelatihan ini adalah 30, 50, dan 100, dengan setiap *epoch* memiliki ukuran *batch size* masing-masing sebesar 8, 16, dan 32. Hasil dari percobaan untuk menemukan model terbaik dapat dilihat pada Tabel 4.1

**Tabel 4. 1** Hasil Percobaan *Training*

<i>Batch</i>	<i>Epochs</i>	<i>mAP</i>	<i>mAP 50-95</i>	<i>Precision</i>	<i>Recall</i>
8	30	0.95384	0.91085	0.92822	0.91776
8	50	0.94922	0.90097	0.94806	0.92186
8	100	0.95087	0.90779	0.95525	0.9323
16	30	0.95227	0.90935	0.94299	0.89759
16	50	0.94812	0.90094	0.95	0.91437
16	100	0.96009	0.92492	0.93554	0.93502
32	30	0.95351	0.9117	0.95836	0.92582
32	50	0.95348	0.91653	0.94066	0.92551
32	100	0.95682	0.92113	0.95418	0.93089

Setelah melalui seluruh proses pelatihan yang tampak pada tabel 4.1, maka didapatkan model terbaik dengan nilai *mAP* 0.96, *mAP 50-95* 0.924, *precision* 0.935, dan *recall* 0.935. Model dengan hasil terbaik pada YOLOv8 secara default akan tersimpan dengan nama file best.pt. grafik dari hasil proses pelatihan dapat dilihat pada gambar 4.11.

**Gambar 4. 11** Grafik Pelatihan

Pada gambar 4.11 di atas, dapat dilihat bahwa grafik *precision* dan *recall* menunjukkan akurasi model yang meningkat dengan cepat pada awal pelatihan dan kemudian menjadi stabil yang menunjukkan bahwa model mampu mendeteksi sebagian besar instance positif pada dataset. Grafik mAP50 menunjukkan peningkatan yang signifikan selama iterasi awal pelatihan dan kemudian stabil pada nilai yang tinggi. Ini menunjukkan bahwa model memiliki performa deteksi yang baik pada ambang batas IoU (*Intersection over Union*) 0.50. Pada grafik mAP50-95 dapat dilihat adanya peningkatan yang konsisten yang menunjukkan bahwa model memiliki performa deteksi yang baik pada berbagai ambang batas IoU.

Secara keseluruhan grafik pada gambar 4.11 menunjukkan peningkatan yang stabil di awal pelatihan, kemudian menjadi stabil seiring bertambahnya iterasi. Pada awal pelatihan, ada beberapa fluktuasi yang signifikan pada nilai metrik dikarenakan model sedang dalam tahap awal pembelajaran dan sedang mencoba menyesuaikan parameter-parameter. Setelah beberapa iterasi, metrik-metrik tersebut mencapai nilai maksimum dan menjadi lebih stabil yang menandakan bahwa model telah mencapai kemampuan optimalnya dalam mendeteksi sesuai dengan dataset yang diberikan.

## 4.5 Pengujian Sistem

Tahapan ini adalah tahap di mana dilakukan pengujian pada sistem yang menggunakan model terbaik dari hasil pelatihan yang sudah dilakukan di tahap sebelumnya. Pengujian ini bertujuan untuk mengukur kemampuan model dalam mendeteksi 4 jenis penyakit yang menjangkit pada tanaman jambu biji baik dari buah maupun daun tanaman. Yaitu penyakit *antraknosa*, *red rust* (karat merah), *scab* (bercak buah), dan *phytophthora* (jamur air).

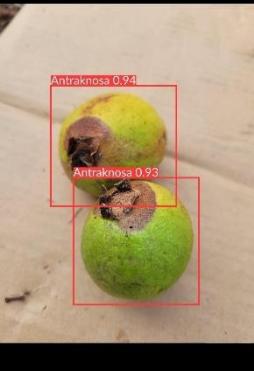
### 4.5.1 Pengujian Akurasi

Pengujian dilakukan dengan menggunakan kamera pada *smartphone* yang telah diinstal aplikasi yang menggunakan model tersebut. Uji sistem model deteksi akan melakukan evaluasi model mendeteksi keempat jenis penyakit pada tanaman jambu biji. Pengujian akurasi ini menggunakan data yang diambil dari salah satu kebun jambu biji dengan keseluruhan data uji sebanyak 192 data yang merupakan

10% dari keseluruhan data penelitian, kemudian data uji dibagi 4 sesuai *class* sehingga data uji menjadi 48 citra di setiap *class* penyakit. Contoh hasil pengujian deteksi penyakit menggunakan aplikasi yang sudah dikembangkan dapat dilihat pada tabel 4.2 berikut.

**Tabel 4. 2** Hasil Uji Sistem

No	Data	Aktual	Prediksi	Status
1	Object : Antraknosa, Confidence: 93.83711% 	<i>Antraknosa</i>	<i>Antraknosa</i>	Benar
2	Object: Antraknosa, Confidence: 89.21224% 	<i>Antraknosa</i>	<i>Antraknosa</i>	Benar
3	Object : Antraknosa, Confidence: 82.60028% 	<i>Antraknosa</i>	<i>Antraknosa</i>	Benar

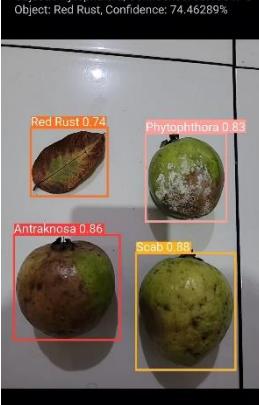
4	 <p>Object: Antraknosa, Confidence: 92.73889% Object: Antraknosa, Confidence: 94.31506%</p>	<i>Antraknosa</i> , <i>Antraknosa</i>	<i>Antraknosa</i> , <i>Antraknosa</i>	Benar
5	 <p>Object: Scab, Confidence: 76.40908%</p>	<i>Antraknosa</i>	<i>Scab</i>	Salah
6	 <p>Object : Phytophthora, Confidence: 86.23709%</p>	<i>Phytophthora</i>	<i>Phytophthora</i>	Benar

7	<p>Object : Phytophthora, Confidence: 93.35704%</p> 	<i>Phytophthora</i>	<i>Phytophthora</i>	Benar
8	<p>Object : Phytophthora, Confidence: 81.35186%</p> 	<i>Phytophthora</i>	<i>Phytophthora</i>	Benar
9	<p>Object : Phytophthora, Confidence: 93.61104%</p> 	<i>Phytophthora</i>	<i>Phytophthora</i>	Benar

10	 <p>Object: Antraknosa, Confidence: 69.55026%</p>	<i>Phytophthora</i>	<i>Antraknosa</i>	Salah
11	 <p>Object: Red Rust, Confidence: 89.89258%</p>	<i>Red Rust</i>	<i>Red Rust</i>	Benar
12	 <p>Object: Red Rust, Confidence: 86.083984%</p>	<i>Red Rust</i>	<i>Red Rust</i>	Benar

13	<p>Object: Red Rust, Confidence: 88.41109% Object: Red Rust, Confidence: 89.6095%</p> 	<i>Red Rust, Red Rust</i>	<i>Red Rust, Red Rust</i>	Benar
14	<p>Object: Red Rust, Confidence: 93.88403%</p> 	<i>Red Rust</i>	<i>Red Rust</i>	Benar
15	<p>Object: Antraknosa, Confidence: 68.25729%</p> 	<i>Red Rust</i>	<i>Antraknosa</i>	Salah

16	<p>Object: Scab, Confidence: 77.73207%</p> 	<i>Scab</i>	<i>Scab</i>	Benar
17	<p>Object: Scab, Confidence: 93.22638%</p> 	<i>Scab</i>	<i>Scab</i>	Benar
18	<p>Object : Scab, Confidence: 90.44387%</p> 	<i>Scab</i>	<i>Scab</i>	Benar

19	 <p>Object: Scab, Confidence: 92.73306%</p>	<i>Scab</i>	<i>Scab</i>	Benar
20	 <p>Object: Antraknosa, Confidence: 72.70455%</p>	<i>Scab</i>	<i>Antraknosa</i>	Salah
21	 <p>Object: Scab, Confidence: 87.59766% Object: Antraknosa, Confidence: 86.47461% Object: Phytophthora, Confidence: 83.10547% Object: Red Rust, Confidence: 74.46289%</p>	<i>Scab,</i> <i>Antraknosa,</i> <i>Phytophthora,</i> <i>Red Rust</i>	<i>Scab,</i> <i>Antraknosa,</i> <i>Phytophthora,</i> <i>Red Rust</i>	Benar

Tabel diatas merupakan beberapa contoh dari data hasil uji sistem yang telah dikembangkan untuk mendeteksi penyakit pada tanaman jambu biji. Setelah melakukan pengujian pada sistem selanjutnya peneliti melakukan evaluasi berdasarkan hasil tersebut menggunakan *confusion matrix*. Hasil *confusion matrix* dari penelitian ini dapat dilihat pada tabel 4.3 berikut.

**Tabel 4. 3** *Confusion Matrix*

		Aktual				Total
		Antrknosa	Phytophthora	Red Rust	Scab	
Prediksi	Antrknosa	45	1	4	3	53
	Phytophthora	1	47	0	0	48
	Red Rust	1	0	44	0	45
	Scab	1	0	0	45	46
	Total	48	48	48	48	192

Dilihat dari *confusion matrix* yang mengevaluasi hasil uji sistem diatas, maka tercatat ada beberapa kesalahan dalam mendekripsi penyakit tanaman jambu biji dari total 192 data pengujian. Kesalahan tersebut terjadi karena adanya kesamaan tekstur maupun warna antara buah jambu biji, termasuk bentuk dan ciri khas dari buah dan daun jambu biji yang terserang penyakit, sehingga aplikasi salah dalam mendekripsi jenis penyakit yang ada.

**Tabel 4. 4** Nilai Pengujian TP, FP, dan FN

	TP	FP	FN
Antrknosa	45	8	3
Phytophthora	47	1	1
Red Rust	44	1	4
Scab	45	1	3
Total	181	11	11

Pada tabel di atas dapat dilihat sebanyak 181 data digolongkan sebagai *True Positive*, sehingga dapat diartikan data memiliki hasil yang sesuai dengan nilai aktualnya. Untuk nilai *False Positive* ada sebanyak 11 data dimana data yang seharusnya bernilai negatif di kenali oleh sistem bernilai positif. Sementara untuk nilai *False Negative* ada sebanyak 11 data dimana data yang seharusnya bernilai positif diberi nilai negatif oleh sistem. Sehingga dari nilai tersebut dapat diperoleh *Precision*, *Recall* dan *F-Score* melalui perhitungan berikut.

*1. Precision*

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Precision Antraknosa} = \frac{45}{53} \times 100\% = 84,90\%$$

$$\text{Precision Phytophthora} = \frac{47}{48} \times 100\% = 97,91\%$$

$$\text{Precision Red Rust} = \frac{44}{45} \times 100\% = 97,77\%$$

$$\text{Precision Scab} = \frac{45}{46} \times 100\% = 97,82\%$$

*2. Recall*

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Recall Antraknosa} = \frac{45}{48} \times 100\% = 93,75\%$$

$$\text{Recall Phytophthora} = \frac{47}{48} \times 100\% = 97,91\%$$

$$\text{Recall Red Rust} = \frac{44}{48} \times 100\% = 91,66\%$$

$$\text{Recall Scab} = \frac{45}{48} \times 100\% = 93,75\%$$

### 3. F1-Score

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

$$F1 - Score Antraknosa = 2 \times \frac{(93,75\% \times 84,90\%)}{(93,75\% + 84,90\%)} = 89,10\%$$

$$F1 - Score Phytophthora = 2 \times \frac{(97,91\% \times 97,91\%)}{(97,91\% + 97,91\%)} = 97,91\%$$

$$F1 - Score Red Rust = 2 \times \frac{(91,66\% \times 97,77\%)}{(91,66\% + 97,77\%)} = 94,51\%$$

$$F1 - Score Scab = 2 \times \frac{(93,75\% \times 97,82\%)}{(93,75\% + 97,82\%)} = 95,74\%$$

Dari seluruh perhitungan yang telah dilakukan di atas, maka nilai-nilai tersebut dapat dirangkum pada tabel berikut.

**Tabel 4.5** Nilai Precision, Recall, dan F1-Score

	Precision	Recall	F1-Score
Antraknosa	84,90%	93,75%	89,10%
Phytophthora	97,91%	97,91%	97,91%
Red Rust	97,77%	91,66%	94,51%
Scab	97,82%	93,75%	95,74%

Berdasarkan seluruh uji sistem yang dilakukan pada model aplikasi deteksi jenis penyakit tanaman jambu biji menggunakan algoritma YOLO-V8 maka didapatkan nilai akurasi sistem lewat perhitungan berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

$$Accuracy = \frac{181}{192} \times 100\% = 94,27\%$$

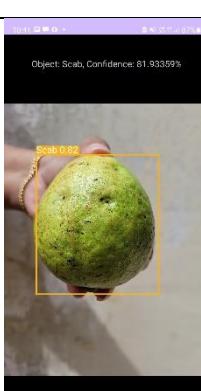
Dengan nilai akurasi sebesar 94,27%, maka dapat dikatakan bahwa sistem dapat melakukan deteksi pada 4 jenis penyakit tanaman jambu biji secara *realtime* dengan baik dimana sistem dapat dijalankan pada perangkat *mobile*. Di sisi lain, sistem juga menghasilkan beberapa kesalahan dalam prediksi yang menunjukkan bahwa metode yang digunakan masih belum sempurna dan memiliki kekurangan dalam mendeteksi objek buah dan daun yang terkena penyakit. Kesalahan tersebut disebabkan oleh kurangnya jumlah dan variasi data yang dilatih sehingga apabila adanya kemiripan pada buah dan daun jambu biji yang akan dideteksi, maka sistem menjadi bingung dalam menentukan jenis penyakit yang menjangkit pada tanaman jambu biji tersebut.

#### 4.5.2 Pengujian Dengan Jarak

Pengujian dengan jarak adalah proses evaluasi perangkat atau sistem dengan mengukur selisih antara hasil yang dihasilkan oleh sistem dan nilai yang diharapkan atau dianggap benar. Metode ini biasanya digunakan untuk menilai kualitas atau akurasi sistem dalam menghasilkan output dengan membandingkannya dengan nilai yang seharusnya benar.

**Tabel 4. 6** Hasil Pengujian Pada Jarak

	<b>Jarak</b>			
	<b>5cm</b>	<b>10cm</b>	<b>20cm</b>	<b>30cm</b>
<b>Antraknosa</b>	 <input type="button" value="DETECT"/> <span style="font-size: small;">III    □    &lt;</span>	 <input type="button" value="DETECT"/> <span style="font-size: small;">III    □    &lt;</span>	 <input type="button" value="DETECT"/> <span style="font-size: small;">III    □    &lt;</span>	 <input type="button" value="DETECT"/> <span style="font-size: small;">III    □    &lt;</span>

<b><i>Red Rust (Karat Merah)</i></b>				
<b><i>Scab (Bercak Buah)</i></b>				
<b><i>Phytophthora (Jamur Air)</i></b>				

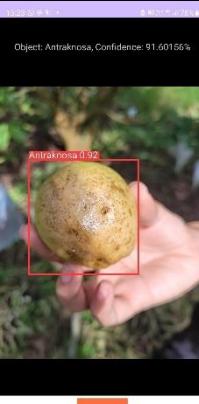
Dalam Tabel 4.6, dijelaskan bagaimana pengujian dilakukan pada ukuran jarak yang berbeda yaitu mulai dari 0 hingga 30 cm. Dari hasil pengujian dapat dilihat bahwa pada jarak sekitar  $\pm 10$  cm dari kamera, sistem mampu mendeteksi jenis penyakit pada jambu biji dengan baik. Hal yang sama terjadi pada jarak sekitar  $\pm 20$ cm. Namun, ketika pengujian dilakukan pada jarak sekitar  $\pm 30$ cm, sistem tidak dapat mengenali jenis penyakit jambu biji yang sudah dilatih sebelumnya. Jarak yang

dianggap ideal untuk melakukan deteksi pada sistem ini adalah antara 5cm hingga 25cm. Jika jarak terlalu dekat berkisar kurang dari 5cm atau terlalu jauh dari 25cm, maka akan terjadi kesalahan dalam mendeteksi pada sistem dan sistem tidak dapat mengenali objek dengan akurat, bahkan tidak dapat mendeteksi penyakit jambu biji.

#### 4.5.3 Pengujian *Angle*

Pengujian dengan *angle* atau sudut yang berbeda adalah teknik yang digunakan untuk mengukur kesanggupan sistem dalam mendeteksi sebuah objek dari berbagai sudut atau kemiringan. Metode ini biasanya digunakan untuk menilai kualitas atau akurasi sistem dalam menghasilkan output dengan membandingkannya dengan nilai yang seharusnya benar. Hasil pengujian pada sudut yang berbeda dapat dilihat pada tabel 4.7 berikut.

**Tabel 4. 7** Hasil Pengujian *Angle*

	<i>Angle</i>		
	Sejajar	Atas/bawah	Samping
<i>Antraknosa</i>			
<i>Red Rust</i> <b>(Karat Merah)</b>			

<b><i>Scab (Bercak Buah)</i></b>			
<b><i>Phytophthora (Jamur Air)</i></b>			

Dalam Tabel 4.7, dijelaskan bagaimana pengujian dilakukan pada tiga sudut yang berbeda yaitu sejajar, atas/bawah, dan samping. Dari hasil pengujian dapat dilihat bahwa sistem masih mampu melakukan deteksi dengan baik dari berbagai sudut. Namun, ketika pengujian dilakukan pada daun tanaman, maka sistem hanya bisa mendeteksi pada sudut tertentu saja dan hal ini dapat dilihat pada tabel 4.8 berikut.

**Tabel 4. 8** Pengujian Angle 70 Sampai 90 Derajat Dari Permukaan Daun

	Angle 70 sampai 90 derajat		
	Atas	Bawah	Samping
<b>Red Rust (Karat Merah)</b>			

Apabila sistem melakukan pengujian pada sudut 70 sampai 90 derajat dari permukaan daun, maka sistem akan kehilangan kemampuannya untuk mengenali objek pada daun dikarenakan permukaan daun yang bentuknya mendatar tidak seperti buah yang melingkar.

## **BAB V**

### **KESIMPULAN DAN SARAN**

Pada bab ini, dibahas mengenai hasil penelitian deteksi jenis penyakit pada tanaman jambu biji berdasarkan citra daun dan buah menggunakan YOLO-V8 dalam rangka menentukan jenis penyakit yang menjangkit tanaman jambu biji dengan memanfaatkan aplikasi berbasis *android*. Selain itu, juga diberikan saran-saran dan referensi untuk pengembangan penelitian selanjutnya.

#### **5.1 Kesimpulan**

Dari keseluruhan penelitian terkait penggunaan algoritma You Only Look Once (YOLO) versi 8 dengan hasil akhir yang didapatkan pada pengujian sistem maka dapat disimpulkan:

1. Penggunaan algoritma *You Only Look Once* (YOLO) versi 8 mampu mendeteksi 4 jenis penyakit pada tanaman jambu biji yang terdiri dari *antraknosa*, *phytophthora* (jamur air), *red rust* (karat merah), dan *scab* (bercak buah) secara *realtime*.
2. Pendekslan menggunakan algoritma *You Only Look Once* (YOLO) versi 8 memperoleh hasil akurasi yang tergolong sangat baik sebesar 94,27% dengan nilai rata-rata *precision* sebesar 94,60%, *recall* sebesar 94,26%, dan *F1-Score* sebesar 94,31%.
3. Jarak pengambilan optimal untuk sistem mampu mendekksi objek dengan benar berada pada rentang 5 – 25 cm.
4. Kegagalan dalam mendekksi dapat disebabkan oleh karakteristik penyakit pada tanaman jambu biji yang memiliki kemiripan.
5. Jika ada lebih dari satu objek yang terkena penyakit maka model akan sanggup memprediksi apabila objek tersebut masih tampak full dan berada dalam satu *frame* dan memiliki jarak optimal di depan kamera *smartphone*.

## 5.2 Saran

Adapun saran untuk pengembangan penelitian selanjutnya adalah:

1. Diharapkan pada penelitian selanjutnya menggunakan data yang lebih variatif dengan tampilan data, jarak, serta sudut pengambilan yang lebih banyak.
2. Melakukan penelitian dengan penerapan metode algoritma lain untuk dapat dibandingkan hasilnya dengan hasil yang didapat menggunakan algoritma YOLOv8
3. Penelitian selanjutnya diharapkan mencoba *batch size* yang lebih tinggi dan *epoch* yang lebih bervariasi apabila menggunakan spesifikasi *software* dan *hardware* yang tinggi.

## DAFTAR PUSTAKA

- Agbulos, K., Sarmiento, Y., & Villaverde, J. (2021). Identification of Leaf Blast and Brown Spot Diseases on Rice Leaf with YOLO Algorithm. doi:10.1109/ICCSSE52761.2021.9545153
- Alwanda, M. R., Ramadhan, R. P., & Alamsyah, D. (2020). Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle. *1*.
- Amusa, N. A., Ashaye, O. A., Amadi, J., & Oladapo, O. (2006). Guava Fruit Anthracnose and the Effects on its Nutritional and Market Values in Ibadan, Nigeria. doi:10.3923/jas.2006.539.542
- Asim, M., Ullah, S., Razzaq, A., & Qadri, S. (2022). Varietal Discrimination of Guava (*Psidium guajava*) Leaves Using Multi Features Analysis. doi:10.1080/10942912.2022.2158863
- Cavalin, P., & Oliveira, L. (2019). Confusion Matrix-Based Building of Hierarchical Classification. doi:10.1007/978-3-030-13469-3\_32
- Daoud, E., Khalil, N., & Gaedke, M. (n.d.). Implementation of A One Stage Object Detection Solution to Detect Counterfeit Products Marked With A Quality Mark. *17*. doi:10.33965/ijcsis\_2022170103
- Ghoni, U., Fatkhunnajah, E., Saputra, H., & Cahyani, A. (2021). Deteksi Penyakit Daun Pada Citra Daun Jambu Biji Menggunakan Segmentasi Warna.
- Golchubian, A., Marques, O., & Nojoumian, M. (2021). Photo Quality Classification Using Deep Learning. doi:10.1007/s11042-021-10766-7
- Hanik, N. R., Hidayati, S. N., Fitriani, R. D., Cahyanti, F. A., Oktavianingtyas, D., & Wahyuni, T. (2023). Identification of Pests and Diseases Crystal Guava (*Psidium guajava* L.) in Ngargoyoso District, Karanganyar Regency.
- Howlader, M. R., Habiba, U., Faisal, R. H., & Rahman, M. M. (2019). Automatic Recognition of Guava Leaf Diseases using Deep Convolution Neural Network. doi:10.1109/ECACE.2019.8679421

- Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of YOLO Algorithm Developments. doi:10.1016/j.procs.2022.01.135
- Kusumanto, R., & Tompunu, A. N. (2011). Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB.
- Laksono, A. T., Endryansyah, Rusmamto, P. W., & Zuhrie, M. S. (2022). Pengolahan Citra Digital Buah Murbei Dengan Algoritma LDA (Linear Discriminant Analysis. 4.
- Lou, H., Duan, X., Guo, J., Liu, H., Gu, J., Bi, L., & Chen, H. (2023). DC-YOLOv8: Small Size Object Detection Algorithm Based On Camera Sensor. doi:10.20944/preprints202304.0124.v1
- Naeem, H., Ahmad, J., & Tayyab, M. (2013). Real-Time Object Detection and Tracking. doi:10.1109/INMIC.2013.6731341
- Nissa, N. F., Janiati, A., Cahya, N., Anton, & Astuti, P. (2021). Application of Deep Learning Using Convolutional Neural Network (CNN) Method for Women's Skin Classification. 8. doi:10.15294/sji.v8i1.26888
- Panda, S., & Somani, V. (2022). Certain Investigations of Guava Leaf Disease Measurement in Necrotic Areas with Image Processing Techniques. doi:10.1109/ICSCDS53736.2022.9761035
- Qadri, S. A., Huang, N.-F., Wani, T. M., & Bhat, S. A. (2023). Plant Disease Detection and Segmentation using End-to-End YOLOv8: A Comprehensive Approach. doi:10.1109/ICCSCE58721.2023.10237169
- Sirisha, U., Praveen, S. P., Srinivasu, P. N., Barsocchi, P., & Bhoi, A. K. (2023). Statistical Analysis of Design Aspects of Various YOLO-Based Deep Learning Models for Object Detection. doi:<https://doi.org/10.1007/s44196-023-00302-w>
- Soeb, M. J., Jubayer, M. F., Tarin, T. A., Al Mamun, M. R., Ruhad, F. M., Parven, A., . . . Islam, M. M. (2023). Tea Leaf Disease Detection and Identification Based on YOLOv7 (YOLO-T).

Terven, J. R., & Esparaza, D. M. (2023). *A COMPREHENSIVE REVIEW OF YOLO: FROM YOLOV1 AND BEYOND.*

Thangaraj, R., Mohan, Moulik, Logeshwari, Loganathan, & Koushika. (2023). A Comparative Study of Deep Learning Models for Guava Leaf Disease Detection. doi:10.1109/ICAECT57570.2023.10117860



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI**  
**UNIVERSITAS SUMATERA UTARA**  
**FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI**

Jalan Universitas No. 9A Gedung A, Kampus USU Medan 20155, Telepon: (061) 821007  
Laman: <http://Fasilkomti.usu.ac.id>

**KEPUTUSAN  
DEKAN FAKULTAS ILMU KOMPUTER  
DAN TEKNOLOGI INFORMASI  
NOMOR : 2751/UN5.2.14.D/SK/SPB/2024**

**DEKAN FAKULTAS ILMU KOMPUTER  
DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA**

- Membaca : Surat Permohonan Mahasiswa Fasilkom-TI USU tanggal 11 Juli 2024 perihal permohonan ujian skripsi:  
Nama : CHRISTOPHER MIANDO IMANUEL MANURUNG  
NIM : 191402117  
Program Studi : Sarjana (S-1) Teknologi Informasi  
Judul Skripsi : Deteksi Jenis Penyakit Tanaman Jambu Biji Berdasarkan Citra Daun dan Buah Menggunakan Yolo V-8
- Memperhatikan : Bawa Mahasiswa tersebut telah memenuhi kewajiban untuk ikut dalam pelaksanaan Meja Hijau Skripsi Mahasiswa pada Program Studi Sarjana (S-1) Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara TA 2023/2024.
- Menimbang : Bawa permohonan tersebut diatas dapat disetujui dan perlu ditetapkan dengan surat keputusan
- Mengingat : 1. Undang-undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional.  
2. Peraturan Pemerintah Nomor 17 tahun 2010 tentang pengelolaan dan penyelenggara pendidikan.  
3. Keputusan Rektor USU Nomor 03/UN5.1.R/SK/SPB/2021 tentang Peraturan Akademik Program Sarjana Universitas Sumatera Utara.  
4. Surat Keputusan Rektor USU Nomor 1876/UN5.1.R/SK/SDM/2021 tentang pengangkatan Dekan Fasilkom-TI USU Periode 2021-2026
- MEMUTUSKAN**
- Menetapkan :  
Pertama : Membentuk dan mengangkat Tim Penguji Skripsi mahasiswa sebagai berikut:  
Ketua : Mohammad Fadly Syah Putra, B.Sc., M.Sc.  
NIP: 198301292009121003  
Sekretaris : Fahrurrozi Lubis B.IT., M.Sc.IT  
NIP: 198610122018052001  
Anggota Penguji : Ivan Jaya S.Si, M.Kom.  
NIP: 198407072015041001  
Anggota Penguji : Ade Sarah Huzaifah S.Kom., M.Kom  
NIP: 198506302018032001  
Moderator : -  
Panitera : -
- Kedua : Segala biaya yang diperlukan untuk pelaksanaan kegiatan ini dibebankan pada Dana Penerimaan Bukan Pajak (PNPB) Fasilkom-TI USU Tahun 2024.
- Ketiga : Keputusan ini berlaku sejak tanggal ditetapkan dengan ketentuan bahwa segala sesuatunya akan diperbaiki sebagaimana mestinya apabila dikemudian hari terdapat kekeliruan dalam surat keputusan ini.

Tembusan :

1. Ketua Program Studi Sarjana (S-1) Teknologi Informasi
2. Yang bersangkutan
3. Arsip

Medan  
Ditandatangani secara elektronik oleh:  
Dekan

#TTE

Maya Silvi Lydia  
NIP 197401272002122001