

**PENGEMBANGAN APLIKASI DETEKSI PENYAKIT PADA DAUN PADI  
MENGUNAKAN *COMPUTER VISION* BERBASIS PYTHON DAN *YOLO***

**SKRIPSI**

*Ditulis untuk memenuhi sebagian persyaratan penyelesaian studi dan  
mendapatkan gelar sarjana komputer*



**Oleh**

**GUSWITA HELMI**

**NIM. 2170005**

**PROGRAM STUDI INFORMATIKA**

**UNIVERSITAS ADZKIA**

**2025**

## LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : GUSWITA HELMI

NIM : 2170005

Program Studi : Informatika

Menyatakan :

1. Sesungguhnya skripsi/tugas akhir yang saya susun ini merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam skripsi/tugas akhir yang saya peroleh dari hasil karya tulis orang lain, telah saya tuliskan sumbernya dengan jelas, sesuai dengan kaidah penulisan ilmiah.
2. Jika dalam pembuatan skripsi/tugas akhir baik pembuatan program maupun skripsi/tugas akhir secara keseluruhan ternyata terbukti dibuatkan oleh orang lain, maka saya bersedia menerima sanksi akademik, berupa pembatalan skripsi/tugas akhir dan mengulang penelitian serta mengajukan judul baru.

Demikianlah surat pernyataan ini saya buat dengan sesungguhnya tanpa ada paksaan dari pihak manapun.

Padang, 25 Maret 2025

Guswita Helmi

**LEMBAR PENGESAHAN SKRIPSI**

**PENGEMBANGAN APLIKASI DETEKSI PENYAKIT PADA DAUN PADI  
MENGUNAKAN *COMPUTER VISION* BERBASIS PYTHON DAN *YOLO***

**Yang dipersiapkan dan disusun oleh**

**GUSWITA HELMI**

**2170005**

Telah memenuhi syarat untuk dipertahankan didepan Dewan Penguji

Pada Ujian Komprehensif

Padang, 25 Maret 2025

Pembimbing I

Pembimbing II

**(Yomei Hendra, S.Kom., M.Kom.)**

**NIDN. 1015059103**

**(Aldo Eko Syaputra, S.Kom., M.Kom.)**

**NIDN. 1007079601**

**LEMBAR PENGESAHAN PENGUJI SIDANG SKRIPSI**

**PENGEMBANGAN APLIKASI DETEKSI PENYAKIT PADA DAUN PADI  
MENGUNAKAN *COMPUTER VISION* BERBASIS PYTHON DAN *YOLO***

**OLEH :**

**GUSWITA HELMI**

**2170005**

**PROGRAM STUDI INFORMATIKA**

**Skripsi ini telah dinyatakan LULUS oleh  
Penguji Materi Pada Sidang Skripsi Program Studi Strata 1  
Program Studi Informatika  
Universitas Adzkia Pada Hari/Tgl : Selasa / 25 / Maret/ 2025**

**TIM PENGUJI :**

1. **Jefri Rahmad Mulia, S.Kom., M.Kom** -----
2. **Muhammad Thoriq, S.Kom., M.Kom** -----

**Padang, 25 Maret 2025**

**Mengetahui**

**Ketua Prodi Informatika**

**Universitas Adzkia**

**(Muhammad Thoriq, S.Kom., M.Kom)**

**NIDN. 1005109203**

**LEMBAR PENGESAHAN LULUS SIDANG SKRIPSI**  
**PENGEMBANGAN APLIKASI DETEKSI PENYAKIT PADA DAUN PADI**  
**MENGGUNAKAN *COMPUTER VISION* BERBASIS PYTHON DAN *YOLO***

Yang dipersiapkan dan disusun oleh

**GUSWITA HELMI**

**2170005**

Yang telah dipertahankan didepan Dewan Penguji

Pada tanggal 25 Maret 2025

Dan dinyatakan telah lulus

Memenuhi syarat

Pembimbing I

Pembimbing II

**(Yomei Hendra, S.Kom., M.Kom.)**

**NIDN. 1015059103**

**(Aldo Eko Syaputra, S.Kom., M.Kom.)**

**NIDN. 1007079601**

Diterima dan disahkan sebagai salah satu syarat untuk  
Meraih gelar **Sarjana Komputer** (Program Studi Informatika)

Wakil Rektor I  
Universitas Adzkia

Ketua Program Studi Informatika  
Universitas Adzkia

**Dr. Alfadlani, M.T.**

**NIK. 197501122005011002**

**(Muhammad Thoriq, S.Kom., M.Kom.)**

**NIDN. 1005109203**

## ***ABSTRACT***

***Title*** : **Pengembangan Aplikasi Deteksi Penyakit Pada Daun Padi Menggunakan *Computer Vision* Berbasis Python Dan *Yolo***

***Name*** : **Guswita Helmi**

***NIM*** : **2170005**

***Study Program*** : **Informatika**

***Education Level*** : **S1**

***Mentor*** : **1. Yomei Hendra, S.Kom., M.Kom**  
**2. Aldo Eko Syaputra, S.Kom., M.Kom**

*Rice is one of the main agricultural commodities in Indonesia and is highly susceptible to various leaf diseases such as blast, blight, brown spot, and tungro. These diseases can significantly reduce agricultural productivity. Therefore, a system is needed to help farmers detect rice leaf diseases automatically and accurately. This study aims to develop a rice leaf disease detection application using computer vision technology based on Python and YOLO. The research method includes several stages: problem scoping, data acquisition, data exploration, modeling, evaluation, and deployment. The dataset was obtained from Roboflow and consists of five classes: blast, blight, brown spot, healthy, and tungro. The YOLO model was trained using Google Colab with specific parameters to improve detection accuracy.*

*The test results show that the developed model achieved an accuracy of 95% and a mean Average Precision (mAP) of 95%, indicating excellent performance in detecting rice leaf diseases. The web-based application was built using Flask and Bootstrap, enabling users to upload images of rice leaves and receive real-time detection results. With this application, farmers can easily identify diseases in rice plants and take appropriate actions to address the issues effectively.*

***Keywords: Computer Vision, Yolo, Disease Detection, Rice Leaf, Python***

## ABSTRAK

**Judul Skripsi** : Pengembangan Aplikasi Deteksi Penyakit Pada Daun Padi Menggunakan *Computer Vision* Berbasis Python Dan *Yolo*

**Nama** : Guswita Helmi

**NIM** : 2170005

**Program Studi** : Informatika

**Jenjang Pendidikan** : S1

**Pembimbing** : 1. Yomei Hendra, S.Kom., M.Kom  
2. Aldo Eko Syaputra, S.Kom., M.Kom

Padi merupakan salah satu komoditas pertanian utama di Indonesia yang rentan terhadap berbagai penyakit daun, seperti *blast*, *blight*, *brown spot*, dan *tungro*. Penyakit ini dapat menyebabkan penurunan produktivitas pertanian secara signifikan. Oleh karena itu, diperlukan suatu sistem yang dapat membantu petani dalam mendeteksi penyakit daun padi secara otomatis dan akurat. Penelitian ini bertujuan untuk mengembangkan aplikasi deteksi penyakit pada daun padi menggunakan teknologi *computer vision* berbasis Python dan *yolo*. Metode penelitian meliputi tahapan *problem scoping*, *data acquisition*, *data exploration*, *modeling*, *evaluation*, dan *deployment*. Dataset yang digunakan diperoleh dari *roboflow* dan terdiri dari lima kelas, yaitu *blast*, *blight*, *brown spot*, *healthy*, dan *tungro*. Model *yolo* dilatih menggunakan *google colab* dengan parameter tertentu untuk meningkatkan akurasi deteksi.

Hasil pengujian menunjukkan bahwa model yang dikembangkan memiliki tingkat akurasi sebesar 95% dan nilai *mAP* sebesar 95%, yang menunjukkan performa yang sangat baik dalam mendeteksi penyakit daun padi. Implementasi aplikasi berbasis web menggunakan *flask* dan *bootstrap* memungkinkan pengguna untuk mengunggah gambar daun padi dan mendapatkan hasil deteksi secara *real-time*. Dengan adanya aplikasi ini, petani dapat mengenali penyakit pada tanaman padi dan mengambil tindakan yang tepat untuk mengatasi permasalahan tersebut.

**Kata Kunci:** *Computer Vision*, *Yolo*, Deteksi Penyakit, Daun Padi, Python

## KATA PENGANTAR

Alhamdulillahirabbilalamin, dengan mengucapkan puji dan syukur kehadirat Allah SWT, atas rahmat dan karunia-nya yang telah diberikan kepada penulis sehingga hasil penelitian ini dapat disajikan dalam bentuk tulisan. Penelitian ini adalah tindak lanjut dari ilmu yang didapatkan dari proses perkuliahan. Hasil dari penelitian ini diharapkan dapat membantu pihak-pihak yang membutuhkan terutama untuk dapat meningkatkan efesien dan efektifitas kerja dalam menyajikan informasi.

Dalam melakukan penelitian ini penulis mengucapkan terima kasih, kepada:

1. Bapak Drs. H. Muhardanus, MH., Dt. Sampono Kayo selaku Ketua Yayasan Adzkie Sumatera Barat.
2. Bapak Prof. Dr. Irwan Prayitno, M.Sc., Psikolog selaku Rektor Universitas Adzkie.
3. Bapak Muhammad Thoriq, S.Kom., M.Kom selaku Ketua Program Studi Informatika.
4. Bapak Yomei Hendra, S.Kom., M.Kom selaku pembimbing I yang telah banyak memberikan pengetahuan dan arahan kepada penulis.
5. Bapak Aldo Eko Syaputra, S.Kom., M.Kom selaku pembimbing II yang telah banyak memberikan pengetahuan dan arahan kepada penulis.
6. Bapak dan Ibu dosen program studi Informatika Universitas Adzkie, yang telah banyak membantu penulis selama perkuliahan sehingga penulis dapat menyelesaikan perkuliahan ini.
7. Bapak dan Ibu karyawan dan karyawan serta staf , dosen Universitas Adzkie yang telah banyak membantu penulis dalam segi ilmu maupun waktu semasa penulis menjadi mahasiswa disini.
8. Kedua orang tua tercinta, ayahanda Oldi Gusman Jaya dan Ibunda Helmi Tati yang telah menjadi pilar utama dalam setiap langkah perjalanan ini. Doa, kasih sayang, dukungan, dan semangat yang tiada henti telah menjadi sumber kekuatan bagi penulis dalam menyelesaikan pendidikan S1 ini. Tanpa

bimbingan dan pengorbanan yang tulus, perjalanan ini tidak akan mungkin terwujud.

9. Kedua adik tercinta, Raju Andika Syaputra dan Olivia Wulandari yang menjadi penyemangat dan motivasi untuk menyelesaikan tugas akhir ini. Semoga dengan ini bisa menjadi pondasi dan acuan bagi kedua adik untuk melangkah kedepannya.
10. Teman-teman kelas Informatika angkatan 2021 yang telah memberikan dukungan, motivasi, dan kebersamaan yang sangat berharga selama perkuliahan sampai selesainya penyelesaian tugas akhir ini.
11. Teruntuk diri saya sendiri Guswita Helmi nim 2170005, yang sudah memberikan usaha terbaik hingga sampai pada titik ini. Tidak mudah menghadapi setiap tantangan, tetapi dengan kesabaran dan tekad yang kuat, semua dapat dilalui. Terima kasih telah bertahan, belajar dari setiap kegagalan, dan terus melangkah maju. Semoga perjalanan ini menjadi awal dari pencapaian yang lebih besar di masa depan.

Akhir kata, semoga skripsi ini dapat memberikan kontribusi terhadap masyarakat dan khususnya terhadap Universitas Adzkie. Penulis mengucapkan terima kasih dan mudah-mudahan laporan skripsi ini berguna dan dapat menambah wawasan bagi para pembaca.

Padang, 25 Maret 2025

Penulis

## DAFTAR ISI

<b>LEMBAR PERNYATAAN .....</b>	<b>ii</b>
<b>LEMBAR PENGESAHAN SKRIPSI .....</b>	<b>iii</b>
<b>LEMBAR PENGESAHAN PENGUJI SIDANG SKRIPSI .....</b>	<b>iv</b>
<b>LEMBAR PENGESAHAN LULUS SIDANG SKRIPSI .....</b>	<b>v</b>
<b>ABSTRACT .....</b>	<b>vi</b>
<b>ABSTRAK .....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>x</b>
<b>DAFTAR TABEL .....</b>	<b>xii</b>
<b>DAFTAR GAMBAR .....</b>	<b>xiii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah .....	6
1.3. Hipotesis.....	6
1.4. Batasan Masalah.....	7
1.5. Tujuan Penelitian.....	7
1.6. Manfaat Penelitian .....	7
<b>BAB II LANDASAN TEORI .....</b>	<b>9</b>
2.1. Kecerdasan Buatan.....	9
2.2. <i>Computer Vision</i> .....	10
2.3. Pengolahan Citra .....	11
2.4. <i>Yolo</i> .....	12
2.5. Python .....	15
2.6. <i>Flask</i> .....	15
2.7. <i>Bootstrap</i> .....	16
2.8. <i>Roboflow</i> .....	16
2.9. Pertanian.....	16
2.10. Padi.....	16
2.11. <i>State Of The Art</i> .....	19

<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>25</b>
3.1. Kerangka Penelitian .....	25
3.2. Tahapan Penelitian .....	26
3.3. Jadwal Penelitian.....	31
<b>BAB IV ANALISA DAN PERANCANGAN.....</b>	<b>33</b>
4.1. Analisa Data .....	33
4.2. Analisa Proses .....	46
4.3. Analisa Sistem.....	46
4.4. Perancangan Model.....	47
4.5. Perancangan <i>Interface</i> .....	50
<b>BAB V IMPLEMENTASI DAN PENGUJIAN .....</b>	<b>57</b>
5.1. Implementasi.....	57
5.2. Pengujian.....	83
<b>BAB VI PENUTUP .....</b>	<b>86</b>
6.1. Kesimpulan .....	86
6.2. Keterbatasan Sistem.....	87
6.3. Saran.....	87
<b>DAFTAR PUSTAKA.....</b>	<b>89</b>

## DAFTAR TABEL

<b>Tabel 2. 1</b> <i>State Of The Art</i> .....	19
<b>Tabel 3. 1</b> Jadwal Penelitian .....	31
<b>Tabel 4. 1</b> Tabel Perangkat Keras .....	47
<b>Tabel 4. 2</b> Tabel Perangkat Lunak .....	47
<b>Tabel 5. 1</b> Perbandingan Hasil Setiap <i>Epoch</i> .....	57
<b>Tabel 5. 2</b> <i>Confusion Matrix</i> .....	62
<b>Tabel 5. 3</b> Identifikasi TP, FP, FN, TN.....	62
<b>Tabel 5. 4</b> Perhitungan Setiap Kelas .....	65
<b>Tabel 5. 5</b> Pengujian Halaman Menu.....	83
<b>Tabel 5. 6</b> Pengujian Halaman <i>About</i> .....	84
<b>Tabel 5. 7</b> Pengujian Halaman <i>Check</i> .....	84
<b>Tabel 5. 8</b> Pengujian Halaman Menu.....	85

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Analogi Jaringan Penglihatan Manusia .....	11
<b>Gambar 2. 2</b> Analogi <i>Computer Vision</i> .....	11
<b>Gambar 2. 3</b> Koordinasi Citra Digital .....	12
<b>Gambar 2. 4</b> <i>Confusion Matrix</i> .....	13
<b>Gambar 2. 5</b> Logo Python.....	15
<b>Gambar 3. 1</b> Kerangka Penelitian.....	25
<b>Gambar 3. 2</b> <i>Ai Project Cycle</i> .....	27
<b>Gambar 3. 3</b> Contoh Dataset.....	28
<b>Gambar 3. 4</b> <i>Bounding Box</i> Pada Gambar .....	29
<b>Gambar 4. 1</b> <i>Sub Folder Dataset</i> .....	34
<b>Gambar 4. 2</b> <i>Folder Train</i> .....	34
<b>Gambar 4. 3</b> <i>Folder Valid</i> .....	35
<b>Gambar 4. 4</b> <i>Folder Test</i> .....	35
<b>Gambar 4. 5</b> <i>Bounding Box</i> Pada Gambar .....	37
<b>Gambar 4. 6</b> <i>Cloning Yolov5 Dan Install Dependencies</i> .....	39
<b>Gambar 4. 7</b> Daftar File Yang Sudah Di <i>Cloning</i> .....	39
<b>Gambar 4. 8</b> Menghubungkan <i>Google Collab Dan Google Drive</i> .....	40
<b>Gambar 4. 9</b> Perintah Mengekstrak File Zip .....	40
<b>Gambar 4. 10</b> Mengunduh Dataset <i>Roboflow</i> .....	41
<b>Gambar 4. 11</b> Proses <i>Training Data</i> Menggunakan <i>Epoch 100</i> .....	41
<b>Gambar 4. 12</b> Proses <i>Training Data</i> .....	42
<b>Gambar 4. 13</b> Hasil <i>Training Data</i> .....	42
<b>Gambar 4. 14</b> Prediksi Label Pada <i>Batch</i> Dataset Validasi .....	43
<b>Gambar 4. 15</b> Perintah Deteksi Gambar .....	44
<b>Gambar 4. 16</b> Perintah <i>Compress Folder</i> .....	45
<b>Gambar 4. 17</b> Download File Hasil <i>Compress</i> .....	45
<b>Gambar 4. 18</b> Model Analogi <i>Computer Vision</i> .....	48
<b>Gambar 4. 19</b> Masuk Ke Aplikasi <i>Agricare</i> .....	48
<b>Gambar 4. 20</b> Fitur Untuk Mendeteksi Penyakit Pada Daun Padi .....	49
<b>Gambar 4. 21</b> Gambaran Program Yang Sedang Berjalan .....	49

<b>Gambar 4. 22</b> Hasil Deteksi Pada Aplikasi.....	50
<b>Gambar 4. 23</b> Halaman <i>Home</i> .....	51
<b>Gambar 4. 24</b> Halaman <i>Home 2</i> .....	51
<b>Gambar 4. 25</b> Halaman <i>Home 3</i> .....	52
<b>Gambar 4. 26</b> Halaman <i>About</i> .....	52
<b>Gambar 4. 27</b> Halaman <i>About 2</i> .....	53
<b>Gambar 4. 28</b> Halaman <i>About 3</i> .....	53
<b>Gambar 4. 29</b> Halaman <i>Check</i> .....	54
<b>Gambar 4. 30</b> Halaman <i>Contact</i> .....	55
<b>Gambar 4. 31</b> Halaman <i>Contact 2</i> .....	55
<b>Gambar 4. 32</b> Halaman <i>Contact 3</i> .....	56
<b>Gambar 5. 1</b> <i>Confusion Matrix</i> .....	58
<b>Gambar 5. 2</b> <i>Recall</i> .....	59
<b>Gambar 5. 3</b> <i>Precision</i> .....	59
<b>Gambar 5. 4</b> <i>Precision Recall Curve</i> .....	60
<b>Gambar 5. 5</b> <i>F1-Score</i> .....	61
<b>Gambar 5. 6</b> <i>Loss Graph</i> .....	61
<b>Gambar 5. 7</b> Halaman Menu Pada Tampilan <i>Desktop</i> .....	67
<b>Gambar 5. 8</b> Halaman Menu Pada Tampilan <i>Mobile</i> Saat Pertama Dibuka.....	67
<b>Gambar 5. 9</b> Halaman Menu Pada Tampilan <i>Mobile</i> .....	68
<b>Gambar 5. 10</b> Penyakit Pada Daun Padi Pada Halaman Menu .....	68
<b>Gambar 5. 11</b> Penyakit <i>Blight</i> Pada Daun Padi .....	69
<b>Gambar 5. 12</b> Penyakit <i>Blast</i> Pada Daun Padi.....	70
<b>Gambar 5. 13</b> Penyakit <i>Brown spot</i> Pada Daun Padi.....	71
<b>Gambar 5. 14</b> Padi <i>Healthy</i> Pada Daun Padi .....	72
<b>Gambar 5. 15</b> Penyakit <i>Tungro</i> Pada Daun Padi .....	73
<b>Gambar 5. 16</b> Faq Pertanyaan Seputar <i>Agricare</i> .....	73
<b>Gambar 5. 17</b> Salah Satu <i>Faq</i> Pertanyaan Seputar <i>Agricare</i> .....	74
<b>Gambar 5. 18</b> Halaman <i>About</i> .....	75
<b>Gambar 5. 19</b> <i>About Agricare</i> .....	75
<b>Gambar 5. 20</b> Tutorial Penggunaan Aplikasi.....	75
<b>Gambar 5. 21</b> Halaman <i>Check</i> .....	76

<b>Gambar 5. 22</b> <i>Fitur Camera</i> .....	77
<b>Gambar 5. 23</b> <i>Camera Berhasil Di Capture</i> .....	77
<b>Gambar 5. 24</b> <i>Daun Padi Di Capture Secara Langsung</i> .....	78
<b>Gambar 5. 25</b> <i>Fitur Video</i> .....	78
<b>Gambar 5. 26</b> <i>Video Berhasil Di Deteksi</i> .....	79
<b>Gambar 5. 27</b> <i>Fitur Upload</i> .....	79
<b>Gambar 5. 28</b> <i>Gambar Berhasil Di Upload</i> .....	80
<b>Gambar 5. 29</b> <i>Halaman Contact</i> .....	81
<b>Gambar 5. 30</b> <i>Developer</i> .....	81
<b>Gambar 5. 31</b> <i>Whatsapp Dan Email</i> .....	81
<b>Gambar 5. 32</b> <i>Pengguna Diarahkan Ke Whatsapp</i> .....	82
<b>Gambar 5. 33</b> <i>Email Yang Dikirim Pengguna</i> .....	82

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Pertanian memainkan peran penting dalam menentukan ketahanan pangan dunia, karena dengan adanya pertanian banyak masyarakat dunia yang terbantu dalam memenuhi sumber kebutuhan bahan pangan (Putra Pranjaya et al., 2024). Menurut (Hawari et al., 2022) pertanian adalah kegiatan yang dilakukan oleh manusia untuk memproduksi pangan dengan menggunakan sumber daya alam. Kemajuan perekonomian Indonesia paling banyak dipengaruhi oleh sektor pertanian. Menurut (Yuliany et al., 2022) sektor pertanian sangat penting untuk ketahanan pangan, dan teknologi *artificial intelligence* banyak digunakan pada sektor pertanian. Menurut (Khumaidi & Hikmah, 2021) Indonesia terkenal dengan tanahnya yang subur, yang memungkinkan pertumbuhan berbagai jenis pertanian seperti singkong, jagung, dan padi.

Menurut (Yusuf et al., 2020) padi adalah salah satu komoditas pertanian yang memberikan kontribusi yang cukup besar terhadap perekonomian nasional. Menurut (Julianto et al., 2023) padi merupakan komoditas pangan utama bagi banyak negara, termasuk Indonesia, yang mengandalkannya sebagai sumber utama asupan karbohidrat. Menurut (Sayuthi et al., 2020) untuk membudidayakan tanaman padi memiliki banyak hambatan yang dihadapi, salah satunya adalah serangan hama. Upaya untuk menjaga kecukupan pangan, maka perlu dilakukan pengelolaan terhadap faktor penghambat, seperti serangan hama. Menurut (Ibrahim & Latifa, 2023) dalam industri pertanian, menentukan waktu panen yang tepat sangat penting untuk mencapai hasil panen terbaik.

Menurut (BPS-Statistics Indonesia, 2024) produksi padi di Indonesia sepanjang Januari hingga Desember 2023 mencapai sekitar 53,98 juta ton GKG, atau mengalami penurunan sebanyak 767,98 ribu ton GKG (1,40 persen) dibandingkan 2022 yang sebesar 54,75 juta ton GKG. Produksi padi tertinggi pada 2023 terjadi pada bulan Maret, yaitu sebesar 8,92 juta ton GKG, sementara produksi terendah terjadi pada bulan Desember, yaitu sekitar 1,97 juta ton GKG. Pada Januari 2024, produksi padi diperkirakan sebesar 1,50 juta ton GKG, dan potensi produksi

padi sepanjang Februari hingga April 2024 mencapai 17,10 juta ton GKG. Dengan demikian, total potensi produksi padi pada *subround* Januari hingga April 2024 diperkirakan mencapai 18,59 juta ton GKG, atau mengalami penurunan sekitar 22,25 juta ton GKG (17,54 persen) dibandingkan 2023 yang sebesar 22,25 juta ton GKG.

Produksi padi di Indonesia sering kali mengalami penurunan akibat serangan penyakit yang menyerang tanaman padi, yang tidak terdeteksi sejak dini. Minimnya pengetahuan petani tentang gejala awal penyakit membuat petani sulit untuk mengambil tindakan pencegahan yang tepat waktu. Akibatnya, penyakit menyebar dengan cepat dan mempengaruhi kualitas serta kuantitas hasil panen. Kondisi ini tidak hanya menurunkan produksi padi secara signifikan, tetapi juga berpotensi menyebabkan gagal panen yang berdampak buruk terhadap ketahanan pangan dan kesejahteraan ekonomi petani .

Menurut (Rijal et al., 2024) penyakit pada tanaman adalah salah satu penyebab penurunan kuantitas dan kualitas hasil pertanian. Penyakit yang menyerang daun tanaman padi dapat mengakibatkan penurunan jumlah produksi padi. Jenis penyakit pada daun tanaman padi sangat beragam. Orang awam terkadang tidak tahu jenis penyakit pada daun tanaman padi, sehingga akan berakibat terjadinya kesalahan mengidentifikasi jenis penyakitnya dan penanganannya.

Menurut (Saputra et al., 2021) banyak faktor yang dapat menyebabkan gagal panen dalam produktivitas tanaman padi, yaitu hama dan penyakit daun padi. Menurut (Agustiani et al., 2022) penyakit padi merupakan salah satu musuh utama dari terjadinya penurunan hasil panen yang mempengaruhi kualitas padi yang dihasilkan. Tanaman yang terserang penyakit menunjukkan gejala berupa bercak dengan pola dan warna tertentu. Gejala tersebut dapat diamati pada bagian tubuh padi, seperti telinga, daun, batang, dan akar padi. Karena penampang daun lebih lebar dari bagian lainnya, gejala penyakit paling jelas terlihat pada daun, perubahan warna dan bentuk bercak lebih jelas terlihat.

Menurut (Deden Miftah Fauzi et al., 2024) penyakit *brown spot*, *bacterial blight*, dan *blast* adalah salah satu penyakit padi yang paling meresahkan petani. Penyakit ini biasanya menyerang tanaman padi yang berumur lebih dari dua bulan

masa tanam, menyebabkan daun menjadi kuning, bercak, dan batang padi menjadi busuk. Sedangkan menurut (Putri Ayuni et al., 2023) gangguan pada daun padi adalah penyakit pada tanaman padi yang berdampak pada hasil produksi beras. Penyakit yang menyerang tanaman padi biasanya terjadi pada bagian daun, menyebabkan perubahan warna dan bercak. Penyakit daun yang paling sering menyerang tanaman padi adalah *blast*, *brown spot*, *leaf smut*, dan penyakit lainnya.

Menurut (Arifin et al., 2023) industri 4.0 mendorong otomatisasi sistem di berbagai sektor, termasuk dalam pengembangan *smart agriculture*, dimana teknologi seperti *computer vision* dan *artificial intelligence* yang mengimplementasikan pengolahan citra digital terus mengalami kemajuan. Menurut (Virgiawan et al., 2024) seiring dengan pesatnya perkembangan penelitian kecerdasan buatan, salah satunya tentang pendeteksian objek. Dengan adanya teknologi, manusia dapat secara otomatis dan efisien dalam mengenali dan mengidentifikasi objek-objek yang ada dalam sebuah gambar secara otomatis dan efisien. Pendeteksian objek merupakan salah satu bidang penting dalam *computer vision*, yaitu cabang ilmu yang berfokus pada bagaimana komputer dapat memahami, menganalisis, dan menafsirkan data visual, khususnya gambar atau video. Sedangkan menurut (Kurniawan, et al., 2022) *computer vision* adalah upaya manusia membuat perangkat lunak komputer yang bisa mengenali objek di dalam gambar digital. Ini berbeda dengan metode *image processing* (pengolahan citra) yang biasa dimanfaatkan untuk memanipulasi gambar digital, namun tidak bertujuan memahami isi gambar.

Pada saat ini, bidang pengenalan dan pendeteksian objek *computer vision* sedang berkembang dengan pesat dan mulai digunakan di berbagai bidang industri dan medis (Aini et al., 2021). Berbagai industri termasuk pertanian kini menggunakan teknologi *computer vision* yang telah berkembang pesat. *Computer vision* digunakan pada bidang pertanian untuk mengawasi kesehatan tanaman, menemukan hama, dan mengoptimalkan pemupukan dan irigasi. Drone yang dilengkapi dengan kamera dan sensor dapat mengambil gambar lahan pertanian dan menggunakan algoritma untuk menilai kondisi tanaman secara *real-time*. *Computer vision* adalah teknik yang efektif, murah, mudah digunakan, dan tidak merusak tanaman. Sistem yang dihasilkan *computer vision* dapat dijalankan pada komputer,

sehingga mudah digunakan masyarakat untuk menemukan penyakit tanaman (Khaira et al., 2024).

Pada era sekarang ini, bidang agrikultur tidak hanya memanfaatkan kecerdasan buatan, yang telah mengalami kemajuan untuk meningkatkan efisiensi produksi dengan mengurangi dampak negatif lingkungan. Pemanfaatan teknologi dapat mengurangi biaya penyemprotan pestisida pada tanaman sebanyak 90%. Dalam bidang agrikultur, kombinasi kecerdasan buatan dan *computer vision* telah menjadi sangat efektif, dan terkadang digunakan bersama dengan teknik big data untuk menyelesaikan masalah seperti mendeteksi penyakit tanaman. Kecerdasan buatan yang mengalami kemajuan secara signifikan terkhusus pada machine learning sehingga muncul *deep learning*. *You Only Look Once (Yolo)* adalah metode deteksi objek yang lebih baik dalam pendekatan *deep learning*, hal ini dikarenakan *Yolo* mengeluarkan fitur dari gambar dan menghitung *bounding box* dengan probabilitas kelas, kemudian memberikan hasil kelas objek dan informasi lokasi gambar (Ardiansyah & Hasan, 2023).

Pada penelitian ini metode kecerdasan buatan yang digunakan dalam pendekatan *computer vision* adalah berbasis python dan *YOLO (You Look Only Once)*. Python adalah salah satu bahasa pemrograman yang paling banyak digunakan oleh para *developer* dan perusahaan besar untuk membuat aplikasi desktop, web, dan *mobile*. Sedangkan *Yolo* adalah suatu perkembangan dari algoritma *Convolutional Neural Network (CNN)* yang dapat mendeteksi objek menggunakan model terpadu yang sudah terlatih, dimana jaringan konvolusional tunggal secara bersamaan memprediksi beberapa kotak pembatas dan probabilitas kelas di dalam kotak tersebut, dimana sistem pada algoritma *Yolo* dapat mendeteksi sebuah gambar ataupun video secara *real-time*, dan menghasilkan nilai yang akurat, sehingga *Yolo* dapat mendeteksi objek dengan tingkat keakurasian dan *frame rate* yang sangat tinggi.

Penelitian mengenai penyakit pada daun padi sudah banyak dilakukan sebelumnya. Pada penelitian yang dilakukan oleh (Putra Pranjaya et al., 2024) berjudul “Klasifikasi penyakit pada daun tanaman padi berbasis *Yolo V5 (You Only Look Once)*”, penelitian ini berhasil mengembangkan sistem secara *real-time* yang dapat membantu petani menemukan dan mengobati penyakit pada tanaman padi.

Pengujian secara *real-time* menunjukkan nilai mAP tertinggi 95%, akurasi 88%, dan *recall* 100%. Untuk kelas *narrow brown spot* nilai akurasi adalah 93%, kelas hawar daun akurasi 81%, dan kelas daun padi normal akurasi 91%.

Penelitian yang dilakukan oleh (Deden Miftah Fauzi et al., 2024) berjudul “Deteksi jenis penyakit pada tanaman padi menggunakan *Yolo V5*”, penelitian ini menunjukkan performa yang sangat baik, dengan matriks kinerja yaitu, *precision* 92,5%, *recall* 90,8%, *F1-score* 91,6%, dan mAP 91,6%. Penelitian oleh (Ardiansyah & Hasan, 2023) berjudul “Deteksi dan Klasifikasi Penyakit pada Daun Kopi Menggunakan *Yolo v7*”, penelitian ini menunjukkan bahwa metode *Yolo v7* efektif dalam mendeteksi dan mengklasifikasikan penyakit pada daun tanaman kopi. Hasilnya menunjukkan akurasi yang tinggi (*precision* 0,926 dan *F1-score* 0,93) dan kinerja yang baik pada kelas binary (*F1-score* 0,99, *precision* 0,991, dan *recall* 1).

Penelitian lain oleh (Anwar et al., 2023) berjudul “Klasifikasi Penyakit Tanaman Cabai Rawit Dilengkapi Dengan Segmentasi Citra Daun Dan Dan Buah Menggunakan *Yolo V7*”, penelitian ini menunjukkan bahwa akurasi segmentasi *Yolo v7* 0,970 pada mAP50 untuk segmentasi daun dan buah tanaman cabai. Untuk model DCNN dengan *transfer learning* yang didasarkan pada model *EfficientNetV2M*, akurasi klasifikasi penyakit daun adalah 0,912, dan akurasi klasifikasi buah cabai 0,889. Penelitian lainnya dilakukan oleh (Pokhrel, 2024) berjudul “Penyakit Pada Buah Kakao Menggunakan *Yolo v8* Berbasis Website”, penelitian ini berhasil mengembangkan model *Yolov8* untuk mendeteksi hama dan penyakit buah kakao dan mencapai akurasi yang sangat baik, yaitu 94%. *F1-score* tertinggi mencapai 0,93 pada nilai *confidence* 0,455, menunjukkan keseimbangan yang baik antara presisi dan *recall*. Nilai presisi mencapai rata-rata maksimal 1,00 pada nilai *confidence* 0,973, menunjukkan tingkat presisi yang sangat tinggi. Nilai *recall* mencapai rata-rata maksimal 0,94 pada nilai *confidence* 0,00.

Berdasarkan permasalahan pada produksi tanaman padi yang menurun akibat penyakit tanaman padi, penulis mengangkat sebuah penelitian yang berjudul “Pengembangan Aplikasi Deteksi Penyakit pada Daun Padi Menggunakan *Computer vision* Berbasis Python dan *Yolo*”. Penelitian ini bertujuan untuk mengembangkan aplikasi yang dapat mendeteksi penyakit pada daun padi secara

otomatis menggunakan teknologi *computer vision* yang diharapkan dapat membantu membantu petani meningkatkan produktivitas pertanian dengan mengurangi kesalahan diagnosis penyakit dan menyediakan penanganan penyakit tanaman padi yang tepat waktu.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang masalah yang sudah di uraikan di atas, maka rumusan masalah dalam penelitian ini adalah:

1. Bagaimana Pengembangan Aplikasi Deteksi Penyakit pada Daun Padi Menggunakan *Computer vision* Berbasis Python dan *Yolo*?
2. Bagaimana mengaplikasikan *Computer vision* berbasis Python dan *Yolo* yang terdiri dari beberapa tahapan yaitu *Problem Scoping*, *Data Acquisition*, *Data Exploration*, *Modelling*, *Evaluation*, dan *Deployment*?
3. Bagaimana Aplikasi Deteksi Penyakit pada Daun Padi Menggunakan *Computer vision* Berbasis Python dan *Yolo* dapat membantu dalam mengidentifikasi penyakit tanaman padi secara mandiri?

## 1.3. Hipotesis

Berdasarkan rumusan masalah yang diuraikan di atas, maka hipotesis dalam penelitian ini adalah:

1. Pengembangan Aplikasi Deteksi Penyakit pada Daun Padi Menggunakan *Computer vision* Berbasis Python dan *Yolo* dapat menghasilkan alat yang efektif dan efisien dalam mengenali penyakit tanaman padi.
2. Penerapan tahapan-tahapan seperti *problem scoping*, *data acquisition*, *data exploration*, *modelling*, *evaluation*, dan *deployment* pada aplikasi berbasis python dan *Yolo* dapat meningkatkan akurasi dalam mendeteksi penyakit pada daun padi.
3. Aplikasi Deteksi Penyakit pada Daun Padi Menggunakan *Computer vision* Berbasis Python dan *Yolo*, mampu membantu pengguna dalam mengidentifikasi penyakit padi secara mandiri, meningkatkan pemahaman petani, dan memungkinkan tindakan pencegahan lebih dini.

#### 1.4. Batasan Masalah

Berdasarkan rumusan masalah dan hipotesis yang penulis dapatkan, maka batasan masalah dalam penelitian ini dibatasi pada Pengembangan Aplikasi Deteksi Penyakit pada Daun Padi Menggunakan *Computer vision* Berbasis Python dan *Yolo*.

#### 1.5. Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan penelitian dalam penelitian ini adalah:

1. Mengembangkan Aplikasi Deteksi Penyakit pada Daun Padi Menggunakan *Computer vision* Berbasis Python dan *Yolo*.
2. Menerapkan tahapan-tahapan seperti *problem scoping*, *data acquisition*, *data exploration*, *modelling*, *evaluation*, dan *deployment* dalam pengembangan aplikasi deteksi penyakit pada daun padi untuk memastikan efisiensi dan akurasi aplikasi.
3. Membantu pengguna meningkatkan produktivitas pertanian, meningkatkan kesadaran, dan kemampuan pengguna untuk mengidentifikasi penyakit tanaman padi secara mandiri.

#### 1.6. Manfaat Penelitian

Dengan melakukan penulisan ini, penulis dapat menguraikan manfaat-manfaat sebagai berikut:

1. Penulis

Mengaplikasikan ilmu-ilmu akademis yang didapat selama perkuliahan ke dalam Pengembangan Aplikasi Deteksi Penyakit pada Daun Padi Menggunakan *Computer vision* Berbasis Python dan *Yolo*. Penulis dapat mengembangkan kreatifitas dalam membuat aplikasi yang dapat digunakan untuk menyelesaikan permasalahan, terutama yang berkaitan dengan pendeteksian atau sebagai bahan penelitian yang digunakan oleh banyak orang.

2. Masyarakat

Merancang aplikasi yang mendukung mobilitas pengguna yang akurat dalam mendeteksi penyakit pada tanaman padi. Pengguna dapat mengetahui secara cepat mengenai penyakit yang menyerang tanaman padi.

3. Universitas

Menambah literatur di bidang teknologi pertanian, khususnya mengenai penerapan *computer vision* untuk deteksi penyakit pada tanaman.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Kecerdasan Buatan**

Kecerdasan buatan juga dikenal sebagai *Artificial intelligence* pertama kali muncul pada tahun 1950 (Kaul et al., 2020). Kecerdasan buatan adalah kemampuan untuk meningkatkan kecerdasan sebuah sistem melalui pengaturan yang sesuai dengan standar ilmiah (Esteva et al., 2021). Kecerdasan buatan adalah cabang ilmu komputer yang berfokus pada pembuatan sistem dan mesin yang dapat melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia. Kecerdasan buatan menggunakan data dan algoritma untuk memodelkan kemampuan seperti pemahaman bahasa, pengenalan wajah, pemecahan masalah, pengambilan Keputusan, dan pembelajaran dari data.

Menurut Sri Hartati (2021), berikut gambaran singkat tentang cabang keilmuan dari kecerdasan buatan, yaitu:

##### 1. Sistem Pakar (*Expert System*)

Sistem pakar didefinisikan sebagai perangkat lunak berpengetahuan intensif yang dapat melakukan beberapa tugas yang biasanya membutuhkan keahlian manusia.

##### 2. Penalaran Komputer (*Computer Reasoning*)

*Case-based reasoning* (CBR) adalah paradigma utama dalam penalaran otomatis dan pembelajaran mesin.

##### 3. Pembelajaran Mesin (*Machine Learning*)

Pembelajaran Mesin adalah teknologi yang memungkinkan sistem belajar langsung dari contoh, data, dan pengalaman.

##### 4. Jaringan Saraf Tiruan (*Artificial Neural Networks*)

Jaringan saraf tiruan adalah representasi dari otak manusia yang mensimulasikan proses belajar.

#### 5. Pemrosesan Bahasa Alami (*Natural Language Processing*)

Pemrosesan bahasa alami adalah saluran inteligensi buatan dan linguistic yang ditujukan untuk membuat komputer memahami pernyataan atau kata-kata yang ditulis dalam Bahasa manusia.

#### 6. Pengenalan Pola (*Pattern Recognition*)

Pengenalan pola adalah cabang keilmuan pengenalan pola yang menyediakan Solusi untuk berbagai masalah, mulai dari pengenalan ucapan, pengenalan wajah, hingga karakter tulisan tangan dan diagnosis medis.

#### 7. Indra Komputer (*Computer vision*)

Indra komputer bertujuan untuk menghasilkan model komputasi sistem visual manusia.

#### 8. Robotika

Robotika merupakan subjek tanpa batas yang tajam menyatu ke dalam bidang-bidang, seperti kecerdasan buatan, otomatisasi, dan *remote control*.

#### 9. Sistem Pendukung Keputusan (*Decision Support System*)

Sistem pendukung keputusan adalah sistem informasi berbasis komputer yang dirancang sedemikian rupa hingga membantu manajer memilih salah satu dari banyak Solusi alternatif untuk suatu masalah.

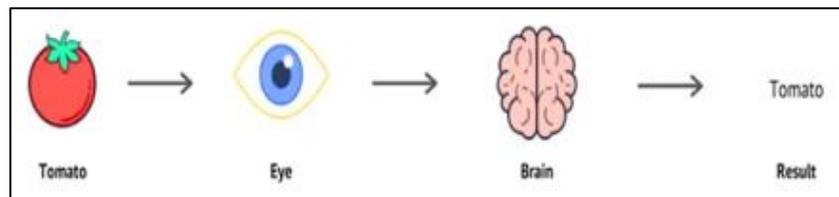
### 2.2. *Computer Vision*

*Computer vision* adalah komponen *artificial intelligence* yang meniru jaringan syaraf manusia, yaitu otak dan mata. Visualisasi mesin memiliki berbagai tingkatan tugas, tugas terendah termasuk mengenali tepi pada gambar dan tugas Tingkat tinggi termasuk memahami adegan keseluruhan untuk mendeteksi objek dalam gambar atau video secara *real-time* (Anugrah Pramesta et al., 2022). Tiga komponen utama telah berkontribusi secara signifikan pada kemajuan dalam bidang *computer vision*, yaitu:

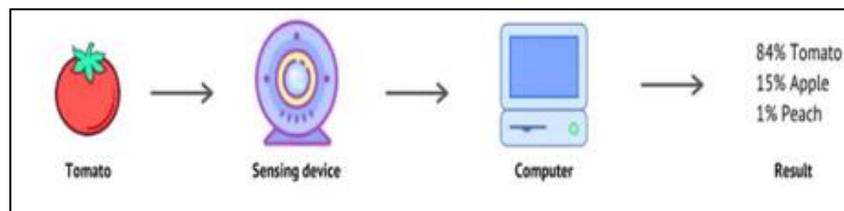
1. Matangnya teknik *deep learning*, yang merupakan jenis pembelajaran mesin yang memungkinkan pembelajaran langsung dari fungsi yang sangat kompleks menggunakan data mentah.
2. Kemajuan dalam komputasi lokal yang kuat melalui penggunaan *GPU*.

3. Ketersediaan kumpulan data berlabel yang dapat diakses secara publik, yang dapat digunakan untuk melatih algoritma ini.

*Computer vision* berfungsi untuk membuat keputusan tentang objek fisik nyata yang didapat dari perangkat atau sensor, dan *computer vision* membuat komputer dapat mengenali gambar layaknya manusia (Susim & Darujati, 2021). Gambar berikut menunjukkan analogi persamaan antara jaringan penglihatan manusia dan *computer vision*, yaitu:



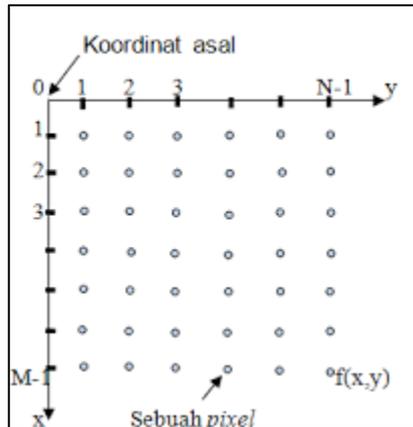
**Gambar 2. 1** Analogi Jaringan Penglihatan Manusia



**Gambar 2. 2** Analogi *Computer Vision*

### 2.3. Pengolahan Citra

Pengolahan citra adalah suatu teknologi untuk menyelesaikan masalah pemrosesan gambar (Susim & Darujati, 2021). Pengolahan citra adalah metode pengolahan gambar yang meningkatkan kualitas gambar sehingga lebih mudah diinterpretasikan oleh manusia atau mesin komputer berupa foto ataupun gambar bergerak (Rilo Pambudi et al., 2020). Pengolahan citra diwakili sebagai bit dalam array yang memiliki nilai *real* dan kompleks.



**Gambar 2. 3** Koordinasi Citra digital

Pada gambar diatas, kita dapat melihat bahwa sebuah citra adalah matriks yang terdiri dari M sebagai kolom dan N sebagai baris, dan ada pixel yang merupakan elemen terkecil dari citra.

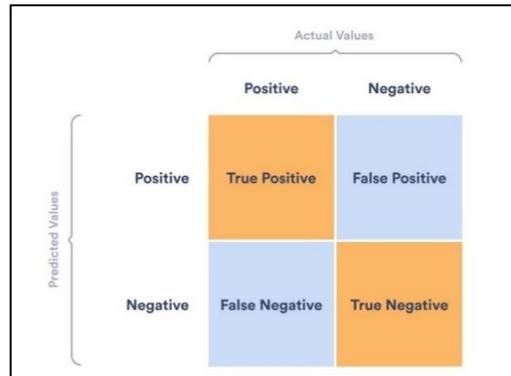
#### 2.4. *Yolo*

*You Only Look Once (Yolo)* adalah metode deteksi objek yang lebih baik dalam pendekatan *deep learning*, hal ini dikarenakan *Yolo* mengeluarkan fitur dari gambar dan menghitung *bounding box* dengan probabilitas kelas, kemudian memberikan hasil kelas objek dan informasi lokasi gambar (Ardiansyah & Hasan, 2023).

Untuk mengetahui apakah dataset yang sudah dilatih menghasilkan model yang baik, *Yolo* memiliki beberapa parameter yang dapat digunakan untuk mengetahui apakah modelnya baik atau tidak. Parameter tersebut adalah:

1. *Confusion matrix*

*Confusion matrix* adalah sebuah tabel yang digunakan untuk mengevaluasi kinerja model dalam klasifikasi atau deteksi. Saat mendeteksi objek dengan *Yolo*, *confusion matrix* dapat digunakan untuk menunjukkan dan menganalisis seberapa baik model *Yolo* dalam mengklasifikasikan objek dan menemukan apakah ada kesalahan dalam proses deteksi.



**Gambar 2. 4** *Confusion matrix*

- a. *True Positives* (TP) adalah jumlah objek yang benar terdeteksi oleh model.
- b. *True Negatives* (TN) adalah jumlah objek yang tidak ada dan tidak terdeteksi dengan benar oleh model.
- c. *False Positives* (FP) adalah jumlah objek yang salah dianggap sebagai objek oleh model.
- d. *False Negatives* (FN) adalah jumlah objek yang tidak terdeteksi oleh model padahal seharusnya terdeteksi.

*Confusion matrix* membantu dalam menghitung matrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score*. Namun, nilai spesifik *confusion matrix* *Yolo* akan bergantung pada implementasi dan data yang digunakan dalam proses deteksi objek.

## 2. *Accuracy*

*Accuracy* adalah matriks yang digunakan untuk mengukur sistem secara keseluruhan dapat melakukan klasifikasi dengan benar. Untuk mengetahui nilai akurasi, persamaan berikut dapat digunakan:

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

### 3. *Precision*

*Precision* menunjukkan seberapa akurat model *Yolo* dalam mengklasifikasikan objek sebagai positif. Semakin tinggi nilai *precision*, semakin sedikit objek yang terdeteksi salah sebagai positif atau *false positive*. Untuk mengetahui nilai *precision* persamaan berikut dapat digunakan:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

### 4. *Recall*

*Recall* menunjukkan seberapa baik model *Yolo* menemukan objek yang sebenarnya positif. Semakin tinggi nilai *recall*, semakin sedikit objek yang terlewat atau *false negative*. Untuk mengetahui nilai *recall* persamaan berikut dapat digunakan:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

### 5. *F1-score*

*F1-score* sangat membantu ketika mengevaluasi model secara menyeluruh, terutama ketika ada kelas yang tidak seimbang dalam dataset. *F1-score* memiliki rentang nilai 0 hingga 1 dan menunjukkan performa yang sempurna dengan presisi dan *recall* yang optimal. Untuk mengetahui nilai *F1-score* persamaan berikut dapat digunakan:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

### 6. *Mean Average Precision (mAP)*

mAP menghitung presisi rata-rata untuk setiap kelas objek yang ada dalam dataset, lalu mengambil rata-rata tersebut. Presisi dihitung pada setiap level ambang batas (*threshold*) yang berbeda untuk menghasilkan kurva presisi-panggilan (*precision-recall*). Untuk menghitung mAP, kurva dihitung dengan

mengambil area rata-rata di bawah kurva (*Area Under the Curve* atau AUC). Untuk mengetahui nilai mAP persamaan berikut dapat digunakan:

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \quad (5)$$

## 2.5. Python

Python adalah Bahasa pemrograman tingkat tinggi yang di ciptakan oleh Guido Van Rossum dan dirilis tahun 1991. Python merupakan Bahasa yang sangat populer belakangan ini. Selain itu, python adalah Bahasa pemrograman yang dapat digunakan untuk berbagai tujuan, seperti *machine learning* dan *deep learning* (Alfarizi et al., 2023). Python adalah bahasa pemrograman berorientasi objek yang dapat digunakan untuk membuat perangkat lunak dan berjalan di berbagai sistem operasi. Python saat ini mejadi bahasa yang populer di bidang ilmu data dan analisis karena dukungannya terhadap repository. Repository ini menyediakan fungsi analisis data dan pembelajarn mesin, data *preprocessing tools*, dan visualisasi data.(Koretsky, 2023).



**Gambar 2. 5** *Logo Python*

## 2.6. Flask

*Flask* ringan digunakan karena termasuk dalam jenis *microframework* karena tidak memerlukan alat atau Pustaka khusus untuk digunakan (Ngantung & Pakereng, 2021). *Flask* adalah kerangka kerja (*framework*) yang sangat terkenal dan mudah digunakan untuk membuat aplikasi web menggunakan bahasa pemrograman python. Dikenal karena fleksibilitasnya yang tinggi, *flask* telah

menjadi pilihan populer bagi pengembang dalam mengembangkan aplikasi website.

### **2.7. Bootstrap**

*Bootstrap* adalah alat bantu untuk membuat tampilan website yang mempercepat pekerjaan pengembang dan pendesain halaman website (Widayati et al., 2021). Sedangkan menurut (I Kadek Juni Arta & Nyoman Bagus Suweta Nugraha, 2020), *bootstrap* adalah salah satu *framework html, css, dan javascript* yang paling populer digunakan oleh web *developer* untuk mengembangkan website *responsive* yang dirancang khusus untuk bagian pengembangan *front-end* website.

### **2.8. Roboflow**

*Roboflow* adalah platform web yang berfungsi untuk berhubungan dengan kumpulan dataset dan merupakan kerangka pengembang *computer vision* yang menawarkan peningkatan pemrosesan dan teknik pelatihan model untuk pengumpulan data. Dengan *roboflow*, dapat membagikan dataset sekaligus pemrosesannya, menggunakan *bounding box* untuk menandai atau *annotate objek* yang akan ditemukan, dan melakukan *preprocessing* seperti *grayscale*, dan *augmentasi* (Hayati et al., 2023).

### **2.9. Pertanian**

Pertanian adalah kegiatan pemanfaatan sumber daya hayati oleh manusia untuk menghasilkan makanan, bahan baku industri, atau energi, serta untuk mengelola lingkungan hidupnya. Sektor pertanian dapat diandalkan dalam pemuliahn ekonomi nasional karena telah terbukti terus berkontribusi pada perekonomian nasional meskipun badai krisis telah melanda.

### **2.10. Padi**

Padi (*oriza sativa*) adalah salah satu produksi pangan setelah jagung dan gandum yang menempai urutan ketiga sebagai jenis biji-bijian terbaik (Khoiruddin et al., 2022). Padi adalah tanaman yang menghasilkan beras, bahan makanan pokok bagi Sebagian besar penduduk di berbagai negara, terutama di Asia. Padi merupakan tanaman yang termasuk dalam keluarga rumput-rumputan (*poaceae*)

dan tumbuh dengan baik di lingkungan tropis dan subtropic. Padi biasanya di tanam di sawah yang tergenang air, tetapi ada juga varietas yang dapat tumbuh di ladang kering. Padi melalui berbagai fase dalam siklus hidupnya, mulai dari fase benih, fase vegetatif (pertumbuhan daun dan batang), fase reproduktif (pembentukan bulir padi), dan fase pematangan (biji padi siap panen). Pertanian padi membutuhkan perawatan yang intensif, termasuk pengendalian hama dan penyakit.

Menurut (Deden Miftah Fauzi et al., 2024) penyakit *brown spot*, *bacterial blight*, dan *blast* adalah salah satu penyakit padi yang paling meresahkan petani.

#### 1. *Brown spot*

*Brown spot* atau yang lebih di kenal dengan nama bercak coklat adalah penyakit yang disebabkan oleh jamur *bipolaris oryzae*, yang dapat mempengaruhi tanaman padi.

#### 2. *Bacterial Blight*

*Bacterial Blight* adalah penyakit yang disebabkan oleh bakteri *xanthomonas oryzae pv. Oryzae (xoo)* yang menyerang tanaman padi.

#### 3. *Blast*

*Blast* adalah penyakit yang disebabkan oleh jamur *pyricularia grisea* atau *pyricularia oryzae*. Penyakit ini menyerang bagian daun dan leher malai padi.

Menurut (Putri Ayuni et al., 2023) penyakit daun yang paling sering meyerang tanaman padi adalah *blast*, *brown spot*, *leaf smut*, dan penyakit lainnya.

#### 1. *Leaf Smut*

*Leaf smut* adalah penyakit yang disebabkan oleh jamur *Exserohilum turcicum*. Penyakit ini menyerang bagian daun padi dan menyebabkan kerusakan pada tanaman padi.

Pada penelitian ini, jenis penyakit daun yang akan digunakan adalah *blast*, *blight*, *brown spot*, *healthy*, dan *tungro*.

#### 1. *Healthy*

*Healthy* adalah daun padi dalam kondisi sehat, tanpa gejala penyakit atau kerusakan. Daun yang sehat biasanya berwarna hijau cerah, tidak ada bitnik-bintik,

bercak, atau perubahan warna yang mengindikasikan adanya serangan hama atau penyakit.

## 2. *Tungro*

*Tungro* adalah penyakit yang menyerang tanaman padi yang disebabkan oleh *rice tungro baciliform virus* (RTBV) dan *rice tungro spherical virus* (RTSV), yang ditularkan oleh wereng hijau (*nephotettix virescens*).

### 2.11. *State Of The Art*

Pada *State of the art*, penulis mencari referensi yang akan mendukung penelitian ini. Referensi yang digunakan adalah referensi yang paling relevan dengan penelitian ini. Studi literatur yang digunakan sebagai acuan dapat dilihat pada tabel 2.1.

**Tabel 2. 1** *State Of The Art*

No	Judul Artikel	Tahun	Metode	Hasil
1	Deteksi jenis penyakit pada tanaman padi menggunakan <i>yolo v5</i>	2024	<i>Yolo v5</i>	Hasil penelitian menunjukkan performa yang sangat baik, dengan matriks kinerja yaitu, <i>precision</i> 92,5%, <i>recall</i> 90,8%, <i>f1-score</i> 91,6%, dan <i>map</i> 91,6%.
2	Deteksi dan tracking objek secara <i>real-time</i> berbasis <i>computer vision</i> menggunakan metode <i>yolo v3</i>	2024	<i>Yolo v3</i>	Hasil penelitian menunjukkan bahwa sistem dapat mendeteksi banyak item dalam satu gambar dengan waktu kurang dari 3 detik dan jangkauan deteksi hingga 100 cm saat objek dekat dengan kamera dan 150 cd/m <sup>2</sup> saat objek berada dalam cahaya paling terang.
3	Perancangan sistem pendeteksi kualitas tanaman tomat berdasarkan daun menggunakan <i>computer vision</i> dengan <i>yolo</i> dan <i>opencv</i>	2024	<i>Yolo</i> dan <i>opencv</i>	Hasil penelitian ini menunjukkan bahwa menggunakan metode <i>yolo</i> dan <i>opencv</i> sangat membantu dalam pengembangan sistem prediksi

				yang dapat mendeteksi kualitas daun tomat dengan cepat, efektif, dan akurat.
4	Deteksi citra daun untuk klasifikasi penyakit daun padi menggunakan pendekatan <i>deep learning</i> dengan model <i>cnn</i>	2024	<i>Convolutional neural network</i> (CNN)	Hasil penelitian menunjukkan bahwa skenario kedua dengan 50 <i>epoch</i> memiliki akurasi pelatihan sebesar 99,05% dan akurasi <i>validasi</i> sebesar 80%, dengan nilai <i>recall</i> , <i>precision</i> , dan <i>f1-score</i> yang sangat baik, yaitu 1,00. Ini menunjukkan bahwa model sangat efektif dalam mengklasifikasikan penyakit pada daun padi.
5	Rancang bangun aplikasi deteksi alat pelindung diri (apd) untuk pekerja proyek dengan menggunakan algoritma <i>yolo v5</i>	2024	<i>Yolo v5</i>	Hasil penelitian ini menunjukkan setelah melakukan pelatihan dataset sebanyak 100, aplikasi ini mencapai tingkat akurasi yang sangat memuaskan. Akurasi helm dan rompi 96%, masker 95%, dan sepatu 92%.
6	Klasifikasi penyakit pada daun tanaman padi berbasis <i>yolo v5</i> ( <i>you only look once</i> )	2023	<i>Yolo v5</i>	Hasil penelitian menunjukkan bahwa penelitian ini berhasil mengembangkan sistem secara <i>real-time</i> yang dapat membantu petani menemukan dan mengobati penyakit pada tanaman padi. Pengujian secara <i>real-time</i> menunjukkan nilai <i>map</i> tertinggi

				95%, akurasi 88%, dan <i>recall</i> 100%. Untuk kelas <i>narrow brown spot</i> nilai akurasi adalah 93%, kelas hawar daun akurasi 81%, dan kelas daun padi normal akurasi 91%.
7	Klasifikasi penyakit tanaman cabai rawit dilengkapi dengan segmentasi citra daun dan buah menggunakan <i>yolo v7</i>	2023	<i>Yolo v7</i>	Hasil penelitian menunjukkan bahwa akurasi segmentasi <i>yolo v7</i> 0,970 pada map50 untuk segmentasi daun dan buah tanaman cabai. Untuk model dcnn dengan <i>transfer learning</i> yang didasarkan pada model <i>efficientnetv2m</i> , akurasi klasifikasi penyakit daun adalah 0,912, dan akurasi klasifikasi buah cabai 0,889.
8	Deteksi dan klasifikasi penyakit pada daun kopi menggunakan <i>yolo v7</i>	2023	<i>Yolo v7</i>	Hasil penelitian menunjukkan bahwa metode <i>yolo v7</i> efektif dalam mendeteksi dan mengklasifikasikan penyakit pada daun tanaman kopi. Hasilnya menunjukkan akurasi yang tinggi ( <i>precision</i> 0,926 dan <i>f1-score</i> 0,93) dan kinerja yang baik pada kelas binary ( <i>f1-score</i> 0,99, <i>precision</i> 0,991, dan <i>recall</i> 1).

9	Implementasi <i>raspberry pi</i> dalam alat klasifikasi penyakit mata dengan arsitektur <i>yolo v8</i> menggunakan <i>oftalmoskop</i>	2023	<i>Yolo v8</i>	Hasil penelitian ini menunjukkan bahwa pada empat percobaan, model klasifikasi penyakit mata yang menggunakan arsitektur <i>yolo v8</i> dengan <i>raspberry pi</i> dan kamera webcam menghasikan akurasi yang berbeda. Secara keseluruhan, model yang telah dilatih menunjukkan kinerja yang baik dalam mengklasifikasikan penyakit mata, dengan klasifikasi katarak paling akurat mencapai 100%, <i>glaukoma</i> 99%, normal 77%, dan <i>diabetic retinopathy</i> 78%.
10	Klasifikasi penyakit padi menggunakan algoritma <i>cnn (convolutional neural network)</i>	2022	Algoritma CNN ( <i>convolutional neural network</i> )	Hasil penelitian menunjukkan bahwa metode <i>cnn</i> efektif dalam menemukan gambar daun padi yang berpeyakit dengan akurasi tertinggi untuk data <i>training</i> mencapai 85%, data <i>testing</i> 86%, dan data <i>validation</i> 95%.
11	Implementasi <i>deep learning</i> pada sistem klasifikasi hama tanaman padi menggunakan metode <i>convolutional neural network (cnn)</i>	2022	CNN ( <i>convolutional neural network</i> )	Hasil penelitian menunjukkan bahwa akurasi pelatihan masing-masing model adalah 83,02%, 78,30%, dan 81,13% dan akurasi pengujian adalah 69,33% hingga 77,33%. Ini menunjukkan bahwa

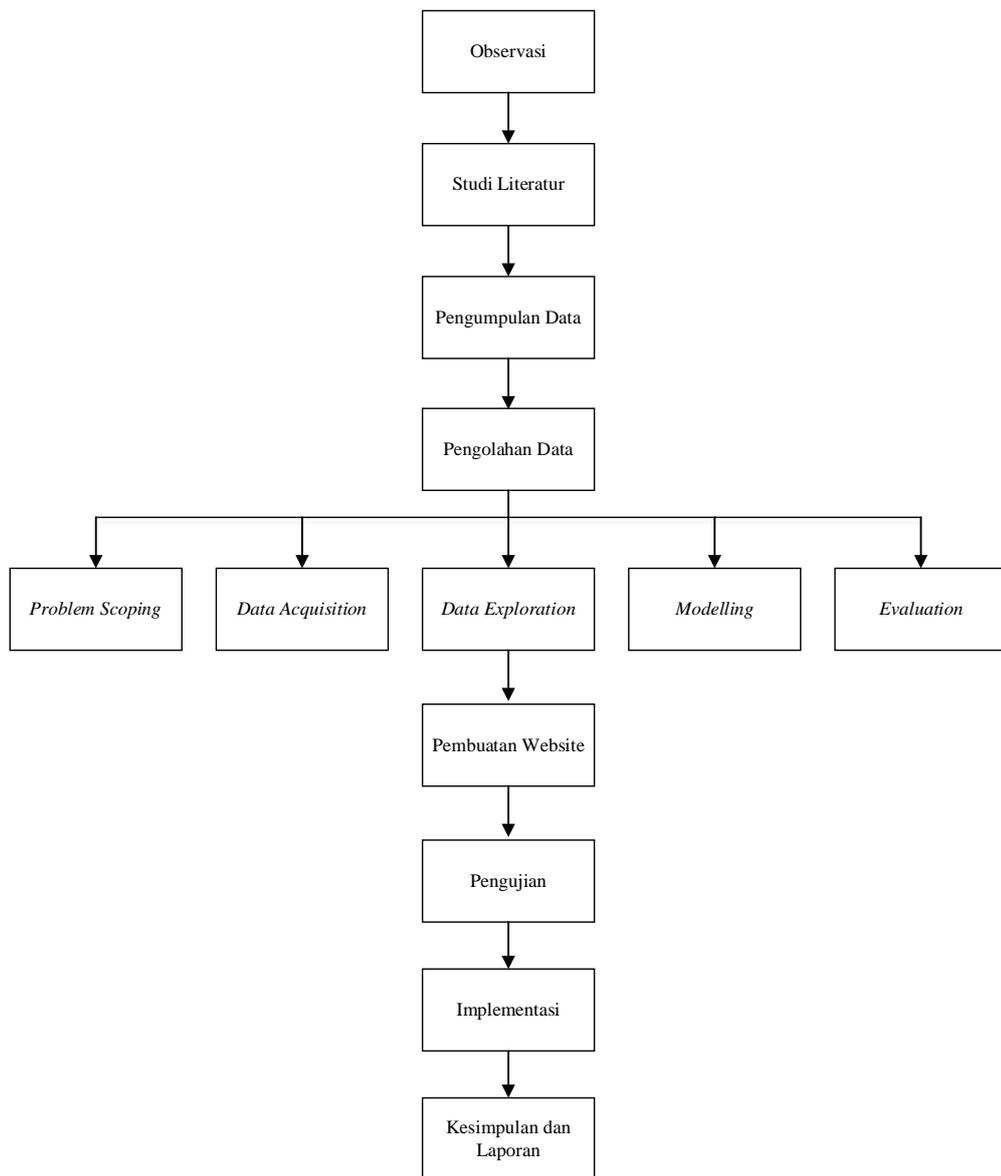
				cnn efektif dalam mengklasifikasikan hama pada tanaman padi.
12	Klasifikasi penyakit daun padi menggunakan <i>convolutional neural network</i>	2022	<i>Convolutional neural network</i> (CNN)	Hasil penelitian menunjukkan bahwa metode cnn dapat mengidentifikasi penyakit daun padi dengan akurasi 98%.
13	Klasifikasi penyakit daun padi menggunakan <i>random forest</i> dan <i>color histogram</i>	2022	Algoritma <i>random forest</i> dan ekstraksi fitur <i>color histogram</i>	Hasil penelitian menunjukkan bahwa algoritma random forest dan ekstraksi fitur histogram warna sangat efektif. Metode ini mencapai akurasi tertinggi 99,65%.
14	Sistem cerdas penghitung jumlah mobil untuk ketersediaan lahan parkir berbasis python dan <i>yolo v4</i>	2021	<i>Yolo v4</i>	Hasil penelitian menunjukkan bahwa sistem berhasil mendeteksi objek mobil dengan akurasi keseluruhan 72,8%.
15	Deteksi dan pengenalan objek dengan model <i>machine learning</i> : model <i>yolo</i>	2021	<i>Literatur review</i> (membandingkan berbagai jurnal ilmiah terkait algoritma <i>yolo</i> )	Hasil penelitian ini menunjukkan bahwa <i>yolo</i> memiliki kecepatan pemrosesan dan akurasi yang tinggi, tetapi masih ada kekurangan. Pada <i>yolo v3</i> , kesulitan menemukan objek berukuran medium, dan tidak ada penjelasan saintifik resmi untuk <i>yolo v5</i> .

16	Aplikasi penghitung jarak dan jumlah orang berbasis <i>yolo</i> sebagai protokol kesehatan covid-19	2021	<i>Yolo v3</i>	Hasil penelitian menunjukkan bahwa aplikasi dapat mendeteksi jumlah dan jarak objek manusia <i>epoch</i> dalam gambar digital. Uji coba yang menggunakan lima dataset pejalan kaki menunjukkan akurasi 89,99% untuk gambar pertama, 76,92% gambar kedua, 89,99% gambar ketiga, dan 100% untuk gambar keempat dan kelima. Aplikasi ini menunjukkan kinerja yang baik dalam menilai <i>social distancing</i> dengan rata-rata akurasi 90,04%.
----	---	------	----------------	---

## BAB III METODOLOGI PENELITIAN

### 3.1. Kerangka Penelitian

Kerangka penelitian yang digunakan dalam penelitian ini dapat dilihat pada gambar 3.1 berikut ini.



**Gambar 3. 1** Kerangka Penelitian

## **3.2. Tahapan Penelitian**

Pada tahapan penelitian ini akan dijelaskan semua proses yang akan dilakukan. Tahapan-tahapan penelitian yang dilakukan adalah sebagai berikut:

### **3.2.1. Observasi**

Pada penelitian ini, observasi dilakukan secara online melalui situs web *Roboflow*, yang menyediakan dataset dan alat khusus untuk pengembangan aplikasi *computer vision*. Pada situs web ini, peneliti mengakses data yang relevan dan mempelajari karakteristik dan struktur dataset yang akan digunakan dalam penelitian ini. Tanpa perlu mengumpulkan data secara langsung di lapangan, metode ini memungkinkan peneliti untuk melihat dan memahami data secara menyeluruh.

### **3.2.2. Studi Literatur**

Pada penelitian ini, dipelajari beberapa literatur atau penelitian terdahulu yang diperkirakan dapat digunakan dalam penunjang penelitian ini. Kemudian literatur-literatur tersebut dipilih dan diseleksi untuk digunakan dalam penelitian ini. Literatur diambil dari berbagai sumber, yaitu jurnal ilmiah dan artikel tentang Pengembangan Aplikasi Deteksi Penyakit Pada Daun Padi Menggunakan *Computer vision* Berbasis Python dan *Yolo*, serta bahan bacaan lain yang mendukung.

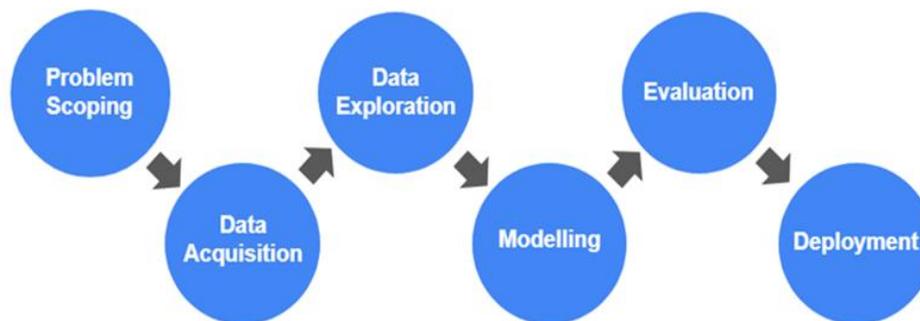
### **3.2.3. Pengumpulan Data**

Pada penelitian ini, pengumpulan data dilakukan dengan memanfaatkan dataset yang disediakan oleh *platform roboflow*. *Roboflow* menyediakan dataset gambar yang telah dilabeli sesuai dengan kebutuhan penelitian, contohnya gambar daun padi yang menunjukkan berbagai kondisi penyakit. Melalui *roboflow* peneliti mendapatkan dataset yang relevan, kemudian diolah dalam proses pengembangan model deteksi penyakit pada daun padi.

### **3.2.4. Pengolahan Data**

Pada penelitian ini, *Ai project cycle* akan digunakan. Siklus ini mencakup serangkaian tahapan yang dilalui dalam mengembangkan dan menerapkan solusi

berbasis kecerdasan buatan (AI), mulai dari pemahaman masalah, pengumpulan dan persiapan data, pelatihan model, evaluasi, implementasi, pemeliharaan dan peningkatan model.



**Gambar 3. 2** *Ai Project Cycle*

a. *Problem Scoping*

*Problem scoping* adalah tahap awal siklus proyek AI dimana masalah yang ingin diselesaikan didefinisikan dan dipahami secara menyeluruh. Tahap ini sangat penting untuk menentukan arah dan ruang lingkup proyek secara keseluruhan.

b. *Data Acquisition*

*Data acquisition* adalah tahapan dalam siklus proyek AI, dimana data dikumpulkan harus relevan dan berkualitas tinggi sehingga dapat digunakan dalam proyek AI. *Data acquisition* adalah dasar dari siklus proyek AI karena kualitas dan kelengkapan data sangat mempengaruhi hasil akhir model AI yang dibuat. Metode pengumpulan data termasuk mengimpor data dari sistem internal, menggunakan API untuk menarik data dari platform lain, atau *scraping* data dari web. Dalam penelitian ini, data dikumpulkan melalui website *roboflow*. Data memiliki lima kelas yang berbeda yaitu *blast*, *blight*, *brown spot*, *healthy*, dan *tungro*. Pada gambar 3.3, menunjukkan contoh gambar dataset perkelas.

1. *Blast*2. *Blight*3. *Brown spot*4. *Healthy*5. *Tungro***Gambar 3.3** *Contoh Dataset*

c. *Data Exploration*

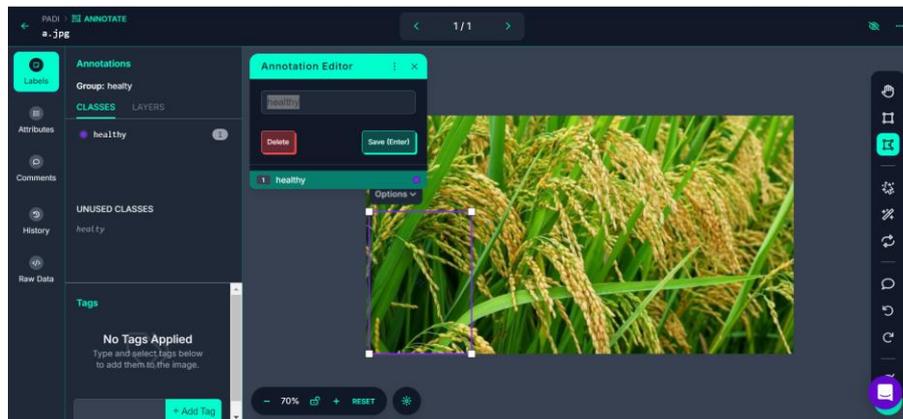
*Data exploration* pada siklus proyek AI adalah proses yang bertujuan untuk memahami, menganalisis, dan memperoleh pengetahuan dari data yang telah dikumpulkan sebelum memulai proses pemodelan. Proses *data exploration* mencakup hal-hal seperti ukuran, kuantitas, dan akurasi.

a. *Preprocessing Data*

Tahap *preprocessing* data bertujuan supaya gambar dapat di proses lebih mudah untuk proses selanjutnya. Pada tahap ini memilih gambar yang sesuai untuk dijadikan dataset dan mengubah setiap ukuran citra menjadi 640 x 640 pixel.

b. *Labelling Data*

*Labelling data* dilakukan dengan cara memberi anotasi pada gambar-gambar tersebut dengan menggunakan *bounding box* dan untuk menentukan kelas pada gambar. Proses anotasi dilakukan menggunakan platform *roboflow*.



**Gambar 3. 4** *Bounding Box* Pada Gambar

c. *Split Data*

*Split data* yaitu membagi dataset menjadi tiga bagian, data pelatihan (*training*) dan data validasi (*validation*), dan data pengujian (*testing*). Data *training* dibagi menjadi 70 % dari keseluruhan dataset, data *validation* 20% dari keseluruhan dataset, dan data *testing* 10% dari keseluruhan dataset.

d. *Modelling*

*Modelling* adalah tahap siklus proyek AI yaitu tahap dimana model AI atau *machine learning* (ML) dibangun, dilatih, dan dievaluasi menggunakan data yang telah dikumpulkan dan diproses dalam siklus proyek AI. Tahap modelling merupakan bagian penting dari proyek AI karena digunakan untuk mempelajari pola dan hubungan dalam data dan membuat model untuk prediksi dan pengambilan keputusan.

e. *Evaluation*

*Evaluation* pada siklus proyek AI adalah tahapan di mana kinerja model dievaluasi secara menyeluruh. Tujuan evaluation adalah untuk mengukur seberapa baik model AI dapat memprediksi atau mengambil keputusan berdasarkan data yang diberikan.

#### f. *Deployment*

*Deployment* dalam siklus proyek AI adalah tahap di mana model AI atau solusi yang dikembangkan diterapkan dan diintegritaskan ke dalam lingkungan produksi atau aplikasi yang sebenarnya.

#### 3.2.5. Pembuatan *Website*

Pada penelitian ini, *website* akan dikembangkan menggunakan *framework flask* sebagai *backend* dan *bootstrap* untuk tampilan antarmuka yang responsif. *Flask* memungkinkan integrasi yang fleksibel antara berbagai komponen aplikasi, sementara *bootstrap* mempermudah pembuatan antarmuka yang digunakan dan diakses di berbagai perangkat. Selain itu, model *yolo* akan di implementasikan pada *website* ini untuk mendeteksi penyakit pada daun padi secara *real-time*. Model *yolo* akan digunakan untuk memproses gambar yang diunggah pengguna, mengidentifikasi jenis penyakit pada daun padi, dan menampilkan hasil deteksi secara langsung di halaman *website*.

#### 3.2.6. Pengujian

Pada penelitian ini, pengujian mencakup beberapa aspek utama, yaitu akurasi dan performa model *yolo* dalam mendeteksi penyakit pada daun padi, kinerja sistem dalam kecepatan pemrosesan gambar pada *website*, serta tampilan responsivitas antarmuka di berbagai perangkat melalui *bootstrap*. Selain itu, pengujian fungsi *upload* dan tampilan hasil deteksi untuk memastikan gambar yang diunggah dapat diproses dengan baik. Pengujian menyeluruh ini penting untuk memastikan aplikasi berfungsi optimal dan memberikan pengalaman yang baik bagi pengguna.

#### 3.2.7. Implementasi

Pada penelitian ini, implementasi aplikasi ini meliputi integrasi model *yolo* untuk mendeteksi penyakit daun padi, pengembangan antarmuka *website* menggunakan *flask* dan *bootstrap*, serta pengaturan alur kerja yang memungkinkan pengguna mengunggah gambar dan menerima hasil deteksi secara *real-time*. Model *yolo* dihubungkan langsung ke halaman *website* agar hasil deteksi dapat

ditampilkan secara otomatis dan interaktif, sehingga seluruh sistem siap digunakan untuk mendeteksi dan mengklasifikasikan penyakit daun padi dengan cepat dan akurat.

### 3.2.8. Kesimpulan dan Laporan

Tahapan ini merupakan langkah terakhir dalam rangkaian penelitian yang telah dilakukan, pada tahapan ini akan dilakukan tahapan penarikan kesimpulan dari hasil penelitian berdasarkan analisis data yang diperoleh selama proses penelitian. Pada tahap ini, penting untuk mengidentifikasi dan merangkum hasil-hasil kunci yang dapat menjelaskan pemenuhan tujuan penelitian serta memberikan pemahaman yang lebih mendalam mengenai data yang diteliti. Setelah proses penarikan kesimpulan selesai, langkah berikutnya adalah menyusun laporan akhir. Penyusunan laporan ini bertujuan untuk menyajikan informasi secara sistematis dan terstruktur, sehingga pembaca dapat dengan mudah memahami maksud, tujuan, dan kontribusi penelitian ini dalam bidang yang relevan. Laporan akhir juga berfungsi sebagai referensi bagi peneliti lain yang ingin melanjutkan atau mengembangkan studi serupa di masa depan.

### 3.3. Jadwal Penelitian

Penelitian ini di mulai pada Agustus 2024 sampai dengan 2025. Jadwal pelaksanaan penelitian ini dapat dilihat dalam tabel jadwal penelitian dibawah ini yaitu sebagai berikut:

**Tabel 3. 1** Jadwal Penelitian

Jenis Penelitian	Jadwal Penelitian							
	Agu	Sep	Okt	Nov	Des	Jan	Feb	Mar
Pengajuan Judul	✓							
Penyusunan Proposal Penelitian		✓	✓	✓				
Observasi		✓	✓					
Studi Literatur		✓	✓					

Pengumpulan Data		✓	✓					
Pengolahan Data		✓	✓	✓	✓			
Seminar Proposal dan Revisi				✓				
Pembuatan <i>Website</i>				✓	✓	✓		
Pengujian							✓	✓
Implementasi							✓	✓
Kesimpulan dan Laporan Akhir				✓	✓	✓	✓	✓
Sidang <i>Komprehensif</i>								✓

## **BAB IV**

### **ANALISA DAN PERANCANGAN**

#### **4.1. Analisa Data**

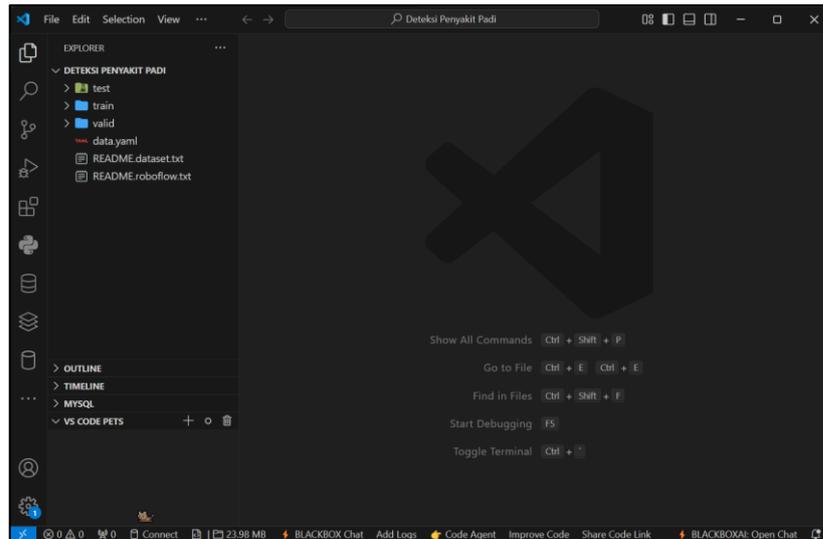
Pada penelitian ini, proses analisa data adalah proses mengelola dan mengevaluasi data untuk menemukan pola, hubungan, atau informasi penting. Tahapan analisa data meliputi *problem scoping*, *data acquisition*, *data exploration*, *modelling*, *evaluation*, dan *deployment*.

##### **4.1.1. Problem Scoping**

Pada tahap ini, peneliti melakukan observasi secara online untuk mengidentifikasi permasalahan dan penyebab utama rendahnya produktivitas padi. Observasi dilakukan melalui situs resmi Badan Pusat Statistik (BPS) tahun 2023. Setelah melakukan observasi, peneliti melakukan kajian terhadap studi literatur untuk menentukan analisis dan identifikasi terhadap permasalahan yang ditemukan, serta menyusun langkah-langkah penyelesaiannya yang dapat dilihat pada table 3.1.

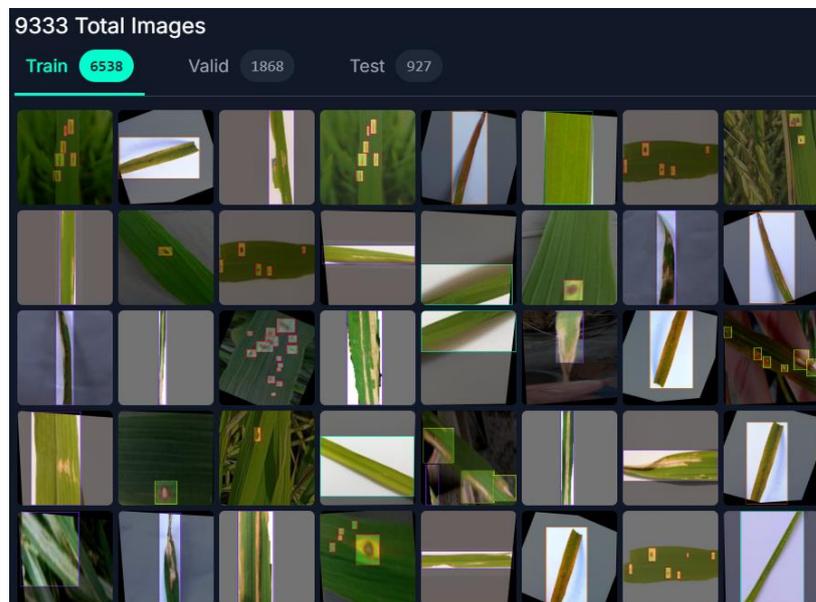
##### **4.1.2. Data Acquisition**

Pada tahap ini dilakukan pengumpulan dataset, yang didapat dari website *Roboflow* di <https://universe.roboflow.com/ta-bismillah-ozmeq/deteksi-penyakit-padi-8t5j7/dataset/13> dengan nama dataset “Deteksi Penyakit Padi *Computer vision* Project”. Total dataset yang didapat dari website ini berjumlah 9333 gambar, yang terdiri dari 3 *folder* yaitu *train* terdiri dari 6.538 gambar, *valid* terdiri dari 1.868 gambar dan *test* yang terdiri dari 927 gambar. Dengan total size sebesar 115 MB.



**Gambar 4. 1** Sub *Folder* Dataset

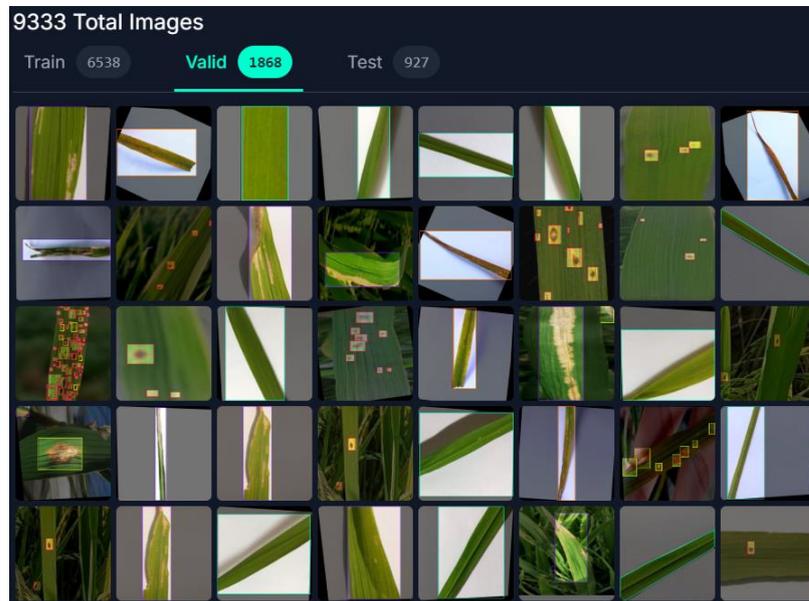
Sub *folder* dataset menyimpan dan mengorganisir data untuk pelatihan, *validasi*, dan pengujian model. Dataset biasanya dibagi ke dalam kategori atau label serta dilengkapi anotasi untuk deteksi objek. Selain itu, struktur ini mempermudah *preprocessing* dan *augmentasi data*.



**Gambar 4. 2** *Folder Train*

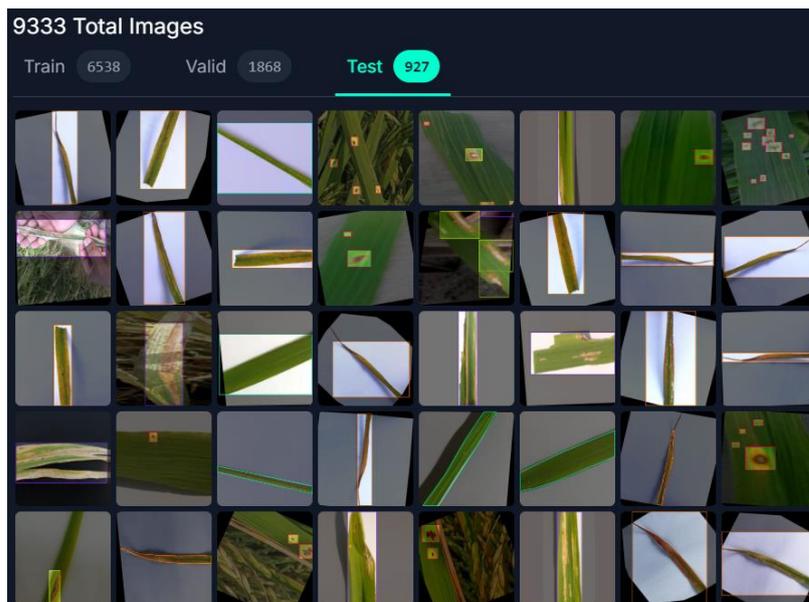
*Folder train* di Roboflow berisi data gambar yang digunakan untuk melatih model machine learning. Biasanya, *folder* ini berisi sebagian besar dataset

agar model dapat mempelajari pola dengan baik. Data dalam *folder train* juga sering melalui proses augmentasi untuk meningkatkan akurasi model.



**Gambar 4.3** *Folder Valid*

*Folder valid* di *Roboflow* berisi data untuk *validasi* model selama pelatihan. Data ini digunakan untuk mengevaluasi performa model sebelum pengujian akhir. *Folder* ini membantu mencegah *overfitting* dan memastikan model bekerja dengan baik pada data baru.



**Gambar 4.4** *Folder Test*

*Folder test* di *Roboflow* berisi data yang digunakan untuk menguji performa model setelah pelatihan. Data ini tidak digunakan selama *training* atau *validasi*, sehingga hasilnya mencerminkan kemampuan model pada data yang benar-benar baru. *Folder* ini penting untuk mengukur akurasi dan keandalan model sebelum diterapkan.

#### 4.1.3. *Data Exploration*

Pada tahap ini, dataset akan diolah terlebih dahulu. *Data exploration* pada siklus proyek AI adalah proses yang bertujuan untuk memahami, menganalisis, dan memperoleh pengetahuan dari data yang telah dikumpulkan sebelum memulai proses pemodelan. Proses *data exploration* mencakup hal-hal seperti ukuran, kuantitas, dan akurasi.

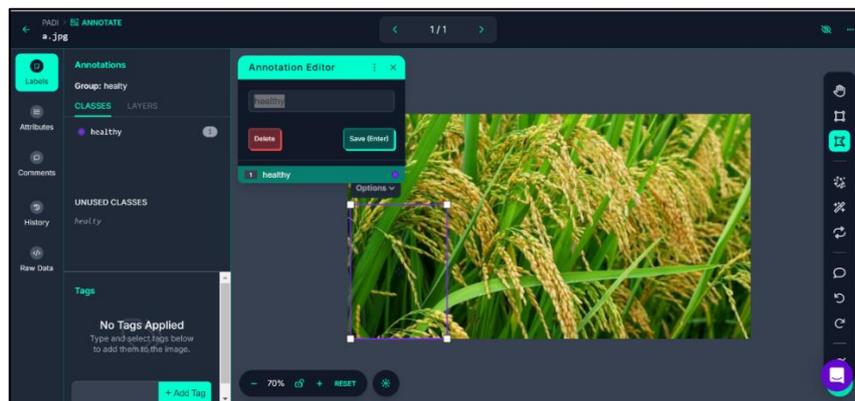
##### a. *Preprocessing Data*

Tahap *preprocessing data* bertujuan supaya gambar dapat di proses lebih mudah untuk proses selanjutnya. Pada tahap ini memilih gambar yang sesuai untuk dijadikan dataset dan mengubah setiap ukuran citra dalam pixel. Dalam penelitian ini, menggunakan dataset yang diperoleh dari website *Roboflow*. Kondisi data yang tersedia di *Roboflow* merupakan data yang sudah siap digunakan, sehingga sangat memudahkan proses penelitian. Gambar yang diberikan memiliki kualitas tinggi (HD) dan sudah melalui proses pelabelan sesuai dengan kategori masing-masing. Dengan demikian, tidak perlu melakukan pelabelan ulang atau pengolahan tambahan terhadap gambar tersebut.

Selain itu, dataset yang tersedia di *Roboflow* terdiri dari gambar asli, bukan hasil rekayasa atau manipulasi, sehingga keakuratan dan *validitas* data dapat terjamin. Hal ini sangat membantu dalam proses pelatihan model, karena tidak perlu melakukan *preprocessing data* secara manual, seperti penyesuaian resolusi, pemberian label, atau peningkatan kualitas gambar. Dengan data yang sudah terstruktur dan siap pakai dapat lebih fokus pada pengembangan model deteksi penyakit daun padi menggunakan *computer vision* berbasis Python dan *YOLO*.

b. *Labelling Data*

*Labelling data* dilakukan dengan cara memberi anotasi pada gambar-gambar tersebut dengan menggunakan *bounding box* dan untuk menentukan kelas pada gambar. Proses anotasi dilakukan menggunakan platform *roboflow*. Proses *bounding box* dilakukan per gambar nya, tetapi prosesnya tidak perlu dilakukan lagi karena pada website *roboflow* sudah disediakan gambar yang lengkap dengan *bounding box* nya.



**Gambar 4.5** *Bounding Box* Pada Gambar

*Bounding box* pada gambar saat labelling di *Roboflow* adalah kotak yang mengelilingi objek untuk menandai lokasinya. Kotak ini berisi koordinat yang digunakan model untuk mendeteksi dan mengenali objek dalam gambar. *Bounding box* membantu meningkatkan akurasi deteksi dalam *training* model.

c. *Split Data*

*Split data* yaitu membagi dataset menjadi tiga bagian, data pelatihan (*training*) dan data validasi (*validation*), dan data pengujian (*testing*). Data *training* dibagi menjadi 70 % dari keseluruhan dataset, data *validation* 20% dari keseluruhan dataset, dan data *testing* 10% dari keseluruhan dataset.

#### 4.1.4. *Modelling*

Pada tahap ini, akan dilakukan proses *training data* menjadi model, yang dilakukan menggunakan *google collab*. Pertama sambungkan *google collab*, langkah kedua membuat perintah untuk *cloning repository ultralytics yolov5* dan

*install dependencies*. Pembuatan model tidak hanya bisa dilakukan di google Colab, tetapi juga dapat dilakukan di berbagai platform lain seperti Jupyter Notebook (lokal atau melalui JupyterHub), Kaggle Notebooks. Selain itu, model juga dapat dikembangkan menggunakan aplikasi berbasis desktop seperti PyCharm atau VS Code dengan konfigurasi lingkungan yang sesuai. Pemilihan platform bergantung pada kebutuhan pengguna, seperti ketersediaan GPU, kemudahan akses, dan integrasi dengan dataset atau framework yang digunakan.

Penelitian ini menggunakan Google Colab karena tidak memakan banyak memori di perangkat lokal, sehingga lebih ringan digunakan. Selain itu, Google Colab menyediakan akses gratis ke GPU dan TPU untuk mempercepat proses pelatihan model. Integrasinya dengan *Google drive* juga memudahkan penyimpanan dan pengelolaan dataset. Platform ini mendukung kolaborasi *real-time* serta memiliki pustaka dan paket yang sudah terinstal, sehingga dapat langsung fokus pada pengembangan model tanpa konfigurasi tambahan. Pada saat melakukan pemodelan data, terdapat beberapa istilah penting yang perlu dipahami. Berikut adalah beberapa di antaranya:

a. *Epoch*

*Epoch* adalah jumlah perulangan penuh saat model memproses seluruh dataset dalam pelatihan. Semakin banyak *epoch*, semakin baik model belajar, tetapi terlalu banyak dapat menyebabkan overfitting.

b. GFLOPs (*Giga Floating Point Operations per Second*)

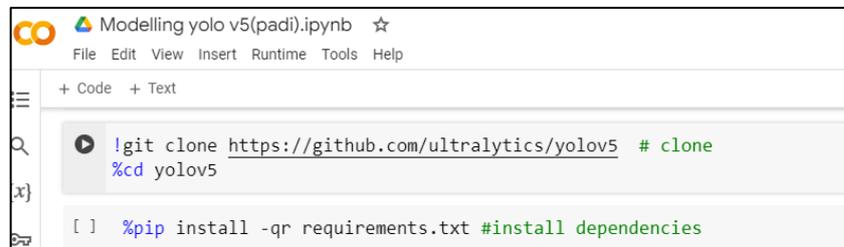
GFLOPs mengukur jumlah operasi matematika yang dapat dilakukan model per detik. Semakin tinggi nilainya, semakin besar kebutuhan komputasi, yang memengaruhi kecepatan, efisiensi, dan perangkat keras yang diperlukan.

c. *Fusing Layers*

*Fusing layers* adalah teknik menggabungkan beberapa layer dalam jaringan neural untuk meningkatkan representasi data dan akurasi deteksi, seperti yang digunakan dalam model *YOLO*.

#### d. Optimizer

Algoritma yang menyesuaikan bobot model untuk meminimalkan kesalahan (loss) dan meningkatkan akurasi. Contohnya adalah Adam, SGD, dan RMSprop.



```

!git clone https://github.com/ultralytics/yolov5 # clone
%cd yolov5

[] %pip install -qr requirements.txt #install dependencies

```

**Gambar 4. 6** Cloning Yolov5 Dan Install Dependencies

*Cloning YOLOv5* berfungsi untuk mengunduh dan mengakses repository secara lokal. Instalasi *dependencies* diperlukan agar semua paket pendukung *YOLOv5* tersedia dan dapat dijalankan. Verifikasi instalasi memastikan *YOLOv5* berfungsi dengan benar melalui deteksi objek.



```

!git clone https://github.com/ultralytics/yolov5 # clone
%cd yolov5

Cloning into 'yolov5'...
remote: Enumerating objects: 17893, done.
remote: Counting objects: 1806 (40/40), done.
remote: Compressing objects: 1806 (12/12), done.
remote: Total 17893 (delta 27), reused 0 (delta 8), pack-reused 17853 (from 4)
Receiving objects: 1806 (17893/17893), 15.71 MiB | 15.12 MiB/s, done.
Resolving deltas: 1806 (11728/11728), done.
/content/yolov5

[] %pip install -qr requirements.txt #install dependencies

[] from google.colab import drive
drive.mount('/content/drive')

[] !unzip /content/drive/MyDrive/data.zip -d /content/yolov5/data

[] %pip install -q roboflow

[] !pip install roboflow

from roboflow import RoboFlow
rf = RoboFlow(api_key="1D81hg31jgthwEppE68")
project = rf.workspace("Ta-Bismillah-otmeq").project("deteksi-penyakit-padi-8t5j7")
version = project.version(13)
dataset = version.download("yolov5")

[] %pwd

[] %cd /content/yolov5

```

**Gambar 4. 7** Daftar File Yang Sudah Di Cloning

Selanjutnya dataset di kompres menjadi file dengan ekstensi zip, kemudian data di *upload* ke dalam *google drive* yang bertujuan memudahkan *google collab* memasukkan data ke dalam *directory*. Setelah file di *upload* buat perintah untuk menghubungkan *google collab* dan *google drive*. Setelah berhasil jalankan perintah *unzip* untuk ekstrak file yang di *upload* ke *google drive* sebelumnya yang otomatis akan masuk ke *directory yolov5/data*.

```

from google.colab import drive
drive.mount('/content/drive')

!unzip /content/drive/MyDrive/data.zip -d /content/yolov5/data

!pip install -q roboflow

from roboflow import RoboFlow
rf = RoboFlow(api_key="ID81hgE31jgIhwEPPEEB")
project = rf.workspace("ta-bismillah-ozmeq").project("deteksi-penyakit-padi-8t5j7")
version = project.version(13)
dataset = version.download("yolov5")

!pwd

!cd /content/yolov5

!dataset.location

!cat (dataset.location)/data.yaml

!import yaml

```

**Gambar 4. 8** Menghubungkan *Google Collab* Dan *Google Drive*

Menghubungkan *Google Collab* dengan *Google drive* memungkinkan penyimpanan dan akses data secara langsung. Proses ini dilakukan dengan mount *Google drive* menggunakan perintah `from google.colab import drive` lalu `drive.mount('/content/drive')`. Setelah terhubung, file di *Google drive* dapat diakses melalui path `/content/drive/MyDrive/`.

```

from google.colab import drive
drive.mount('/content/drive')

!unzip /content/drive/MyDrive/data.zip -d /content/yolov5/data

```

**Gambar 4. 9** Perintah Mengekstrak File Zip

Setelah seluruh proses konfigurasi berhasil diselesaikan dengan baik, langkah selanjutnya adalah menjalankan perintah yang telah disiapkan untuk memulai proses pelatihan data. Proses ini bertujuan untuk melatih model agar mampu mengenali pola dan karakteristik tertentu berdasarkan dataset yang telah



Kode ini digunakan untuk melatih *YOLOv5* dengan gambar ukuran 256x256, batch size 100, dan 100 *epoch*, menggunakan dataset yang sudah disiapkan. Model yang digunakan adalah *YOLOv5s* dengan bobot awal dari *yolov5s.pt*, dan cache digunakan untuk mempercepat pelatihan.



**Gambar 4. 12** Proses *Training data*

Proses *training data* pada *YOLOv5* dimulai dengan menyiapkan dataset yang telah dilabeli dan dikonfigurasi dalam format yang sesuai. Selanjutnya, model dilatih menggunakan perintah `python train.py --data dataset.yaml --weights yolov5s.pt --epochs 50`, yang akan menyesuaikan bobot model berdasarkan dataset. Setelah selesai, hasil *training* berupa bobot baru disimpan dan dapat digunakan untuk deteksi objek.

```

100 epochs completed in 0.977 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.3MB
Optimizer stripped from runs/train/exp/weights/best.pt, 14.3MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7023610 parameters, 0 gradients, 15.8 GFLOPS

```

Class	Images	Instances	P	R	mAP50	mAP50-95
all	1868	2979	0.949	0.914	0.953	0.672
Blast	1868	562	0.904	0.913	0.958	0.571
Blight	1868	543	0.969	0.963	0.984	0.735
Brown Spot	1868	681	0.877	0.706	0.835	0.434
Healthy	1868	579	1	0.996	0.995	0.837
Tungro	1868	614	0.994	0.992	0.994	0.782

```

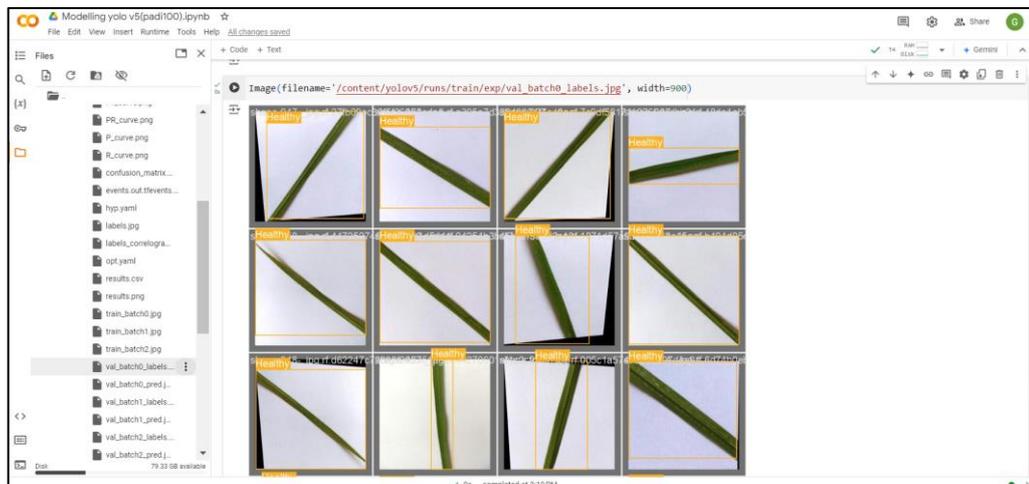
Results saved to runs/train/exp
CPU times: user 21.6 s, sys: 3.6 s, total: 25.2 s
Wall time: 1h 23s

```

**Gambar 4. 13** Hasil *Training data*

Model telah menyelesaikan pelatihan selama 100 *epoch* dalam waktu sekitar 58 menit, dataset telah dipelajari sebanyak 100 kali untuk meningkatkan akurasi deteksi. Setelah pelatihan selesai, bobot terbaik disimpan dalam file

*best.pt*, sementara optimizer dihapus untuk mengurangi ukuran file menjadi 14.3MB, karena optimizer hanya diperlukan saat pelatihan, bukan untuk deteksi. Model kemudian diuji menggunakan *validasi*, di mana dilakukan proses *fusing layers* untuk mengoptimalkan kecepatan tanpa mengurangi akurasi. Model ini memiliki 157 layers dengan 7 juta parameter, yang menunjukkan kompleksitasnya, serta 15.8 GFLOPs, yang mengukur jumlah operasi komputasi yang dilakukan model. Dengan ini, model siap digunakan untuk mendeteksi penyakit daun padi secara lebih efisien dan akurat.



**Gambar 4. 14** Prediksi Label Pada *Batch* Dataset *Validasi*

Prediksi label pada batch dataset *validasi* digunakan untuk mengevaluasi kinerja model setelah pelatihan. Proses ini membandingkan hasil prediksi model dengan label asli untuk mengukur akurasi dan efektivitas deteksi. Evaluasi ini membantu memastikan model siap digunakan pada data baru dengan performa optimal.

#### 4.1.5. *Evaluation*

Pada tahap ini, setelah model dilatih adalah memastikan akurasi. Evaluasi dilakukan dengan memeriksa grafik, *confusion matrix*, dan hasil *validasi* di direktori *yolov5/runs/train/exp/*. Jika hasilnya sesuai model diuji menggunakan perintah *detect.py*. parameter *weights* digunakan untuk mengakses model yang dilatih, dan *--source* untuk data uji berupa gambar dari *folder test*. Hasil deteksi

disimpan di *folder* `yolov5/runs/detect/exp/` dalam bentuk gambar dengan *bounding box*. Model mencapai akurasi keseluruhan (*mAP*) sebesar 96.7% pada *threshold confidence* tertentu, menunjukkan bahwa model mampu mendeteksi penyakit dengan tingkat presisi yang baik.

```

python detect.py --weights /content/yolov5/runs/train/exp/weights/best.pt --img 640 --conf 0.1 --source /content/yolov5/Deteksi-Penyakit-Padi-13
Image 826/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-457.jpg: 0400404 2 Healthy, 6.2ms
Image 827/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-462.jpg: 0400404 2 Healthy, 5.1ms
Image 828/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-462.jpg: 0400404 2 Healthy, 5.4ms
Image 829/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-478.jpg: 0400404 2 Healthy, 5.1ms
Image 830/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-473.jpg: 0400404 2 Healthy, 7.2ms
Image 831/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-484.jpg: 0400404 2 Healthy, 9.8ms
Image 832/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-491.jpg: 0400404 2 Healthy, 5.1ms
Image 833/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-499.jpg: 0400404 2 Healthy, 5.3ms
Image 834/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-504.jpg: 0400404 2 Healthy, 6.8ms
Image 835/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-507.jpg: 0400404 2 Healthy, 6.5ms
Image 836/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-521.jpg: 0400404 2 Healthy, 5.4ms
Image 837/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-531.jpg: 0400404 2 Healthy, 5.1ms
Image 838/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-532.jpg: 0400404 2 Healthy, 5.3ms
Image 839/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-534.jpg: 0400404 2 Healthy, 5.4ms
Image 840/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-548.jpg: 0400404 2 Healthy, 7.0ms
Image 841/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-548.jpg: 0400404 2 Healthy, 6.7ms
Image 842/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-541.jpg: 0400404 2 Healthy, 5.3ms
Image 843/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-547.jpg: 0400404 2 Healthy, 5.9ms
Image 844/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-547.jpg: 0400404 2 Healthy, 5.3ms
Image 845/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-548.jpg: 0400404 2 Healthy, 5.3ms
Image 846/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-562.jpg: 0400404 2 Healthy, 5.3ms
Image 847/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-563.jpg: 0400404 2 Healthy, 5.4ms
Image 848/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-566.jpg: 0400404 2 Healthy, 5.3ms
Image 849/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-578.jpg: 0400404 2 Healthy, 6.9ms
Image 850/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-578.jpg: 0400404 2 Healthy, 5.3ms
Image 851/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-588.jpg: 0400404 1 Healthy, 5.3ms
Image 852/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-593.jpg: 0400404 2 Healthy, 6.2ms
Image 853/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-591.jpg: 0400404 2 Healthy, 5.3ms
Image 854/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-592.jpg: 0400404 2 Healthy, 5.4ms
Image 855/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-601.jpg: 0400404 2 Healthy, 5.3ms
Image 856/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-618.jpg: 0400404 2 Healthy, 5.4ms
Image 857/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-621.jpg: 0400404 2 Healthy, 10.3ms
Image 858/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-621.jpg: 0400404 2 Healthy, 5.3ms
Image 859/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-621.jpg: 0400404 2 Healthy, 5.3ms
Image 860/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-621.jpg: 0400404 2 Healthy, 5.3ms
Image 861/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-626.jpg: 0400404 2 Healthy, 5.4ms
Image 862/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-628.jpg: 0400404 2 Healthy, 5.1ms
Image 863/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-629.jpg: 0400404 2 Healthy, 5.3ms
Image 864/927 /content/yolov5/Deteksi-Penyakit-Padi-13/test/Images/shape-641.jpg: 0400404 2 Healthy, 5.4ms

```

**Gambar 4. 15** Perintah Deteksi Gambar

Perintah deteksi gambar digunakan untuk menguji model *YOLOv5* dalam mengenali objek pada gambar. Proses ini melibatkan input gambar ke dalam model yang telah dilatih untuk menghasilkan prediksi berupa *bounding box* dan label objek. Hasil deteksi kemudian ditampilkan dengan visualisasi yang menunjukkan objek yang teridentifikasi.

#### 4.1.6. Deployment

Pada tahap ini, data yang sudah sesuai bisa di download. Setelah selesai di download, data tersebut perlu di dekompresi agar dapat di ekstrak dari file arsip dengan format zip, bertujuan untuk mengembalikan file ke dalam struktur *folder* dan format aslinya.



arsip yang lebih ringkas. File yang diunduh dapat digunakan kembali untuk prediksi atau deployment tanpa perlu melatih ulang model.

#### 4.2. Analisa Proses

Pada penelitian ini, analisa data dalam pengembangan aplikasi ini menonjolkan penggunaan *yolo* sebagai model deteksi penyakit daun padi. Data yang telah diproses dibagi menjadi tiga bagian utama untuk pelatihan, *validasi*, dan pengujian. Proses pelatihan mencakup penyesuaian parameter seperti *learning rate*, ukuran *batch*, dan jumlah *epoch* untuk mengoptimalkan performa model dalam mengenali jenis penyakit pada padi. *Confusion matrix* digunakan untuk menganalisis hasil pada setiap kategori penyakit. Evaluasi kerja dilakukan menggunakan matriks *precision*, *recall*, *f1-score*, dan *mAP* untuk mengukur tingkat akurasi deteksi. Rumus yang digunakan untuk menghitung *precision*, *recall*, *f1-score*, dan *mAP* sudah dijelaskan pada bab sebelumnya di bab tiga.

#### 4.3. Analisa Sistem

Pada penelitian ini, aplikasi dirancang untuk memproses gambar yang diunggah pengguna melalui antarmuka web responsive, yang dibangun menggunakan flask dan bootstrap. Aplikasi ini terintegrasi sepenuhnya sehingga memungkinkan pengguna mendapatkan hasil deteksi dengan cepat dan akurat melalui perangkat desktop maupun *mobile*. Untuk memudahkan proses perancangan sistem maka dibutuhkan perangkat keras dan perangkat lunak yang digunakan selama melakukan penelitian.

##### a. Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan dan operasional *Agricare* mencakup komponen utama yang mendukung pemrosesan data, eksekusi program, serta interaksi dengan sistem. Perangkat ini memastikan bahwa aplikasi dapat berjalan dengan optimal, baik dalam proses pelatihan model maupun dalam mendeteksi penyakit pada daun padi secara *real-time*. Perangkat keras yang digunakan dalam pembuatan website ini adalah sebagai berikut:

**Tabel 4. 1** Tabel Perangkat Keras

Item	Keterangan
Laptop	Operating system windows 11 – 64-bit, processor i5-10310U
Mouse	Untuk interaksi dengan perangkat komputer

b. Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan *Agricare* terdiri dari berbagai tools dan framework yang mendukung proses pengolahan dataset, pelatihan model, pengembangan antarmuka, serta eksekusi program. Beberapa perangkat lunak utama yang digunakan mencakup platform untuk pengolahan dataset, alat pemrograman untuk pengeditan dan eksekusi kode, serta framework yang mendukung tampilan frontend dan backend aplikasi. Dengan kombinasi perangkat lunak ini, *Agricare* dapat menjalankan proses deteksi penyakit pada daun padi secara optimal dan memberikan pengalaman pengguna yang responsif serta efisien. Perangkat lunak yang digunakan dalam pembuatan website ini adalah sebagai berikut:

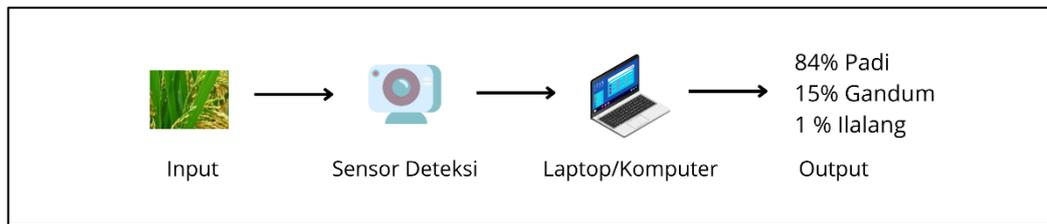
**Tabel 4. 2** Tabel Perangkat Lunak

Item	Keterangan
<i>Roboflow</i>	Digunakan untuk mengolah dataset mentah menjadi dataset yang siap di <i>training</i>
Google Colab	Spesifikasi NVIDIA (R) Cuda Compiler Driver
CMD	Digunakan untuk menjalankan perintah program
Visual Studio Code	Digunakan untuk editing code dan menjalankan program
Bootstrap	Digunakan untuk membuat tampilan frontend
Flask	Digunakan untuk membuat tampilan backend

#### 4.4. Perancangan Model

Pada penelitian ini, perancangan model berbasis *computer vision* dirancang dengan tujuan untuk mendukung dan memudahkan proses

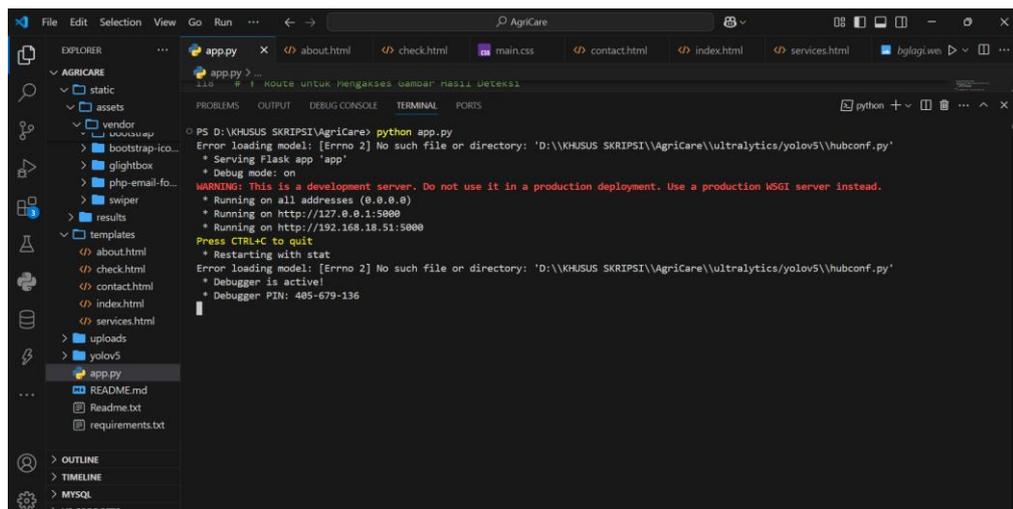
pengimplementasian website secara keseluruhan. Perancangan model dapat dilihat dari gambar 4.19.



**Gambar 4. 18** Model Analogi *Computer vision*

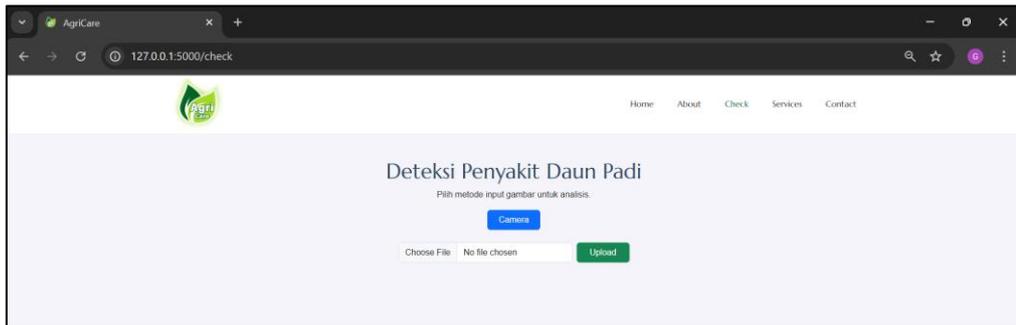
Cara kerja *computer vision* meniru sistem visual manusia. Manusia menggunakan mata untuk melihat objek disekitarnya, lalu citra yang ditangkap mata dikirimkan ke otak untuk di proses dan di interpretasikan guna pengambilan keputusan. Agar lebih jelas bagaimana cara proses deteksi gambar berikut adalah tahapannya:

- a. Langkah pertama, yang dilakukan adalah masuk ke dalam website yang digunakan untuk mendeteksi gambar daun padi. Untuk masuk ke aplikasi ketik *python app.py* pada terminal di vscode. Setelah muncul tahapan berikutnya adalah klik salah satu link seperti pada gambar *Running on http://127.0.0.1:5000* jika menggunakan laptop atau computer atau *Running on http://192.168.18.51:5000* dalam bentuk *mobile*.



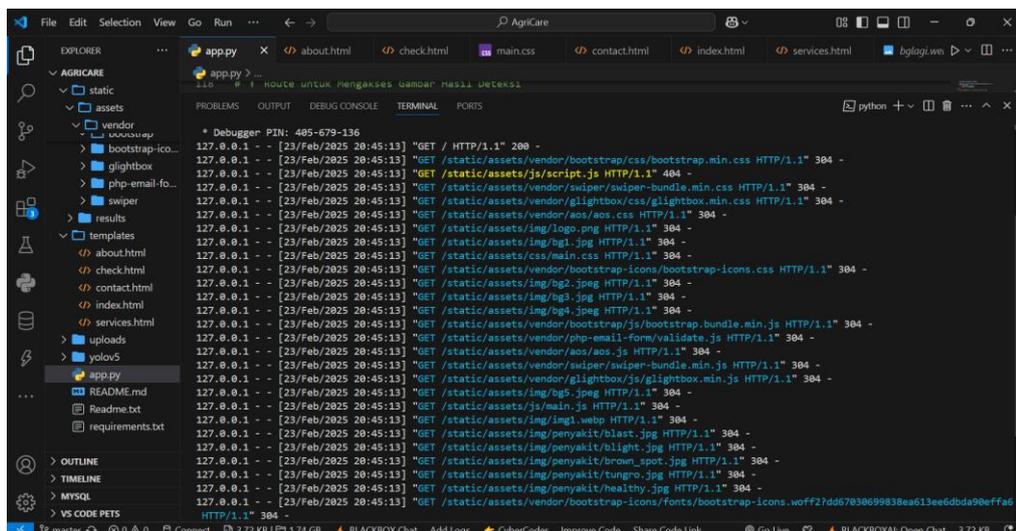
**Gambar 4. 19** Masuk Ke Aplikasi *Agricare*

- b. Selanjutnya, setelah aplikasi muncul langsung saja pilih menu *check*, pada halaman *check* disediakan dua fitur yaitu *camera* dan *upload* Dimana fungsinya sama yaitu untuk mendeteksi jenis penyakit pada daun padi. Jika menggunakan *camera* maka gambar bisa dideteksi secara langsung atau *real-time* dan juga bisa di *capture*. Sedangkan *upload*, diarahkan untuk memilih gambar yang akan dideteksi terlebih dahulu baru proses deteksi akan dikerjakan.



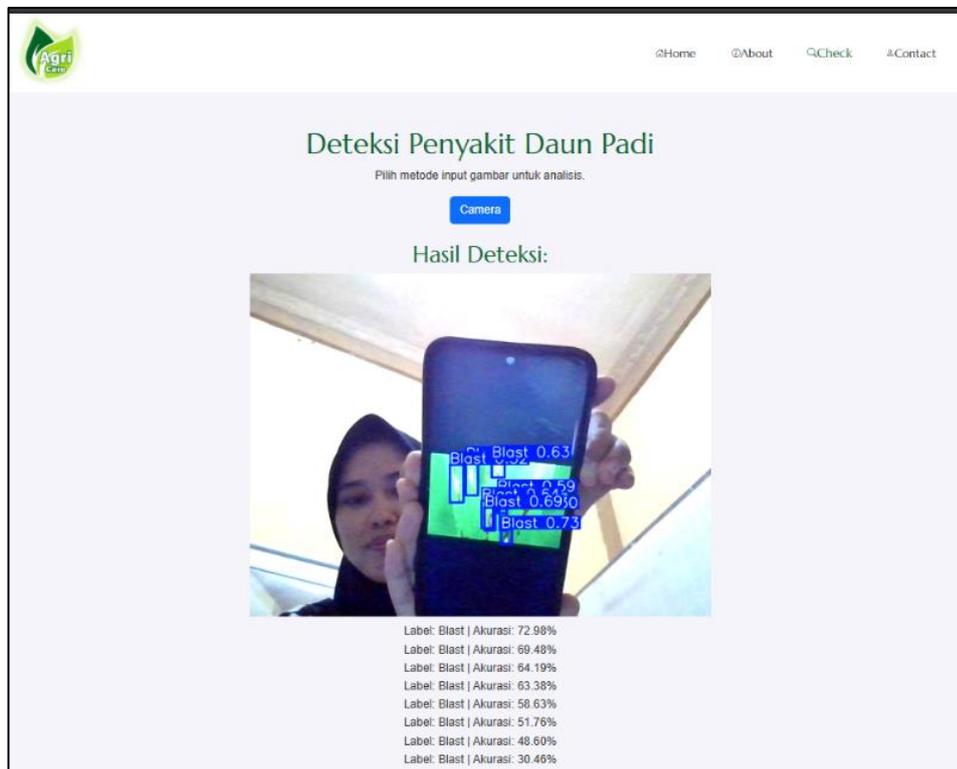
**Gambar 4. 20** Fitur Untuk Mendeteksi Penyakit Pada Daun Padi

- c. Selanjutnya, proses deteksi akan dijalankan dengan menggunakan model yang telah dilatih. Sistem akan memproses data input dan mengidentifikasi objek yang sesuai dengan label yang dikenali. Setelah itu, hasil deteksi akan ditampilkan dalam bentuk visual dengan *bounding box* dan label objek.



**Gambar 4. 21** Gambaran Program Yang Sedang Berjalan

- d. Setelah objek berhasil dideteksi, aplikasi akan menampilkan hasil analisis secara visual. Pengguna akan melihat gambar dengan *bounding box* yang menandai objek yang teridentifikasi. Selain itu, informasi mengenai jenis penyakit yang terdeteksi juga akan ditampilkan pada layar.



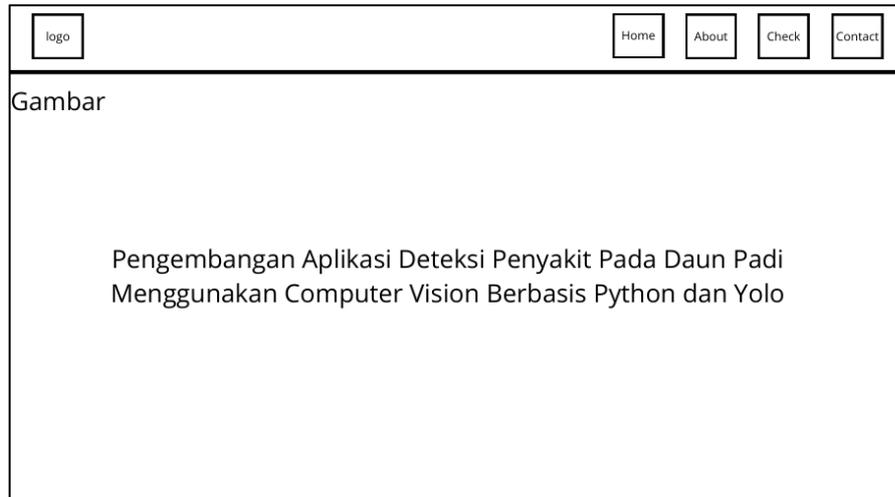
**Gambar 4. 22** Hasil Deteksi Pada Aplikasi

#### 4.5. Perancangan *Interface*

Pada penelitian ini, perancangan antarmuka (*interface*) dirancang untuk memudahkan proses pengembangan sistem secara keseluruhan. Perancangan *interface* yang digunakan terdiri dari tampilan *home*, *about*, *check*, *services*, dan *contact*.

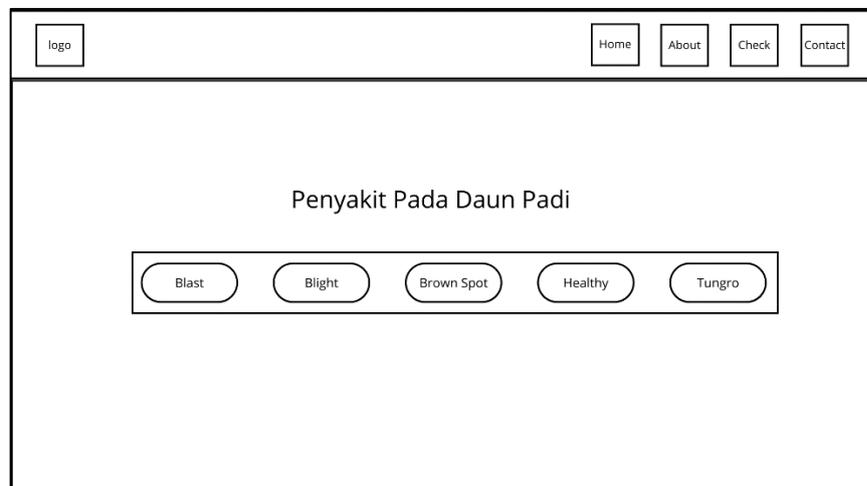
##### 4.5.1. Tampilan *Home*

Tampilan *home* pada perancangan *interface* dirancang untuk memberikan kesan pertama yang jelas dan menarik bagi pengguna. Halaman ini menyajikan informasi utama yang diperlukan dengan tata letak sederhana namun informatif. Tampilan *home* dapat dilihat pada gambar berikut ini.



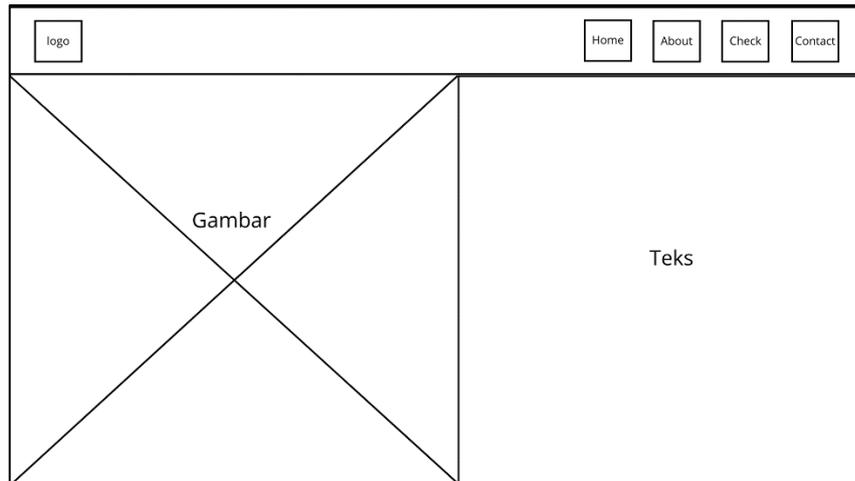
**Gambar 4. 23** Halaman *Home*

Tampilan awal pada halaman *home* menampilkan antarmuka aplikasi yang interaktif, di mana pengguna dapat menggulir atau menggeser elemen yang tersedia. Desain ini memberikan pengalaman yang lebih dinamis dan memudahkan pengguna dalam menjelajahi fitur utama aplikasi.



**Gambar 4. 24** Halaman *Home 2*

Tampilan selanjutnya menampilkan penyakit pada daun padi yang disajikan dalam bentuk visual dan deskripsi informatif. Pengguna dapat melihat jenis padi berdasarkan karakteristiknya, seperti bentuk, warna, dan keunggulan masing-masing, sehingga memudahkan dalam mengenali varietas yang tersedia.



**Gambar 4. 25** Halaman *Home 3*

Tampilan selanjutnya pada *agricare* berisi faq yang menjawab berbagai pertanyaan umum seputar daun padi. Pengguna dapat menemukan informasi tentang jenis penyakit, gejala, dan cara pencegahannya. Halaman ini membantu petani dan pengguna memahami lebih lanjut kesehatan tanaman padi.

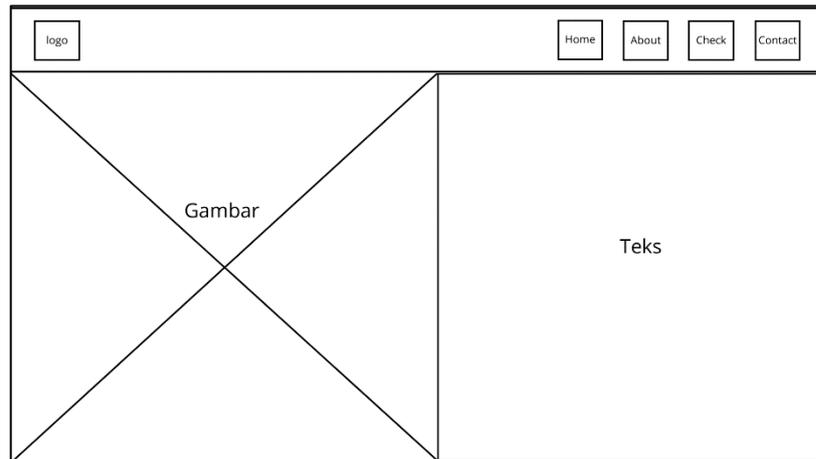
#### 4.5.2. Tampilan *About*

Tampilan *about* pada perancangan *interface* dirancang untuk memberikan informasi yang jelas tentang aplikasi dan tujuannya. Desainnya sederhana namun informatif, dengan navigasi yang mudah agar pengguna dapat memahami konteks aplikasi. Tampilan *about* dapat dilihat pada gambar berikut ini.



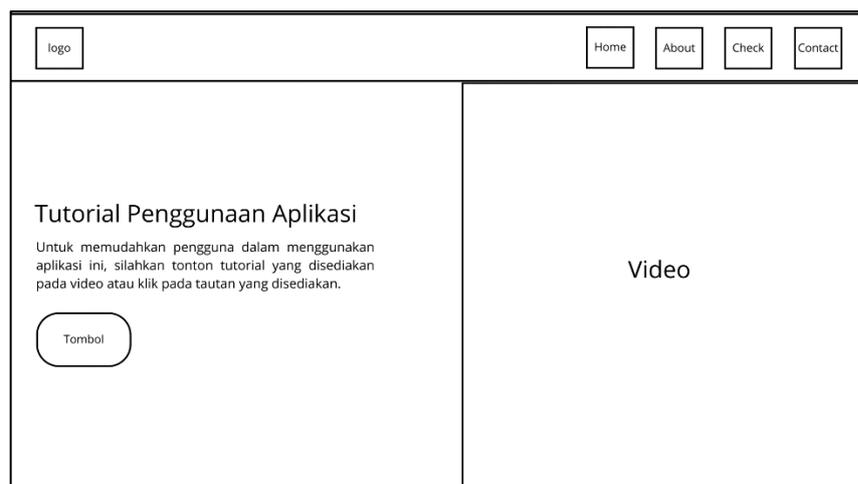
**Gambar 4. 26** Halaman *About*

Halaman *about* pertama menampilkan gambaran singkat tentang *agricare*, sebuah platform yang dirancang untuk mendeteksi penyakit pada daun padi secara akurat dan efisien. *Agricare* menggunakan teknologi *computer vision* dan *yolo* untuk menganalisis kondisi daun padi, membantu petani dan pengguna dalam mengidentifikasi serta memahami berbagai jenis penyakit yang dapat memengaruhi hasil panen.



**Gambar 4. 27** Halaman *About 2*

Tampilan *about* pada *agricare* memberikan penjelasan singkat tentang apa itu *agricare*. Halaman ini menjelaskan tujuan utama *agricare* dalam membantu pengguna mengidentifikasi penyakit pada daun padi. Selain itu, juga dijelaskan bagaimana *agricare* dapat digunakan secara efektif.

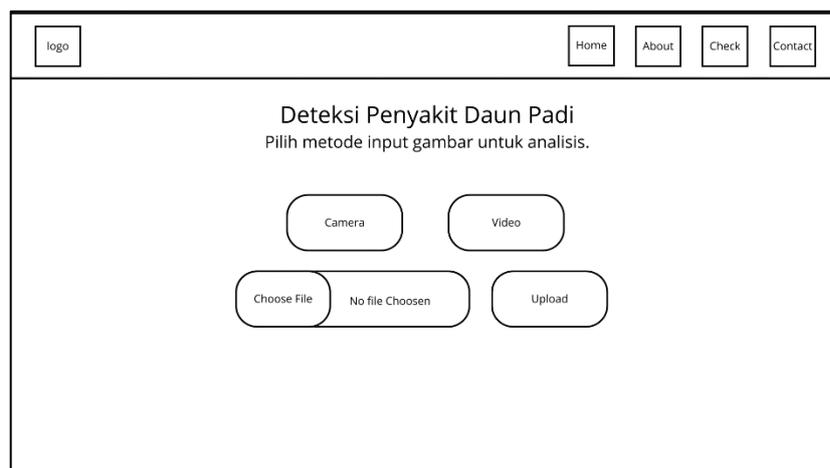


**Gambar 4. 28** Halaman *About 3*

Tampilan *About* berikutnya menyajikan tutorial penggunaan aplikasi yang dirancang untuk memandu pengguna dalam memanfaatkan fitur yang tersedia. Tutorial ini disajikan dalam bentuk video interaktif yang dapat diakses melalui tampilan YouTube, sehingga pengguna dapat dengan mudah memahami cara kerja aplikasi secara praktis dan mendetail.

#### 4.5.3. Tampilan *Check*

Tampilan *Check* pada perancangan *interface* dirancang untuk memudahkan pengguna dalam melakukan pemeriksaan penyakit pada daun padi. Desain antarmuka ini dirancang dengan navigasi yang intuitif, sehingga pengguna dapat dengan mudah mengoperasikan sistem untuk mendapatkan hasil deteksi penyakit padi yang akurat. Tampilan *Check* dapat dilihat pada gambar berikut ini.



**Gambar 4. 29** Halaman *Check*

Halaman *check* merupakan fitur utama *agricare* yang memungkinkan pengguna mendeteksi penyakit pada daun padi secara langsung. Pengguna dapat memilih untuk menggunakan kamera secara *real-time*, *capture* atau mengunggah file gambar untuk dianalisis. Sistem akan memproses gambar dengan teknologi *computer vision* berbasis *yolo*, kemudian menampilkan hasil deteksi berupa jenis penyakit, tingkat keparahan, serta rekomendasi penanganan yang sesuai.

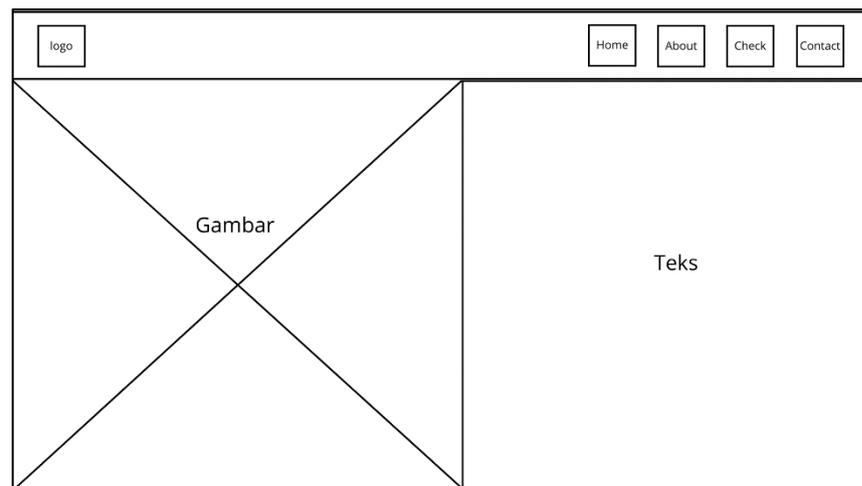
#### 4.5.4. Tampilan *Contact*

Tampilan *Contact* pada perancangan *interface* dirancang untuk memudahkan pengguna dalam menghubungi pengembang atau tim pendukung aplikasi. Desain antarmuka ini dibuat agar pengguna dapat dengan mudah mengakses dan mengirimkan pertanyaan atau masukan terkait aplikasi. Tampilan *contact* dapat dilihat pada gambar berikut ini.



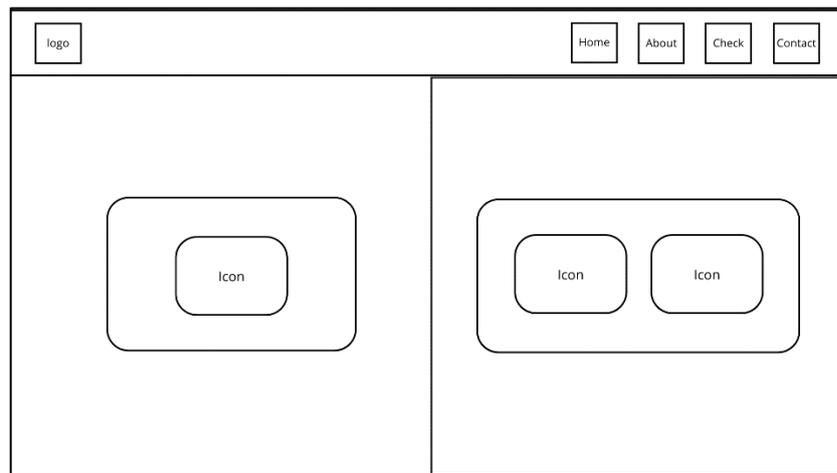
**Gambar 4. 30** Halaman *Contact*

Tampilan pertama *contact* pada *agricare* berisi ajakan kepada pengguna untuk menghubungi tim *agricare*. Halaman ini mengundang pengguna untuk memberikan pertanyaan, masukan, atau dukungan terkait layanan. Dengan ini, pengguna dapat lebih mudah berinteraksi dengan tim pengembang.



**Gambar 4. 31** Halaman *Contact 2*

Tampilan selanjutnya berisi informasi singkat tentang pembuat *agricare*. Halaman ini mencantumkan nama pembuat serta cara menghubungi untuk pertanyaan atau masukan. Tujuannya adalah memberikan transparansi dan akses komunikasi bagi pengguna.



**Gambar 4. 32** Halaman *Contact 3*

Halaman terakhir *contact* pada *agricare* memungkinkan pengguna untuk bertanya langsung kepada pengembang. Pengguna dapat menghubungi melalui whatsapp atau email yang telah disediakan. Halaman ini memudahkan komunikasi bagi yang membutuhkan bantuan atau memiliki pertanyaan.

## BAB V IMPLEMENTASI DAN PENGUJIAN

### 5.1. Implementasi

Pada tahap implementasi, aplikasi penyakit daun padi yang menggunakan model *yolo* diintegrasikan ke dalam aplikasi berbasis website. Model *yolo* yang telah dilatih untuk mendeteksi berbagai penyakit pada daun padi digunakan di *backend* aplikasi. Untuk menghubungkan model dengan website, *framework flask* digunakan untuk menangani permintaan dari pengguna dan menjalankan model. Antarmuka website dirancang menggunakan *html*, *css*, dan *bootstrap* untuk memastikan kemudahan penggunaan. Pengguna dapat mengunggah gambar daun padi melalui halaman *check*, yang kemudian diproses oleh model untuk mendeteksi penyakit. Hasil deteksi ditampilkan kepada pengguna dalam bentuk informasi yang jelas dan mudah dipahami.

#### 5.1.1. Hasil *Modelling*

Setelah model dilatih menggunakan jumlah *epoch* yang berbeda-beda, *folder runs/train/* akan berisi beberapa *folder exp* yang masing-masing menyimpan file yang dapat digunakan untuk melihat hasil dari pelatihan data. Untuk membandingkan nilai *recall*, presisi, dan *F1-score* pada berbagai *epoch*, informasi tersebut dapat dilihat pada tabel berikut.

**Tabel 5. 1** Perbandingan Hasil Setiap *Epoch*

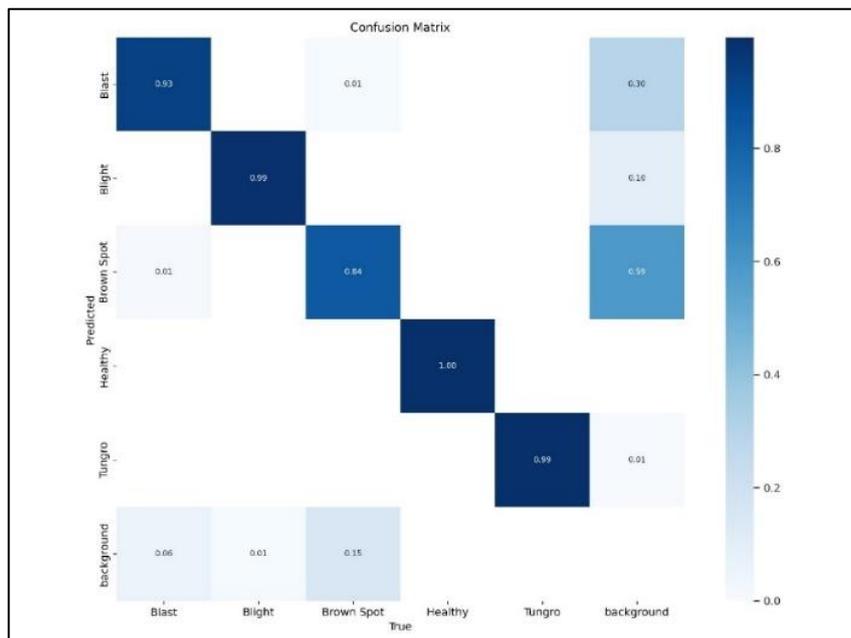
<i>Epoch</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
10	1.00	0.93	0.81
20	1.00	0.96	0.87
50	1.00	0.97	0.91
75	1.00	0.98	0.92
100	1.00	0.98	0.93

Berdasarkan tabel di atas, dapat dilihat bahwa dataset yang dilatih dengan 100 *epoch* menunjukkan hasil yang lebih baik dibandingkan dengan data yang dilatih menggunakan jumlah *epoch* lainnya. Model yang telah dilatih dengan 100

*epoch* menghasilkan kurva performa yang lebih optimal, seperti yang ditunjukkan pada kurva berikut:

a. *Confusion Matrix*

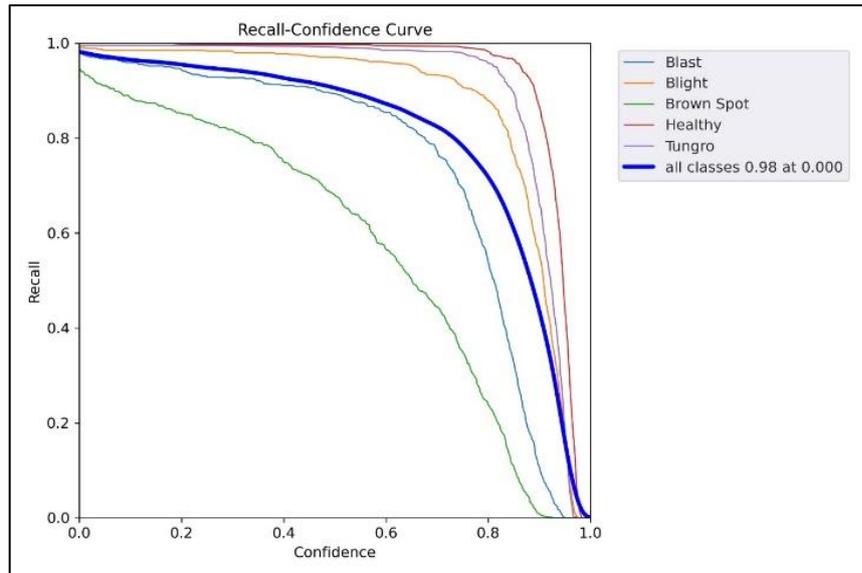
Hal pertama yang perlu diperhatikan untuk menilai apakah model sudah baik atau belum adalah dengan melihat *confusion matrix*. Untuk model yang telah dilatih sebelumnya, jika mengacu pada *confusion matrix* di bawah, dapat disimpulkan bahwa model sudah berfungsi dengan baik.



**Gambar 5.1** *Confusion Matrix*

b. *Recall*

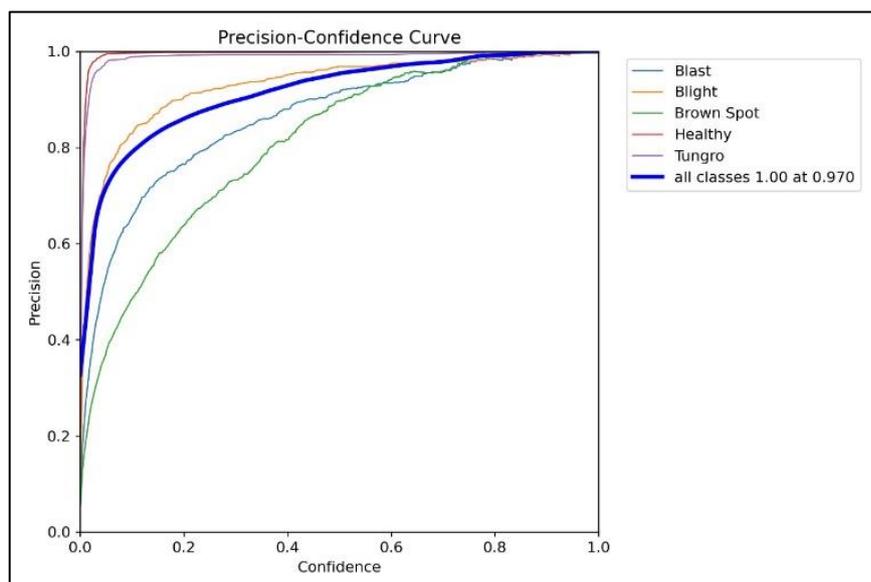
*Recall* adalah parameter penting untuk menilai kinerja model. Berdasarkan kurva *recall* di bawah, setiap kelas menunjukkan nilai *recall* yang mencapai 0 pada *confidence* 0.98, yang menandakan bahwa model telah mencapai skor yang diinginkan dan memiliki performa yang baik.



**Gambar 5. 2 Recall**

c. *Precision*

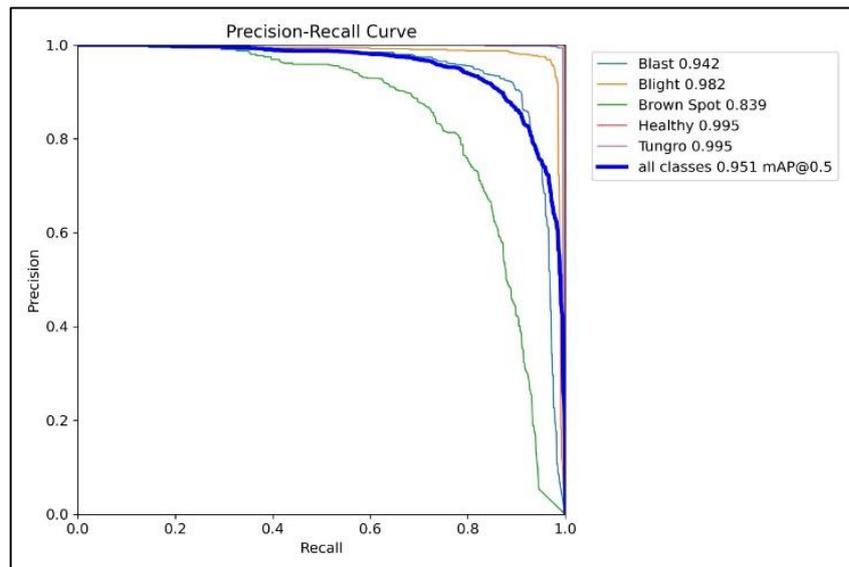
*Precision* dan *recall* memiliki hubungan yang berlawanan. *Recall* akan bernilai baik jika kurvanya semakin lama semakin menurun, sedangkan *precision* menunjukkan kualitas model yang semakin baik jika kurvanya terus meningkat. Berdasarkan kurva di bawah, terlihat bahwa *precision* mengarah ke atas, yang menunjukkan bahwa kualitas model semakin baik. Nilai *precision* mencapai 1 pada nilai *confidence* 0.970, yang menandakan bahwa model bekerja dengan sangat akurat dalam mengidentifikasi kelas pada tingkat kepercayaan tersebut.



**Gambar 5. 3 Precision**

d. *Precision Recall Curve*

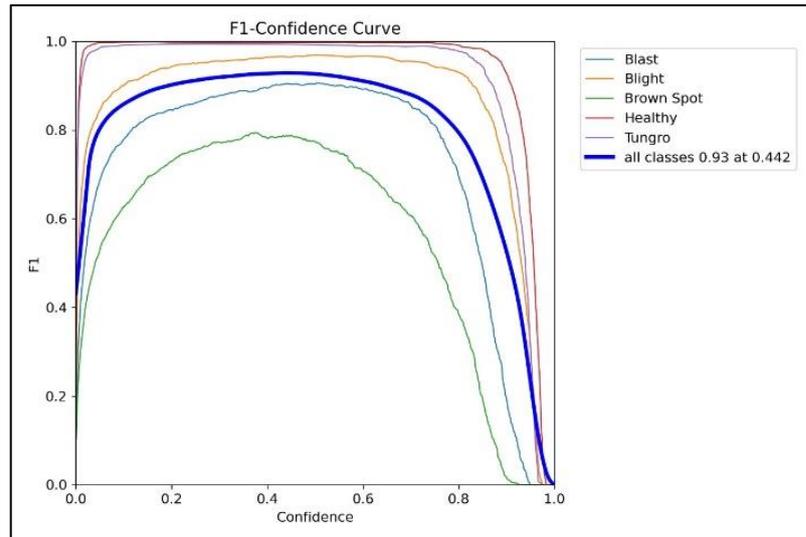
Setelah menganalisis *recall* dan *precision*, kedua parameter tersebut dibandingkan untuk mendapatkan skor. Hasilnya, untuk kelas *blast* 0.942, *blight* 0.982, *brown spot* 0.839, *healthy* 0.995, *tungro* 0.995. Dalam konteks *precision-recall*, yang diinginkan adalah *precision* tinggi dan *recall* rendah, karena pada penelitian ini dibutuhkan pendeteksian objek yang akurat dan tepat.



**Gambar 5. 4** *Precision Recall Curve*

e. *F1-Score*

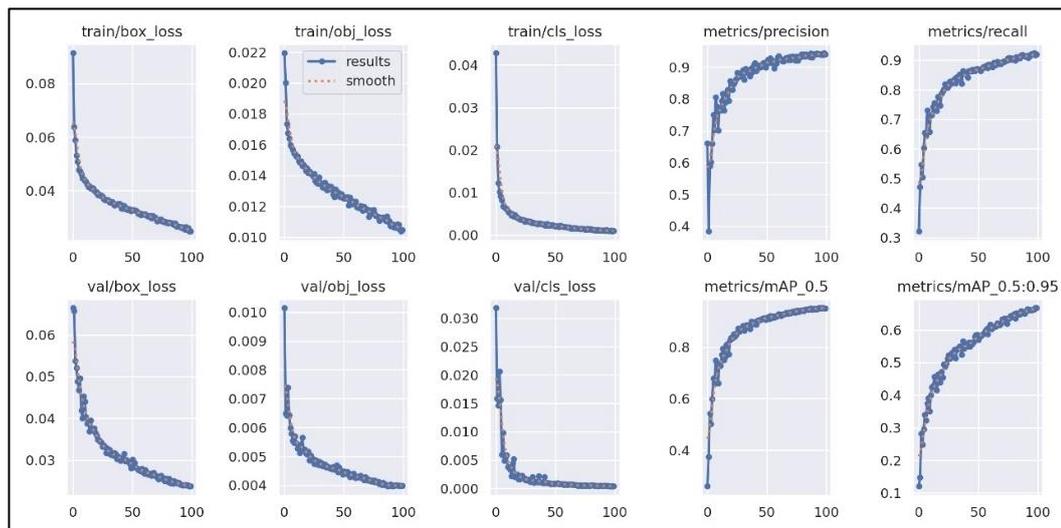
*F1-score* adalah parameter penting untuk menilai apakah model sudah sesuai dengan yang diinginkan. Berdasarkan kurva F1 di bawah, terlihat bahwa F1 skor mencapai 0.93 atau 93% pada *confidence* 0.442. Dengan *F1-score* yang tinggi, dapat disimpulkan bahwa model memiliki kualitas yang baik dan mampu memberikan hasil yang optimal.



**Gambar 5. 5** *F1-Score*

f. *Loss Graph*

Nilai *loss* adalah parameter penting untuk menilai kinerja model. Semakin kecil nilai *loss*, semakin baik model dalam melakukan klasifikasi objek. Kurva *Loss* menunjukkan tren menurun baik untuk pelatihan maupun *validasi*, yang menunjukkan bahwa model sedang belajar dan meningkat seiring waktu. Metrik juga menunjukkan kinerja yang baik, dengan kurva presisi dan *recall* mendekati 1.0 dan nilai *mAP* meningkat secara bertahap.



**Gambar 5. 6** *Loss Graph*

### 5.1.2. Hasil Perhitungan *Accuracy* dan *mAP*

Pada tahap ini, *confusion matrix* akan diuraikan menjadi dua bagian utama, yaitu tabel *confusion matrix* dan tabel Identifikasi TP, FP, FN, dan TN. Pembagian ini disajikan secara rinci yang bisa dilihat pada tabel 4.1 *Confusion matrix* dan tabel 4.2 Identifikasi TP, FP, FN, TN.

**Tabel 5. 2** *Confusion Matrix*

<i>Predicted/True</i>	<i>Blast</i>	<i>Blight</i>	<i>Brown Spot</i>	<i>Healthy</i>	<i>Tungro</i>
<i>Blast</i>	0.93	0.01	0.00	0.00	0.06
<i>Blight</i>	0.00	0.99	0.01	0.00	0.00
<i>Brown Spot</i>	0.01	0.00	0.84	0.00	0.15
<i>Healthy</i>	0.00	0.00	0.00	1.00	0.00
<i>Tungro</i>	0.00	0.00	0.00	0.01	0.99

Tabel 4.1 menunjukkan *confusion matrix* hasil klasifikasi model *yolo* dalam mendeteksi penyakit pada daun padi. Model memiliki akurasi tinggi dengan nilai diagonal utama mendekati 1. Misalnya, kelas *blast* diklasifikasikan dengan akurasi 93%, sedangkan *healthy* terdeteksi sempurna dengan akurasi 100%. Kesalahan klasifikasi terjadi pada beberapa kelas, seperti *brown spot*, yang salah diprediksi sebagai *tungro* sebesar 15%.

**Tabel 5. 3** Identifikasi TP, FP, FN, TN

Kelas	TP	FP	FN	TN
<i>Blast</i>	0.93	$0.00 + 0.01 + 0.06 = 0.07$	0.01	4.07
<i>Blight</i>	0.99	$0.00 + 0.01 + 0.00 = 0.01$	0.01	4.00
<i>Brown spot</i>	0.84	$0.01 + 0.00 + 0.15 = 0.16$	0.01	4.15
<i>Healthy</i>	1.00	0.00	0.00	4.00
<i>Tungro</i>	0.99	0.01	0.00	4.01

Untuk menghitung nilai *Accuracy* dan *mAP*, langkah pertama yang dilakukan adalah menentukan nilai *precision*, *recall*, *f1-score*. Dengan cara ini,

perhitungan *accuracy* dan *mAP* dapat dilakukan secara lebih menyeluruh berdasarkan hasil evaluasi dari masing-masing kelas.

a. *Blast*

Pada deteksi penyakit *blast* ini, dengan perhitungan yang telah diberikan, didapatkan nilai akurasi, *recall*, dan *f1-score* yang mencerminkan kinerja model. Hasil ini menunjukkan efektivitas model dalam mendeteksi penyakit *blast* pada daun padi. Nilai-nilai tersebut menggambarkan akurasi dan kemampuan model dalam mendeteksi dengan benar.

$$Precision = \frac{0.93}{0.93 + 0.07} = 0.93$$

$$Recall = \frac{0.93}{0.93 + 0.01} = 0.989$$

$$F1 - Score = 2 \times \frac{0.93 \times 0.989}{0.93 + 0.989} = 0.959$$

b. *Blight*

Pada deteksi penyakit *blight* ini, perhitungan yang telah dilakukan menghasilkan nilai akurasi, *recall*, dan *f1-score* yang menggambarkan kinerja model. Hasil ini menunjukkan seberapa efektif model dalam mengidentifikasi penyakit *blight* pada daun padi. Nilai-nilai tersebut mencerminkan kemampuan model dalam mendeteksi dengan akurat.

$$Precision = \frac{0.99}{0.99 + 0.01} = 0.99$$

$$Recall = \frac{0.99}{0.99 + 0.01} = 0.99$$

$$F1 - Score = 2 \times \frac{0.99 \times 0.99}{0.99 + 0.99} = 0.99$$

c. *Brown spot*

Pada deteksi penyakit *brown spot* ini, dengan perhitungan yang telah diberikan, didapatkan nilai akurasi, *recall*, dan *f1-score* yang mencerminkan kinerja

model. Hasil ini menunjukkan efektivitas model dalam mendeteksi penyakit *brown spot* pada daun padi. Nilai-nilai tersebut menggambarkan akurasi dan kemampuan model dalam mendeteksi dengan benar.

$$Precision = \frac{0.84}{0.84 + 0.16} = 0.84$$

$$Recall = \frac{0.84}{0.84 + 0.01} = 0.988$$

$$F1 - Score = 2 \times \frac{0.84 \times 0.988}{0.84 + 0.988} = 0.909$$

d. *Healthy*

Pada deteksi daun padi yang sehat ini, dengan perhitungan yang telah diberikan, didapatkan nilai akurasi, *recall*, dan *f1-score* yang mencerminkan kinerja model. Hasil ini menunjukkan efektivitas model dalam mendeteksi daun padi yang sehat. Nilai-nilai tersebut menggambarkan akurasi dan kemampuan model dalam mendeteksi dengan benar.

$$Precision = \frac{1.00}{1.00 + 0.00} = 1.00$$

$$Recall = \frac{1.00}{1.00 + 0.00} = 1.00$$

$$F1 - Score = 2 \times \frac{1.00 \times 1.00}{1.00 + 1.00} = 1.00$$

e. *Tungro*

Pada deteksi penyakit *tungro* ini, dengan perhitungan yang telah diberikan, didapatkan nilai akurasi, *recall*, dan *f1-score* yang mencerminkan kinerja model. Hasil ini menunjukkan efektivitas model dalam mendeteksi penyakit *tungro* pada daun padi. Nilai-nilai tersebut menggambarkan akurasi dan kemampuan model dalam mendeteksi dengan benar.

$$Precision = \frac{0.99}{0.99 + 0.01} = 0.99$$

$$Recall = \frac{0.99}{0.99 + 0.01} = 1.00$$

$$F1 - Score = 2 \times \frac{0.99 \times 1.00}{0.99 + 1.00} = 0.995$$

Berdasarkan hasil pencarian diatas, maka hasil pencarian akan disajikan dalam bentuk tabel untuk mempermudah pemahaman data. Penyajian ini memungkinkan analisis yang lebih jelas dan terstruktur. Dengan tabel, pola klasifikasi dan tingkat akurasi model dapat terlihat lebih detail. Selain itu, kesalahan prediksi juga lebih mudah diidentifikasi.

**Tabel 5. 4** Perhitungan Setiap Kelas

<b>Kelas</b>	<b><i>Precision</i></b>	<b><i>Recall</i></b>	<b><i>F1-score</i></b>
<i>Blast</i>	0.93	0.989	0.959
<i>Blight</i>	0.99	0.99	0.99
<i>Brown spot</i>	0.84	0.988	0.909
<i>Healthy</i>	1.00	1.00	1.00
<i>Tungro</i>	0.99	1.00	0.995

Setelah *precision*, *recall*, *f1-score* ditentukan, langkah selanjutnya menghitung nilai *accuracy* secara keseluruhan untuk setiap kelas. Perhitungan *mAP* dilakukan dengan memperhitungkan akurasi prediksi pada setiap kelas secara rata-rata.

$$Accuracy = \frac{0.93 + 0.99 + 0.84 + 1.00 + 0.99}{5} = 0.95$$

$$mAP = \frac{0.93 + 0.99 + 0.84 + 1.00 + 0.99}{5} = 0.95$$

Berdasarkan perhitungan, akurasi yang diperoleh adalah 0.95, menunjukkan ketepatan model dalam deteksi penyakit. Nilai *mean average*

*precision* (mAP) juga mencapai 0.95, menggambarkan kinerja model secara keseluruhan. Hasil ini menunjukkan efektivitas model dalam memberikan prediksi yang tepat. Secara keseluruhan, model ini menunjukkan performa yang sangat baik dalam mendeteksi penyakit pada daun padi.

### 5.1.3. Implementasi Sistem

Dalam proses implementasi sistem, pengguna akan diperkenalkan dengan cara menggunakan *agricare*, mulai dari mengakses halaman utama, mengunggah gambar daun padi, hingga melihat hasil deteksi penyakit. Sistem ini memungkinkan pengguna untuk mengambil gambar daun padi secara *real-time* atau mengunggah gambar melalui fitur *check*, kemudian model *yolo* akan mendeteksi dan mengklasifikasikan jenis penyakit yang ada. Hasil deteksi akan ditampilkan dengan *bounding box* serta tingkat akurasi yang sesuai. Admin dapat mengelola dataset pelatihan untuk meningkatkan akurasi model, sementara pengguna cukup mengunggah gambar untuk mendapatkan hasil analisis secara otomatis. Tampilan dari sistem *agricare* yang telah dibuat dapat dilihat pada gambar berikut ini:

#### a. Halaman Menu

Halaman menu adalah tampilan pertama yang muncul saat aplikasi dibuka dan berfungsi sebagai akses utama ke fitur-fitur yang tersedia. Terdapat perbedaan tampilan navigator antara versi desktop dan *mobile*. Pada desktop menu ditampilkan secara horizontal di bagian atas, sementara pada *mobile*, menggunakan ikon hamburger menu untuk tampilan yang lebih ringkas. Desain ini memastikan tampilan tetap responsif dan nyaman digunakan di berbagai perangkat.



**Gambar 5. 7** Halaman Menu Pada Tampilan Desktop

Pada tampilan desktop, navigator terletak di bagian atas sejajar dengan ikon *Agricare* dan berisi menu *Home, About, Check, Services, dan Contact*. Desain ini memudahkan pengguna untuk mengakses fitur dengan cepat. Semua menu ditampilkan secara horizontal tanpa perlu membuka daftar tambahan. Tampilan ini memastikan navigasi lebih praktis dan efisien.

**Gambar 5. 8** Halaman Menu Pada Tampilan *Mobile* Saat Pertama Dibuka

Saat pertama kali membuka aplikasi dalam tampilan *mobile*, pengguna akan disambut dengan tampilan awal yang menampilkan logo *Agricare*. Logo ini ditampilkan secara penuh di layar sebagai bagian dari splash screen, memberikan kesan profesional dan memastikan aplikasi telah dimuat dengan baik. Setelah beberapa detik, halaman utama akan terbuka, menampilkan menu navigasi dan fitur utama. Desain ini membantu meningkatkan pengalaman pengguna dengan tampilan yang lebih menarik dan responsif.



**Gambar 5. 9** Halaman Menu Pada Tampilan *Mobile*

Pada tampilan *mobile*, navigator terletak di bagian bawah dalam bentuk ikon tombol. Desain ini membuat navigasi lebih ringkas dan mudah dijangkau. Pengguna dapat mengakses menu dengan cepat tanpa mengganggu tampilan utama. Tata letak ini memastikan pengalaman yang lebih nyaman di perangkat *mobile*.



**Gambar 5. 10** Penyakit Pada Daun Padi Pada Halaman Menu

Pada halaman menu penyakit daun padi, informasi disajikan dalam bentuk button group yang dapat diklik untuk menampilkan detail setiap penyakit. Setiap tombol mewakili satu jenis penyakit, dan saat diklik, akan muncul deskripsi singkat serta gambar penyakit tersebut. Fitur ini memudahkan pengguna dalam mengenali penyakit sebelum melakukan deteksi lebih lanjut dengan *agricare*.

## Penyakit Pada Daun Padi

Blast
Blight
Brown Spot
Healthy
Tungro

### Penyakit Blast pada Padi

Penyakit Blast merupakan salah satu penyakit yang paling merusak pada tanaman padi dan dapat menyebabkan kehilangan hasil panen yang signifikan. Penyakit ini disebabkan oleh jamur *Magnaporthe oryzae* (sebelumnya dikenal sebagai *Pyricularia oryzae*), yang menyerang berbagai bagian tanaman, termasuk daun, batang, malai, dan bahkan biji padi.



**Gejala Penyakit Blast**

- Blast Daun (Leaf Blast): Ditandai dengan munculnya bercak berbentuk belah ketupat atau oval, dengan bagian tengah keabu-abuan dan tepi coklat gelap. Jika infeksi semakin parah, bercak-bercak ini dapat menyatu dan menyebabkan daun mengering serta mati.
- Blast Leher (Neck Blast) atau Blast Malai: Menyerang bagian pangkal malai, menyebabkan jaringan berubah menjadi coklat kehitaman. Malai padi yang terinfeksi tidak dapat berkembang sempurna, sehingga butir padi menjadi kosong atau hampa.

**Faktor Penyebab Penyakit Blast**

- Cuaca lembab dan suhu tinggi (sekitar 24-28°C) dengan curah hujan tinggi.
- Pemupukan nitrogen berlebihan, yang membuat tanaman lebih rentan terhadap infeksi.
- Sistem irigasi yang buruk, yang menyebabkan kelembaban tinggi di sekitar tanaman.
- Penggunaan varietas padi yang rentan terhadap penyakit blast.

**Cara Pengendalian Penyakit Blast**

- Menggunakan varietas padi yang tahan blast, seperti Citerang atau Inpari 33.
- Mengatur pola tanam dan rotasi tanaman dengan tanaman lain untuk mengurangi sumber infeksi.
- Mengatur penggunaan pupuk nitrogen, tidak berlebihan agar tidak memicu pertumbuhan jamur.
- Menjaga sanitasi lahan dengan membersihkan sisa-sisa tanaman yang terinfeksi.
- Penggunaan fungisida berbahan aktif seperti triazole atau strobilurin jika serangan sudah parah.

Agar pengguna lebih paham tentang blast serta penanganannya, artikel ini bisa sangat membantu klik [disini](#).

**Gambar 5. 11** Penyakit *Blight* Pada Daun Padi

Pada halaman penyakit daun padi, terdapat tombol *blast* yang dapat diklik untuk menampilkan informasi tentang penyakit blas. Saat tombol diklik, pengguna akan melihat deskripsi singkat, gambar, serta gejala khas penyakit ini, seperti bercak berbentuk belah ketupat pada daun. Fitur ini memudahkan pengguna dalam mengenali penyakit sebelum melakukan deteksi lebih lanjut dengan *agricare*.

## Penyakit Pada Daun Padi

Blast   Blight   Brown Spot   Healthy   Tungro

### Penyakit Blight pada Padi

Penyakit Blight pada padi adalah salah satu penyakit yang dapat menyebabkan kerusakan parah pada tanaman dan menurunkan hasil panen secara signifikan. Penyakit ini disebabkan oleh bakteri *Xanthomonas oryzae* pv. *oryzae*, yang menyerang jaringan tanaman, terutama pada daun dan batang padi. Blight sering muncul di daerah dengan kelembaban tinggi dan dapat menyebar dengan cepat melalui air hujan, angin, dan serangga.



**Gejala Penyakit Blight**

- **Blight Daun (Bacterial Leaf Blight):** Ditandai dengan bercak kuning kecoklatan yang mulai dari tepi daun dan menyebar ke bagian tengah. Daun yang terinfeksi tampak kering dan menggulung, seolah terbakar. Jika infeksi semakin parah, tanaman menjadi layu dan mati lebih awal.
- **Blight Batang:** Menyebabkan batang menjadi berwarna coklat kehitaman dan rapuh. Menghambat aliran nutrisi dalam tanaman, menyebabkan tanaman tumbuh kerdil. Jika terjadi pada fase generatif, dapat mengurangi jumlah dan kualitas bulir padi.

**Faktor Penyebab Penyakit Blight**

- **Cuaca dan Lingkungan:** Kelembaban tinggi dengan curah hujan yang sering dapat meningkatkan penyebaran bakteri.
- **Varietas Rentan:** Penggunaan bibit padi yang tidak tahan terhadap penyakit blight lebih mudah terinfeksi.
- **Irigasi yang Buruk:** Genangan air dapat menjadi media penyebaran bakteri lebih cepat.
- **Serangga Vektor:** Hama seperti wereng dapat membantu penyebaran penyakit dari satu tanaman ke tanaman lainnya.

**Cara Pengendalian Penyakit Blight**

- Menggunakan varietas padi tahan blight, seperti Inpari 32 atau IR64.
- Mengatur sistem irigasi agar tidak ada genangan air yang dapat mempercepat penyebaran bakteri.
- Rotasi tanaman untuk mengurangi risiko akumulasi bakteri di lahan pertanian.
- Pembersihan lahan dengan membuang sisa tanaman yang terinfeksi untuk mencegah penyebaran.
- Penggunaan bakterisida berbahan aktif tembaga atau antibiotik agrikultural untuk mengendalikan serangan jika sudah parah.

Untuk memahami lebih lanjut tentang penyakit blight dan cara menanganinya, artikel ini bisa sangat membantu [klik disini](#).

**Gambar 5. 12** Penyakit *Blast* Pada Daun Padi

Pada halaman penyakit daun padi, saat tombol *blight* diklik, pengguna dapat melihat informasi tentang hawar daun bakteri yang ditandai dengan bercak kuning kecoklatan yang menyebar. Tampilan ini memudahkan pengguna memahami ciri penyakit sebelum melakukan pemeriksaan lebih lanjut.

**Penyakit Pada Daun Padi**

Blast    Blight    Brown Spot    Healthy    Tungro

### Penyakit Brown Spot pada Padi

Penyakit Brown Spot atau bercak coklat pada padi adalah salah satu penyakit yang sering menyerang tanaman padi dan dapat menyebabkan penurunan hasil panen yang signifikan. Penyakit ini disebabkan oleh jamur *Helminthosporium oryzae* (atau *Drechslera oryzae*), yang berkembang dengan cepat di kondisi lingkungan yang lembap dan kurang nutrisi. Brown Spot dapat merusak daun, batang, dan bahkan gabah, mengurangi kualitas hasil panen.



**Gejala Penyakit Brown Spot**

- Bercak Coklat pada Daun: Munculnya bercak-bercak kecil berbentuk oval atau bulat dengan warna coklat tua di permukaan daun. Bercak berkembang menjadi lebih besar dengan bagian tengah berwarna abu-abu dan pinggirannya coklat tua. Jika infeksi parah, daun menjadi kering dan tanaman tampak merana.
- Infeksi pada Gabah: Gabah yang terinfeksi memiliki bercak berwarna coklat kehitaman. Mengurangi kualitas dan daya kecambah benih.
- Pertumbuhan Tanaman Terhambat: Tanaman padi yang terserang mengalami pertumbuhan yang lambat dan kurang produktif. Bisa menyebabkan "busuk leher" jika menyerang batang saat fase berbunga.

**Faktor Penyebab Penyakit Brown Spot**

- Kelembaban Tinggi: Kondisi lingkungan yang lembap dan suhu hangat (sekitar 16-36°C) mempercepat perkembangan jamur.
- Tanah Kurang Nutrisi: Kekurangan unsur hara, terutama nitrogen dan kalium, meningkatkan kerentanan tanaman terhadap penyakit.
- Sistem Irigasi Buruk: Air yang menggenang dapat menjadi media penyebaran spora jamur.
- Penggunaan Benih yang Terinfeksi: Benih yang sudah terkontaminasi jamur dapat menjadi sumber utama penyebaran penyakit.

**Cara Pengendalian Penyakit Brown Spot**

- Penggunaan Varietas Tahan: Pilih varietas padi yang memiliki ketahanan terhadap penyakit brown spot.
- Pemupukan Seimbang: Gunakan pupuk dengan kandungan nitrogen, fosfor, dan kalium yang cukup untuk meningkatkan daya tahan tanaman.
- Rotasi Tanaman: Mengganti jenis tanaman di lahan pertanian dapat membantu mengurangi penyebaran jamur.
- Penyemprotan Fungisida: Gunakan fungisida berbahan aktif mancozeb atau propiconazole jika serangan sudah meluas.
- Pembersihan Lahan: Singkirkan sisa-sisa tanaman yang terinfeksi untuk mencegah penyebaran spora ke musim tanam berikutnya.

Untuk memahami lebih lanjut tentang penyakit brown spot dan cara menanganinya, artikel ini bisa sangat membantu [klik disini](#).

**Gambar 5. 13** Penyakit *Brown spot* Pada Daun Padi

Pada halaman penyakit daun padi saat tombol *brown spot* diklik, muncul deskripsi tentang bercak coklat, yaitu bintik kecil berbentuk bulat hingga oval berwarna coklat tua pada daun. Fitur ini membantu pengguna mengenali penyakit yang dapat mengurangi produktivitas tanaman.

**Penyakit Pada Daun Padi**

Blast    Blight    Brown Spot    Healthy    Tungro

**Kondisi Padi Sehat (Healthy)**

Padi yang sehat adalah tanaman yang tumbuh optimal tanpa gangguan dari hama, penyakit, atau kekurangan nutrisi. Tanaman padi yang sehat akan memiliki pertumbuhan yang baik, daun hijau segar, batang kuat, serta menghasilkan gabah berkualitas tinggi dengan hasil panen maksimal. Padi yang sehat juga lebih tahan terhadap perubahan cuaca dan faktor lingkungan yang kurang mendukung.



**Ciri-ciri Padi Sehat**

- Daun Hijau dan Segar: Daun padi berwarna hijau cerah tanpa adanya bercak kuning, coklat, atau hitam. Tidak terdapat tanda-tanda pengeringan atau keriting pada tepi daun.
- Batang Kokoh dan Tidak Busuk: Batang padi tegak lurus, tidak mudah roboh, dan tidak menunjukkan gejala busuk pangkal batang. Tidak ada bercak hitam atau coklat yang menandakan infeksi jamur atau bakteri.
- Akar yang Kuat dan Sehat: Akar berkembang dengan baik dan tidak menunjukkan tanda-tanda pembusukan atau warna kecoklatan yang tidak normal. Mampu menyerap nutrisi dengan baik dari tanah.
- Pertumbuhan Optimal: Padi tumbuh dengan tinggi yang merata dan tidak mengalami kerdil. Tidak menunjukkan gejala layu atau menguning akibat kekurangan unsur hara.
- Gabah Berkualitas Tinggi: Gabah padi memiliki ukuran normal, tidak keriput, dan berisi penuh. Tidak terdapat bercak hitam atau coklat pada gabah yang menandakan infeksi penyakit.

**Faktor yang Mendukung Padi Sehat**

- Pengelolaan Tanah yang Baik: Menggunakan sistem drainase yang baik untuk menghindari genangan air yang dapat memicu pertumbuhan jamur dan bakteri. Pengolahan tanah yang optimal sebelum tanam untuk memastikan nutrisi cukup.
- Pemupukan Seimbang: Menggunakan pupuk organik dan anorganik sesuai kebutuhan tanaman. Pastikan ketersediaan nitrogen (N), fosfor (P), dan kalium (K) dalam jumlah yang cukup.
- Sistem Irigasi yang Tepat: Air harus cukup, tetapi tidak berlebihan untuk menghindari pembusukan akar dan penyebaran penyakit. Irigasi teratur untuk menjaga kelembaban tanah sesuai kebutuhan tanaman.
- Pengendalian Hama dan Penyakit: Melakukan pengamatan rutin terhadap tanaman untuk mendeteksi gejala awal penyakit atau hama. Menggunakan pestisida atau fungisida sesuai kebutuhan dengan dosis yang tepat. Menanam varietas yang tahan terhadap penyakit dan hama tertentu.
- Rotasi Tanaman: Melakukan penggiliran tanaman dengan jenis lain (misalnya kacang-kacangan) untuk menjaga kesuburan tanah dan mengurangi akumulasi hama serta penyakit.
- Penanaman dengan Jarak Ideal: Menanam padi dengan jarak yang tidak terlalu rapat agar sirkulasi udara berjalan baik dan kelembaban tidak terlalu tinggi.

Untuk memahami lebih lanjut tentang kondisi padi sehat dan cara menanganinya, artikel ini bisa sangat membantu [klik disini](#).

**Gambar 5. 14** Padi *Healthy* Pada Daun Padi

Pada halaman penyakit daun padi saat tombol *healthy* diklik, pengguna akan melihat contoh daun padi yang sehat, berwarna hijau merata tanpa gejala penyakit. Informasi ini berguna untuk membandingkan kondisi daun sebelum melakukan deteksi lebih lanjut.

## Penyakit Pada Daun Padi

Blast
Blight
Brown Spot
Healthy
Tungro

### Penyakit Tungro pada Padi

Penyakit Tungro adalah salah satu penyakit virus yang paling merusak pada tanaman padi dan dapat menyebabkan penurunan hasil panen yang drastis. Penyakit ini disebabkan oleh dua jenis virus, yaitu Rice Tungro Bacilliform Virus (RTBV) dan Rice Tungro Spherical Virus (RTSV), yang ditularkan oleh wereng hijau (*Neophotix virescens*). Tungro sangat umum terjadi di daerah tropis dan menjadi ancaman serius bagi produksi padi.



**Gejala Penyakit Tungro**

- Daun Menguning: Daun muda berubah menjadi kuning oranye atau kuning kemerahan, dimulai dari ujung dan menyebar ke pangkal. Warna kuning lebih mencolok pada daun yang lebih tua.
- Pertumbuhan Tanaman Terhambat: Tanaman yang terserang mengalami kerdil dan pertumbuhan tidak normal. Pelepah daun menjadi lebih pendek dan batang melemah.
- Produksi Bulir Berkurang: Malai padi yang terinfeksi menjadi kecil dan menghasilkan sedikit bulir. Banyak bulir yang hampa atau tidak terti sempurna.
- Infeksi Berat: Jika infeksi terjadi pada fase awal pertumbuhan, tanaman bisa gagal berproduksi sepenuhnya. Penyebaran penyakit yang luas dapat menyebabkan gagal panen di suatu area pertanian.

**Faktor Penyebab Penyakit Tungro**

- Hama Wereng Hijau: Vektor utama penyakit ini adalah wereng hijau yang menyebarkan virus dari satu tanaman ke tanaman lain.
- Varietas Rentan: Padi yang tidak tahan terhadap virus tungro lebih mudah terinfeksi.
- Lingkungan Lembap: Kondisi iklim yang lembap dan hangat mendukung perkembangbiakan wereng hijau.
- Monokultur: Penanaman padi secara terus-menerus tanpa rotasi tanaman meningkatkan risiko penyebaran tungro.

**Cara Pengendalian Penyakit Tungro**

- Penggunaan Varietas Tahan: Pilih varietas padi yang tahan tungro seperti Inpari 36 atau Inpari 37.
- Pengendalian Wereng Hijau: Lakukan pengamatan rutin dan gunakan insektisida jika populasi wereng meningkat.
- Rotasi Tanaman: Tanam tanaman selain padi setelah panen untuk mengurangi populasi wereng.
- Sanitasi Lahan: Singkirkan tanaman yang terinfeksi untuk mencegah penyebaran lebih lanjut.
- Penyemprotan Insektisida: Gunakan insektisida berbahan aktif imidakloprid atau tiametoksam untuk mengendalikan hama vektor.

Untuk memahami lebih lanjut tentang penyakit tungro dan cara menanganinya, artikel ini bisa sangat membantu klik disini.

**Gambar 5. 15** Penyakit *Tungro* Pada Daun Padi

Pada halaman penyakit daun padi, saat tombol *tungro* diklik, pengguna akan melihat ciri-ciri penyakit *tungro*, seperti daun menguning, pertumbuhan terhambat, dan tanaman menjadi kerdil. Fitur ini membantu pengguna dalam mengenali penyakit yang disebabkan oleh virus *tungro*.

## FAQ: Pertanyaan Seputar AgriCare



Apa sih AgriCare itu? >

Kenapa AgriCare keren buat petani muda? >

Gratis atau bayar nih? >

Seberapa jago AI di AgriCare? >

**Gambar 5. 16** FAQ Pertanyaan Seputar *AgriCare*

Pada halaman faq *agricare* pengguna dapat melihat daftar pertanyaan dan jawaban seputar penggunaan *agricare*, termasuk cara kerja, jenis penyakit yang dideteksi, akurasi, dan kompatibilitas perangkat. Tampilan ini memudahkan pengguna memahami fitur *agricare* sebelum menggunakannya.



**Gambar 5. 17** Salah Satu FAQ Pertanyaan Seputar *Agricare*

Pada halaman *check* salah satu tampilannya dapat dilihat pada gambar di atas. Ketika tanda siku diklik, jawaban dari pertanyaan akan muncul. Hal yang sama berlaku untuk pertanyaan berikutnya, di mana jawaban akan tampil dengan cara yang sama saat tanda siku diklik.

b. Halaman *About*

Halaman *About* adalah halaman yang berisi informasi tentang *agricare*, termasuk tujuan, fungsi, dan manfaat dari aplikasi ini. Pada halaman ini, dijelaskan bagaimana *agricare* membantu pengguna dalam mendeteksi penyakit pada daun padi secara cepat dan akurat menggunakan teknologi *computer vision* berbasis *YOLO*. Selain itu, terdapat informasi mengenai metode yang digunakan, keunggulan aplikasi, serta tim pengembang di balik proyek ini. Halaman ini bertujuan untuk memberikan pemahaman lebih mendalam kepada pengguna tentang peran dan manfaat *agricare* dalam dunia pertanian.



**Gambar 5. 18** Halaman *About*

Tampilan pertama halaman *About* menampilkan gambar dengan teks singkat yang menjelaskan tujuan dan manfaat *agricare*. Gambar ini mendukung pemahaman visual, sementara teks menjelaskan peran aplikasi dalam mendeteksi penyakit daun padi secara cepat dan akurat.



**Gambar 5. 19** *About Agricare*

Tampilan selanjutnya pada halaman *about* menampilkan *about agricare*, yang berisi penjelasan tentang apa itu *agricare*. Bagian ini memberikan gambaran singkat mengenai tujuan dan fungsi *agricare* dalam mendeteksi penyakit pada daun padi secara akurat dan efisien.

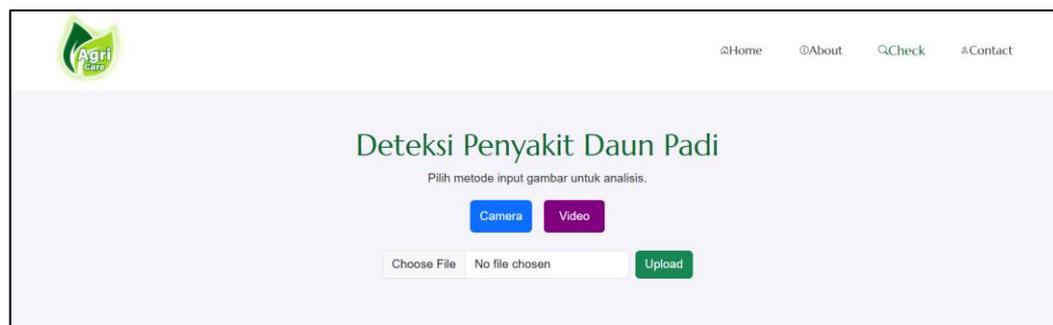


**Gambar 5. 20** Tutorial Penggunaan Aplikasi

Pada halaman menu terdapat Tutorial Penggunaan Aplikasi yang berisi panduan atau petunjuk cara menggunakan *agricare*. Pengguna akan diarahkan ke video tutorial di youtube untuk memahami fitur dan cara kerja aplikasi dengan lebih mudah. Tutorial ini membantu pengguna, terutama yang baru pertama kali menggunakan aplikasi. Dengan panduan ini, pengguna dapat memaksimalkan manfaat dari *agricare* secara efektif.

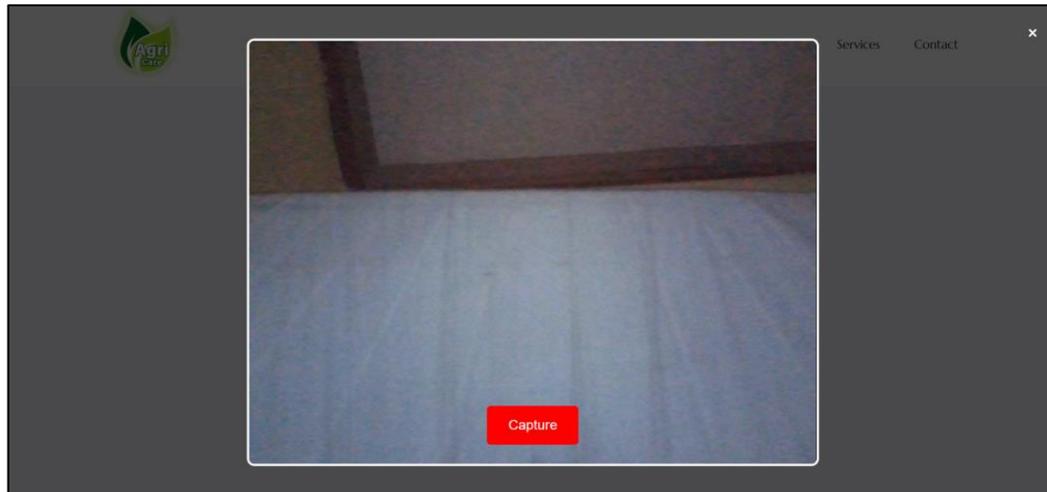
c. Halaman *Check*

Halaman *check* adalah halaman utama dalam *agricare* yang digunakan untuk mendeteksi penyakit pada daun padi. Pengguna dapat mengunggah gambar atau menggunakan kamera secara *real-time* untuk melakukan analisis. Model *yolo* akan memproses gambar dan menampilkan hasil deteksi berupa nama penyakit, *bounding box*, serta tingkat akurasi.



**Gambar 5. 21** Halaman *Check*

Halaman *check* menyediakan tiga fitur yaitu fitur *camera*, *video* dan *upload* file yang bisa digunakan untuk mendeteksi penyakit pada daun padi. Fitur ini dirancang agar mudah digunakan, memungkinkan petani dan pengguna lain mendapatkan informasi kesehatan tanaman dengan cepat dan akurat.



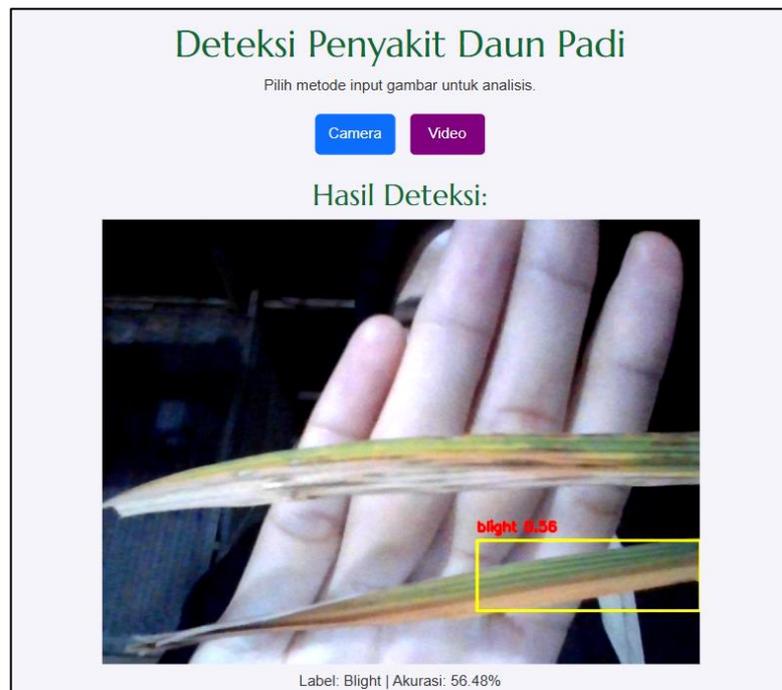
**Gambar 5. 22** Fitur *Camera*

*Camera* digunakan untuk analisis gambar statis. Dalam aplikasi ini, kamera memproses gambar secara langsung untuk mendeteksi kondisi daun padi. Sistem bekerja dengan menganalisis citra tanpa perlu unggahan manual, meningkatkan efisiensi pengguna. Dengan fitur ini, identifikasi penyakit menjadi lebih cepat dan praktis.



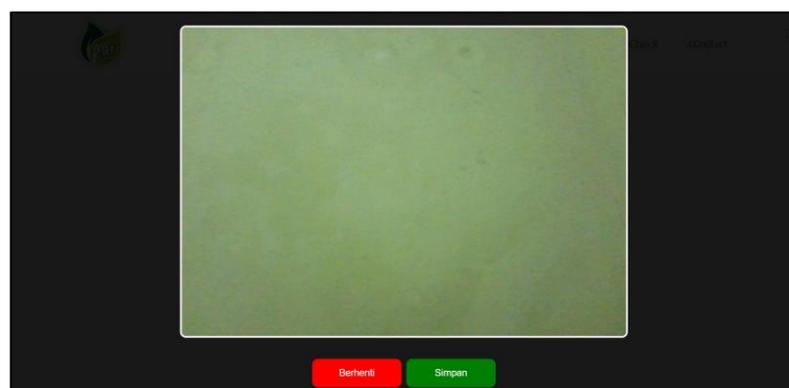
**Gambar 5. 23** Camera Berhasil Di *Capture*

Pada halaman *check*, *camera* berfungsi untuk mengambil tangkapan layar (*capture*). Setelah gambar diambil, sistem akan mendeteksi penyakit dengan menampilkan *bounding box*, nama penyakit, dan akurasi deteksi. Fitur ini memungkinkan pengguna menyimpan hasil analisis secara otomatis tanpa perlu mengunggah gambar secara manual, membuat proses lebih cepat dan praktis.



**Gambar 5. 24** Daun Padi Di *Capture* Secara Langsung

Daun padi dapat di *capture* secara langsung menggunakan *camera*. Setelah itu, sistem akan menganalisis gambar untuk mengidentifikasi jenis penyakitnya. Proses ini dilakukan secara *real-time* menggunakan model *yolo* untuk deteksi yang akurat.



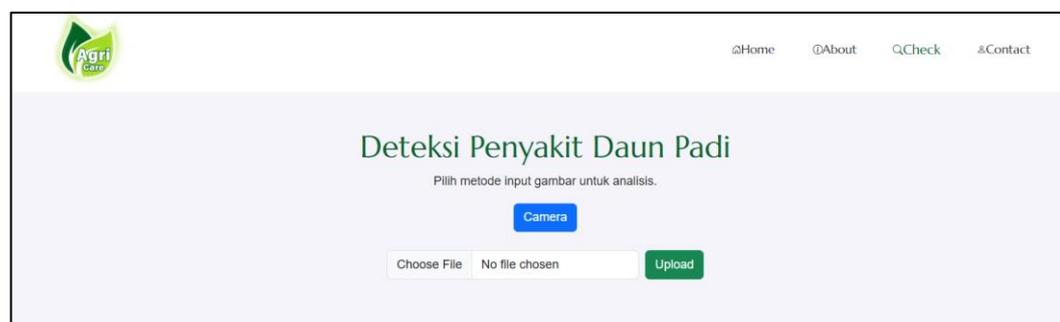
**Gambar 5. 25** Fitur Video

Video berfungsi untuk mengambil dan menganalisis gambar daun padi secara *real-time* menggunakan model *yolo*. Video mempunyai dua button yaitu berhenti dan simpan. Sistem akan mendeteksi penyakit dengan menampilkan *bounding box*, nama penyakit, dan akurasi deteksi, tanpa perlu mengunggah gambar manual. Fitur ini membuat proses lebih cepat dan praktis.



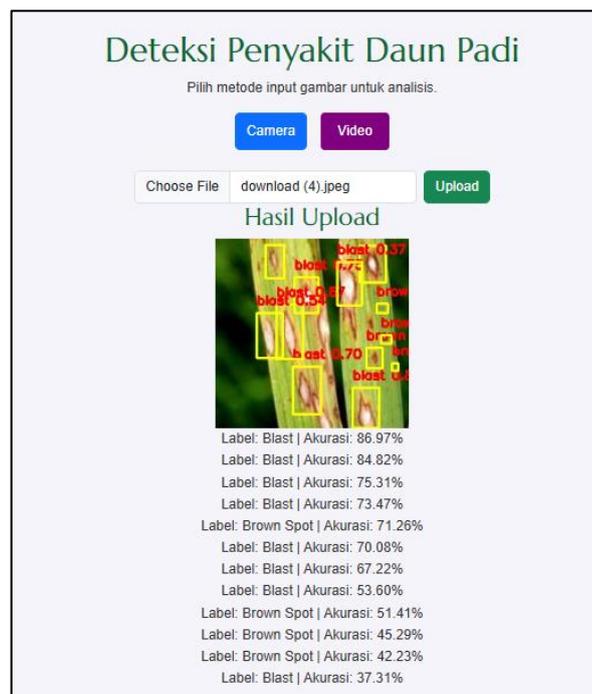
**Gambar 5. 26** Video Berhasil Di Deteksi

Pada halaman *check*, video berfungsi untuk menganalisis gambar daun padi secara *real-time*. Setelah gambar diambil, sistem akan mendeteksi penyakit dengan menampilkan *bounding box*, nama penyakit, dan akurasi deteksi. Fitur ini memudahkan pengguna dalam menyimpan hasil analisis tanpa perlu mengunggah gambar secara manual.



**Gambar 5. 27** Fitur *Upload*

Fitur *upload* memungkinkan pengguna mengunggah gambar daun padi untuk dianalisis oleh model *yolo*. Dengan fitur ini, pengguna dapat memanfaatkan gambar yang sudah tersedia tanpa harus mengambil foto secara langsung melalui kamera. Proses analisis dilakukan secara otomatis untuk memberikan hasil yang cepat dan efisien.



**Gambar 5. 28** Gambar Berhasil Di *Upload*

Setelah gambar diunggah, sistem akan memprosesnya dan menampilkan hasil deteksi berupa *bounding box*, nama penyakit, serta akurasi deteksi. Fitur ini berguna bagi pengguna yang ingin menganalisis gambar yang sudah mereka miliki tanpa harus menggunakan kamera secara *real-time*.

d. Halaman *Contact*

Halaman *contact* berisi informasi yang memungkinkan pengguna untuk menghubungi tim *agricare*. Halaman ini mencakup alamat, email, nomor telepon, serta formulir kontak bagi pengguna yang ingin mengirimkan pertanyaan atau masukan. Selain itu, terdapat tautan ke media sosial agar pengguna dapat mengikuti perkembangan dan mendapatkan informasi terbaru tentang *agricare*.



**Gambar 5. 29** Halaman *Contact*

Tampilan pertama halaman *contact* menampilkan foto dan teks singkat yang mengajak pengguna untuk menghubungi *agricare* untuk informasi lebih lanjut. Desain ini dibuat agar jelas dan menarik, memudahkan pengguna dalam berkomunikasi dengan tim.



**Gambar 5. 30** *Developer*

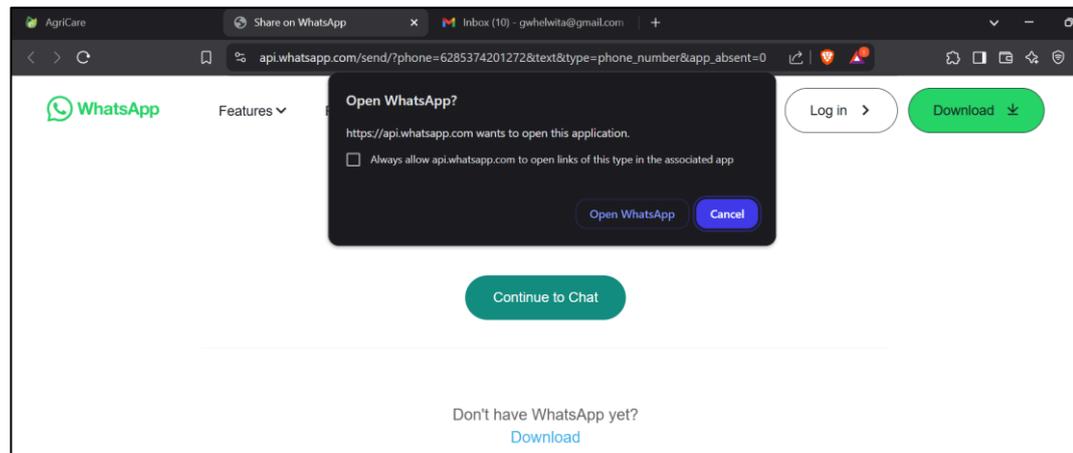
Tampilan selanjutnya dari halaman *Contact* menyajikan informasi singkat tentang *developer Agricare*. Bagian ini menampilkan nama pengembang serta perannya dalam pembuatan aplikasi. Informasi ini bertujuan untuk memberikan transparansi kepada pengguna mengenai siapa yang mengembangkan *Agricare*.



**Gambar 5. 31** Whatsapp Dan Email

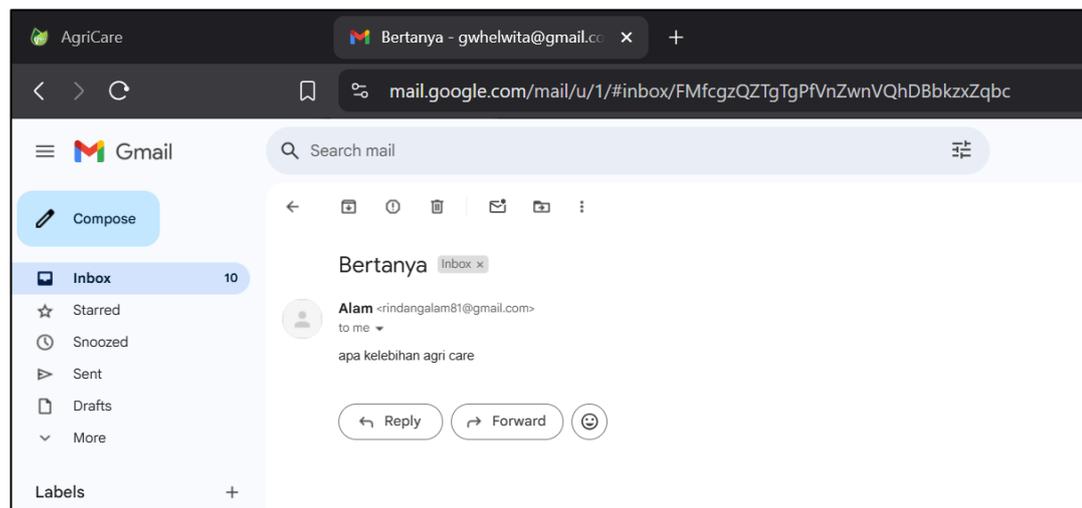
Tampilan selanjutnya menampilkan whatsapp dan email dari pengembangan *agricare* bagi pengguna yang ingin menghubungi *agricare* untuk mendapatkan informasi lebih lanjut atau memberikan saran. Informasi ini

memudahkan pengguna dalam menjangkau pengembang jika memiliki pertanyaan, masukan, atau kebutuhan lainnya terkait aplikasi.



**Gambar 5. 32** Pengguna Diarahkan Ke Whatsapp

Ketika pengguna mengklik ikon atau tombol whatsapp, mereka akan diarahkan langsung ke whatsapp untuk menghubungi tim *agricare*. Di sana, mereka dapat menanyakan informasi terkait fitur *agricare*, seperti cara mendeteksi penyakit daun padi, interpretasi hasil deteksi, serta panduan penggunaan platform. Dengan layanan ini, pengguna dapat memperoleh solusi cepat dan akurat terkait kesehatan tanaman padi mereka.



**Gambar 5. 33** Email Yang Dikirim Pengguna

Ketika pengguna mengklik ikon atau tombol email, mereka akan diarahkan ke email untuk menghubungi tim *agricare*. Mereka dapat menanyakan fitur

*agricare*, seperti deteksi penyakit daun padi, interpretasi hasil, dan panduan penggunaan. Dengan layanan ini, pengguna memperoleh solusi cepat dan akurat terkait kesehatan tanaman padi.

## 5.2. Pengujian

Pengujian sistem *Agricare* menggunakan *black box testing*, pengujian difokuskan pada aspek fungsionalitas aplikasi tanpa melihat kode sumber. Tujuan utama dari metode ini adalah untuk memastikan bahwa setiap fitur berjalan sesuai dengan yang diharapkan berdasarkan spesifikasi dan kebutuhan sistem.

### 5.2.1. Pengujian Kotak Hitam (*Black Box Testing*)

Pengujian *black box testing* pada aplikasi *agricare* dilakukan untuk menguji fungsionalitas sistem tanpa melihat kode sumber. Pengujian ini berfokus pada bagaimana pengguna berinteraksi dengan aplikasi dan memastikan setiap fitur bekerja sesuai yang diharapkan.

#### a. Pengujian Halaman Menu

Halaman *Home* merupakan halaman utama aplikasi *Agricare* yang memberikan gambaran umum tentang tujuan dan fungsi utama aplikasi. Halaman ini juga menjadi titik awal navigasi bagi pengguna untuk mengakses fitur lainnya, seperti informasi tentang *Agricare*, pengecekan penyakit daun padi, serta kontak pengembang.

**Tabel 5. 5** Pengujian Halaman Menu

No	Skenario uji	Input	Ekspetasi output	Status
1	Menampilkan splash screen	Buka halaman utama	Splash screen muncul dan hilang dalam beberapa detik	Berhasil
2	Navigasi menu berfungsi	Klik menu <i>about</i> , <i>check</i> , <i>contact</i>	Pindah ke halaman yang sesuai	Berhasil
3	Carousel berjalan otomatis	Perhatikan slider gambar di hero section	Slider berjalan otomatis dan bisa diubah manual	Berhasil

b. Pengujian Halaman *About*

Halaman *About* berisi informasi mengenai *Agricare*, termasuk tujuan pengembangan dan cara kerja aplikasi dalam mendeteksi penyakit pada daun padi. Halaman ini bertujuan untuk memberikan pemahaman lebih mendalam kepada pengguna tentang manfaat dan teknologi yang digunakan dalam *Agricare*.

**Tabel 5. 6** Pengujian Halaman *About*

No	Skenario uji	Input	Ekspetasi output	Status
1	Menampilkan deskripsi <i>agricare</i>	Buka halaman <i>about</i>	Deskripsi muncul dengan jelas	Berhasil
2	Navigasi menu berfungsi	Klik menu lain dari halaman <i>about</i>	Pindah ke halaman yang sesuai	Berhasil
3	Video tutorial bisa diputar	Klik tombol play di video tutorial	Video tutorial berjalan	Berhasil

c. Pengujian Halaman *Check*

Halaman *Check* merupakan fitur utama dari aplikasi *Agricare*, di mana pengguna dapat mengunggah gambar daun padi atau menggunakan kamera secara langsung untuk mendeteksi penyakit. Sistem kemudian akan menganalisis gambar dan menampilkan hasil identifikasi beserta tingkat akurasi.

**Tabel 5. 7** Pengujian Halaman *Check*

No	Skenario uji	Input	Ekspetasi output	Status
1	Tombol kamera berfungsi	Klik tombol "camera"	Kamera terbuka untuk mengambil gambar	Berhasil
2	Tombol <i>upload</i> gambar berfungsi	Klik tombol " <i>upload</i> " setelah memilih file	Gambar dikirim dan hasil deteksi muncul	Berhasil

3	Hasil deteksi muncul dengan akurasi	<i>Upload</i> gambar daun padi yang sudah diketahui hasilnya	Hasil deteksi sesuai dengan penyakit padi yang diunggah	Berhasil
---	-------------------------------------	--	---	----------

d. Pengujian Halaman *Contact*

Halaman *Contact* menyediakan informasi kontak pengembang *Agricare*, sehingga pengguna dapat menghubungi pengembang untuk mendapatkan informasi lebih lanjut atau memberikan masukan. Halaman ini juga menampilkan profil pengembang serta tautan ke media sosial terkait.

**Tabel 5. 8** Pengujian Halaman Menu

No	Skenario uji	Input	Ekspetasi output	Status
1	Informasi <i>developer</i> muncul	Buka halaman <i>contact</i>	Informasi <i>developer</i> dan kontak tampil	Berhasil
2	Link whatsapp bisa dibuka	Klik tombol whatsapp	Dialihkan ke chat whatsapp	Berhasil
3	Link email berfungsi	Klik tombol email	Dialihkan ke aplikasi email	Berhasil

## BAB VI PENUTUP

### 6.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa pengembangan aplikasi deteksi penyakit pada daun padi menggunakan *computer vision* berbasis python dan *yolo* telah berhasil diterapkan. Proses pengembangan mengikuti tahapan sistematis untuk memastikan model bekerja dengan optimal. Kesimpulan dari hasil penelitian ini adalah sebagai berikut:

- a. Aplikasi *agricare* telah berhasil dikembangkan yaitu sebuah aplikasi berbasis website yang mampu mendeteksi penyakit daun padi menggunakan *computer vision* berbasis python dan *yolo*. Aplikasi ini dirancang dengan antarmuka berbasis flask dan bootstrap, memungkinkan pengguna untuk mengunggah gambar atau menggunakan kamera secara *real-time* untuk mendeteksi penyakit padi dengan akurasi tinggi.
- b. *Computer vision* berbasis python dan *yolo* diterapkan melalui serangkaian tahapan yang sistematis, yaitu problem scoping, data acquisition, data exploration, modelling, *evaluation*, dan deployment. Dataset diperoleh dari *roboflow*, kemudian diproses melalui tahapan pelabelan dan pelatihan menggunakan *google colab*. Hasil pengujian menunjukkan bahwa model yang dikembangkan memiliki akurasi 95% dengan nilai *map* 95%, menandakan efektivitas sistem dalam mengklasifikasikan penyakit daun padi.
- c. Aplikasi *agricare* memberikan solusi bagi pengguna, khususnya petani, untuk mengidentifikasi penyakit tanaman padi secara mandiri tanpa memerlukan alat khusus. Dengan deteksi berbasis *bounding box* dan probabilitas kelas, pengguna dapat mengetahui jenis penyakit yang menyerang tanaman secara cepat, akurat, dan *real-time*. Implementasi teknologi ini membantu meningkatkan efisiensi pertanian, mengurangi risiko gagal panen akibat kesalahan diagnosis, serta memberikan informasi yang mudah diakses bagi petani.

## 6.2. Keterbatasan Sistem

*Agricare* telah menunjukkan hasil yang baik dalam mendeteksi penyakit daun padi dengan tingkat akurasi yang tinggi dan proses identifikasi yang cepat, sistem ini masih memiliki beberapa keterbatasan yang perlu diperbaiki dan dikembangkan lebih lanjut agar dapat memberikan hasil yang lebih optimal dan cakupan deteksi yang lebih luas. Beberapa keterbatasan yang ada pada *agricare* dapat dilihat sebagai berikut:

- a. Sistem hanya dapat mendeteksi lima jenis penyakit pada daun padi, yaitu *blast*, *blight*, *brown spot*, *healthy*, dan *tungro*, sehingga belum mencakup semua jenis penyakit yang dapat menyerang tanaman padi.
- b. Akurasi model dipengaruhi oleh kualitas gambar yang diunggah, terutama dalam kondisi pencahayaan yang kurang baik atau gambar yang buram, yang dapat menurunkan performa deteksi.
- c. *Agricare* masih bergantung pada dataset yang tersedia di *roboflow*, sehingga model belum diuji dengan dataset yang lebih bervariasi dari berbagai kondisi lingkungan pertanian.
- d. Aplikasi memerlukan koneksi internet untuk melakukan deteksi penyakit, karena model di-host secara online, sehingga tidak dapat digunakan secara optimal di daerah dengan keterbatasan jaringan internet.
- e. Sistem belum memiliki fitur rekomendasi penanganan penyakit secara otomatis, sehingga pengguna masih perlu mencari informasi tambahan mengenai cara mengatasi penyakit yang terdeteksi.

## 6.3. Saran

Pengembangan ini bertujuan untuk meningkatkan akurasi, fleksibilitas, dan cakupan sistem dalam mendeteksi penyakit daun padi. Selain itu, optimasi sistem diperlukan agar lebih mudah digunakan oleh petani di berbagai kondisi lingkungan. Dengan perbaikan ini, *agricare* dapat menjadi solusi yang lebih efektif dalam membantu sektor pertanian.

- a. Sistem harus dikembangkan agar dapat mendeteksi lebih banyak penyakit daun padi. Hal ini dapat dilakukan dengan memperluas dataset yang

digunakan. Kolaborasi dengan pakar pertanian juga diperlukan untuk meningkatkan akurasi.

- b. Model perlu lebih adaptif terhadap kualitas gambar yang bervariasi. Teknik preprocessing seperti peningkatan kontras dan normalisasi warna dapat diterapkan. Hal ini akan membantu dalam meningkatkan akurasi deteksi.
- c. Dataset harus mencakup gambar dari berbagai kondisi pertanian. Data yang lebih beragam akan membuat model lebih robust dalam mendeteksi penyakit. Pengujian lebih lanjut diperlukan untuk memastikan performa optimal.
- d. Aplikasi sebaiknya dapat berjalan tanpa koneksi internet. Model *yolo* dapat dioptimalkan agar bisa digunakan secara lokal di perangkat pengguna. Dengan demikian, sistem tetap dapat digunakan di daerah dengan jaringan terbatas.
- e. Sistem perlu memberikan rekomendasi penanganan penyakit secara otomatis. Basis data penyakit dan solusi pertanian dapat ditambahkan untuk membantu pengguna. Integrasi ai atau *chatbot* juga dapat meningkatkan pengalaman pengguna.

## DAFTAR PUSTAKA

- Agustiani, S., Arifin, Y. T., Junaidi, A., Wildah, S. K., & Mustopa, A. (2022). Klasifikasi Penyakit Daun padi Menggunakan random forest dan color histogram. *Jurnal Komputasi*, 10(1), 65-74. <https://doi.org/10.23960/komputasi.v10i1.2961>.
- Aini, Q., Lutfiani, N., Kusumah, H., & Zahran, M. S. (2021). Deteksi dan Pengenalan Objek Dengan Model Machine Learning: Model *Yolo*. *CESS (Journal of Computer Engineering, System and Science)*, 6(2), 192. <https://doi.org/10.24114/cess.v6i2.25840>.
- Anwar, M., Kristian, Y., & Setyati, E. (2023). Klasifikasi penyakit tanaman cabai rawit dilengkapi dengan segmentasi citra daun dan buah menggunakan *YOLO v7*. *INTECOMS: Journal of Information Technology and Computer Science*, 6(1), 540-548. <https://doi.org/10.31539/intecom.s.v6i1.6071>.
- Arifin, N., Insani, C. N., & Rasyid, M. R. (2023). Klasifikasi Tingkat Kematangan Buah Tomat menggunakan *Computer vision* untuk Smart Agriculture. *Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer)*, 22(2), 509-516. <https://doi.org/10.53513/jis.v22i2.8387>.
- Ayuni, D. P., Irsyad, M., Yanto, F., & Sanjaya, S. (2023). Augmentasi Data Pada Implementasi Convolutional Neural Network Arsitektur Efficientnet-B3 Untuk Klasifikasi Penyakit Daun Padi. *ZONAsi: Jurnal Sistem Informasi*, 5(2), 239-249. <https://doi.org/10.31849/zn.v5i2.13874>.
- BPS-Statistics Indonesia. (2024). *Executive Summary: Paddy Harvested Area and Production in Indonesia 2023 (Final Figures)*. 1–54. <https://www.bps.go.id/id/publication/2024/05/06/69834d72f7ef1c32eee5c4b6/ringkasan-eksekutif-luas-panen-dan-produksi-padi-di-indonesia-2023--angka-tetap-.html>

- Fauzi, M. D. M., Al Mudzakir, T., Sukmawati, C. E., & Indra, J. (2024). Deteksi Jenis Penyakit Pada Tanaman Padi Menggunakan *Yolo V5*. *KLIK: Kajian Ilmiah Informatika dan Komputer*, 5(1), 39-48. <https://doi.org/10.30865/klik.v5i1.2009>.
- Hartati, S. (2021). *Kecerdasan Buatan Berbasis Pengetahuan*. Ugm Press.
- Hasan, N. F. (2023). Deteksi dan Klasifikasi Penyakit Pada Daun Kopi Menggunakan *Yolov7*. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 12(1), 30-35. <https://doi.org/10.32736/sisfokom.v12i1.1545>.
- Hawari, F. H., Fadillah, F., Alviandi, M. R., & Arifin, T. (2022). Klasifikasi Penyakit Tanaman Padi Menggunakan Algoritma Cnn (Convolutional Neural Network). *Jurnal Responsif: Riset Sains Dan Informatika*, 4(2), 184-189. <https://doi.org/10.51977/jti.v4i2.856>.
- Ibrahim, M., & Latifa, U. (2023). PENERAPAN ALGORITMA *YOLOV8* DALAM DETEKSI WAKTU PANEN TANAMAN PAKCOY BERBASIS WEBSITE. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(4), 2489-2495. <https://doi.org/10.36040/jati.v7i4.7154>.
- Julianto, B., Farida, I. N., & Dara, M. A. D. W. (2023). Implementasi Metode CNN Pada Aplikasi Android Untuk Deteksi Penyakit Pada Daun Padi. *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, 7(2), 963–970. <https://doi.org/10.29407/inotek.v7i2.3522>.
- Kaul, V., Enslin, S., & Gross, S. A. (2020). History of artificial intelligence in medicine. *Gastrointestinal endoscopy*, 92(4), 807-812. <https://doi.org/10.1016/j.gie.2020.06.040>.
- Khaira, U., Weni, I., & Wilia, W. (2024). Rancang Bangun Aplikasi Deteksi Penyakit Tanaman Jagung Melalui Citra Daun Berbasis Android Menggunakan Algoritma Convolutional Neural Network. *Jurnal Pepadun*, 5(1), 1-11. <https://doi.org/10.23960/pepadun.v5i1.210>.
- Khoiruddin, M., Junaidi, A., & Saputra, W. A. (2022). Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network. *Journal of Dinda: Data*

Science, Information Technology, and Data Analytics, 2(1), 37-45.  
<https://doi.org/10.20895/dinda.v2i1.341>.

Kurniawan, D. (2022). *Pengenalan machine learning dengan python*. Elex Media Komputindo.

NUR HIKMAH, N. U. R., & Ali Khumaidi, A. K. RANCANG BANGUN PROTOTIPE PENGUSIR HAMA BURUNG MENGGUNAKAN SENSOR GERAK RCWL MICROWAVE BERBASIS INTERNET OF THINGS. *Jurnal Simetris*. <https://doi.org/10.24176/simet.v11i2.5071>.

Pramesta, A. Perancangan Aplikasi Identifikasi Penyakit Kerdil dan Leaf *Blight* Pada Tanaman Lada Berdasarkan Citra Daun Berbasis Website (Bachelor's thesis, Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta). <https://doi.org/10.69916/jkbti.v1i3.10>.

Pranjaya, A. P., Rizki, F., Kurniawan, R., & Daulay, N. K. (2024). Klasifikasi Penyakit Pada Daun Tanaman Padi Berbasis *YoloV5* (You Only Look Once). *KLIK: Kajian Ilmiah Informatika dan Komputer*, 4(6), 3127-3136. <https://doi.org/10.30865/klik.v4i6.1916>.

Rijal, M., Yani, A. M., & Rahman, A. (2024). Deteksi Citra Daun untuk Klasifikasi Penyakit Padi menggunakan Pendekatan Deep Learning dengan Model CNN. *Jurnal Teknologi Terpadu*, 10(1), 56-62. <https://doi.org/10.54914/jtt.v10i1.1224>.

Saputra, R. A., Wasdiyanti, S., Supriyatna, A., & Saefudin, D. F. (2021). Penerapan Algoritma Convolutional Neural Network Dan Arsitektur *MobileNet* Pada Aplikasi Deteksi Penyakit Daun Padi. *Jurnal Swabumi*, 9(2), 184-188. <https://doi.org/10.31294/swabumi.v9i2.11678>.

Sayuthi, M., Hanan, A., Muklis, M., & Satriyo, P. (2020). Distribusi hama tanaman padi (*Oryza sativa* L.) pada fase vegetatif dan generatif di Provinsi Aceh. *Jurnal Agroecotania: Publikasi Nasional Ilmu Budidaya Pertanian*, 3(1), 1-10. <https://doi.org/10.22437/agroecotania.v3i1.11286>.

- Virgiawan, I., Maulana, F., Putra, M. A., Kurnia, D. D., & Sinduningrum, E. (2024). Deteksi dan tracking objek secara real time berbasis *computer vision* menggunakan metode *YOLO V3*. *Humantech: Jurnal Ilmiah Multidisiplin Indonesia*, 3(3). <https://doi.org/10.32670/ht.v3i3.4348>.
- Yuliany, S., & Rachman, A. N. (2022). Implementasi Deep Learning pada Sistem Klasifikasi Hama Tanaman Padi Menggunakan Metode Convolutional Neural Network (CNN). *Jurnal Buana Informatika*, 13(1), 54-65. <https://doi.org/10.24002/jbi.v13i1.5022>.
- Yusuf, M. N., & Hakim, D. L. (2020). Analisis Saluran Pemasaran Komoditas Padi (Studi Kasus di Desa Selasari Kecamatan Parigi Kabupaten Pangandaran). *Jurnal Ilmiah Mahasiswa AGROINFO GALUH*, 7(1), 97-111. <http://dx.doi.org/10.25157/jimag.v7i1.2563>.