

**IMPLEMENTASI *DEEP LEARNING* BERBASIS *YOLOv11*  
PADA KLASIFIKASI PENYAKIT TANAMAN PADI**

**TUGAS AKHIR**

**Diajukan Oleh:**

**MINDARINA  
210705115**

**Mahasiswa Fakultas Sains dan Teknologi  
Program Studi Teknologi Informasi**



**FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI AR-RANIRY  
BANDA ACEH  
2025 M / 1445 H**

## LEMBAR PERSETUJUAN

### IMPLEMENTASI *DEEP LEARNING* BERBASIS *YOLOv11* PADA KLASIFIKASI PENYAKIT TANAMAN PADI

#### TUGAS AKHIR

Diajukan Kepada Fakultas Sains dan Teknologi  
Universitas Islam Negeri (UIN) AR-Raniry Banda Aceh  
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1)  
Pada Program Studi Teknologi Informai

Oleh:

**MINDARINA**


2110705115

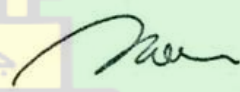
Mahasiswa Fakultas Sains dan Teknologi  
Program Studi Teknologi Informasi

Disetujui Untuk Dimunaqasyahkan Oleh:

Pembimbing I

Pembimbing II

  
**Khairan AR, M.Kom**  
NIP.198607042014031001

  
**Dr. Hendri Ahmadian, S.Si., M.I.M**  
NIP. 198301042014031002

Mengetahui,  
Ketua Program Studi Teknologi Informasi



  
**Malahayati, M.T**  
NIP.198301272015032003

## LEMBAR PENGESAHAN

### IMPLEMENTASI *DEEP LEARNING* BERBASIS *YOLOv11* PADA KLASIFIKASI PENYAKIT TANAMAN PADI

#### TUGAS AKHIR

Telah Diuji Panitia Ujian Munaqasyah Tugas Akhir  
Fakultas Sains dan Teknologi UIN AR-Raniry Banda Aceh dan Dinyatakan Lulus  
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S1)  
Dalam Program Studi Teknologi Informasi

Pada Hari/Tanggal: Rabu 20 Agustus 2025

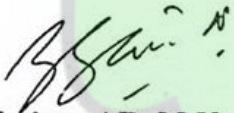
26 Safar 1447 H

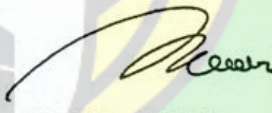
Di Darussalam, Banda Aceh

Panitia Ujian Munaqasyah Tugas Akhir:

Ketua,

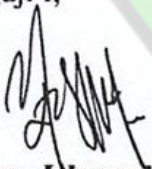
Sekretaris,

  
Khairan AR, M.Kom  
NIP.198607042014031001

  
Dr. Hendri Ahmadian, S.Si., M.I.M  
NIP. 198301042014031002

Penguji I,

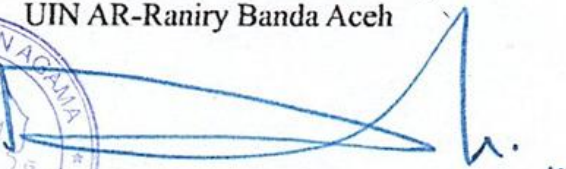
Penguji II,

  
Raihan Islamadina, M.T  
NIP. 198901312020122011

  
Malahayati, M.T  
NIP. 198301272015032003

Mengetahui,  
Dekan Fakultas Sains dan Teknologi  
UIN AR-Raniry Banda Aceh



  
Prof. Dr. Ir Muhammad Dirhamsyah, M.T., IPU  
NIP. 196210021988111001

## LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Mindarina  
NIM : 210705115  
Program Studi : Teknologi Informasi  
Fakultas : Sains dan Teknologi  
Judul : Implementasi Deep Learning berbasis YOLOv11 Pada Klasifikasi Penyakit Tanaman Padi

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggung jawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang di temukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap di kenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN AR-Raniry Banda Aceh, Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 20 Agustus 2025  
Yang Menyatakan



Mindarina

## ABSTRAK

Nama : Mindarina  
NIM : 210705115  
Program Studi : Teknologi Informasi  
Fakultas : Sains dan Teknologi  
Judul : Implementasi Deep Learning berbasis YOLOv11 Pada  
Klasifikasi Penyakit Tanaman Padi  
Tanggal Sidang : 20 Agustus 2025  
Jumlah Halaman : 78  
Pembimbing I : Khairan AR, M.Kom  
Pembimbing II : Dr. Hendri Ahmadian, S.Si., M.I.M

Penyakit pada tanaman padi merupakan salah satu faktor utama yang dapat menurunkan produktivitas pertanian secara signifikan. Deteksi dini dan akurat terhadap penyakit tersebut menjadi langkah penting dalam upaya pencegahan dan pengendalian. Pada penelitian ini, metode *Deep Learning* berbasis *YOLOv11* diterapkan untuk melakukan klasifikasi penyakit tanaman padi, mengingat kemampuannya dalam mendeteksi objek secara cepat dan akurat pada citra. Dataset yang digunakan terdiri dari citra daun padi yang dikategorikan menjadi daun sehat dan beberapa jenis penyakit.

Proses penelitian meliputi tahap pra-pemrosesan data, pelabelan, pelatihan model, dan evaluasi menggunakan metrik *precision*, *recall*, *mean Average Precision (mAP)*, dan *F1-Score*. Hasil testing menunjukkan bahwa model *YOLOv11* yang dibangun mampu mencapai dengan nilai *precision* 0,901, *recall* 0,974, *mAP50* sebesar 0,982, dan *F1-Score* 0,936 pada data testing. Temuan ini menunjukkan bahwa *YOLOv11* memiliki kinerja yang baik dalam mendeteksi dan mengklasifikasikan penyakit tanaman padi, sehingga berpotensi digunakan sebagai dasar pengembangan sistem deteksi otomatis yang dapat membantu petani dalam mengambil keputusan pengendalian penyakit secara lebih cepat dan tepat.

Kata Kunci: *YOLOv11*, *Deep Learning*, Penyakit Daun Padi, Klasifikasi.

## ***ABSTRACT***

*Name* : Mindarina  
*Student ID* : 210705115  
*Study Program* : Teknologi Informasi  
*Faculty* : Sains dan Teknologi  
*Title* : Implementasi Deep Learning berbasis YOLOv11 Pada  
Klasifikasi Penyakit Tanaman Padi  
*Defense Date* : 20 Agustus 2025  
*Number of Page* : 78  
*Supervisor I* : Khairan AR, M.Kom  
*Supervisor II* : Dr. Hendri Ahmadian, S.Si., M.I.M

*Diseases in Rice plants are one of the main factors that can significantly reduce agricultural productivity. Early and accurate Detection of these diseases is an important step in prevention and control efforts. In this study, a Deep Learning method based on YOLOv11 was implemented to classify Rice plant diseases, considering its ability to Detect objects quickly and accurately in images. The dataset used consists of Rice leaf images categorized into Healthy leaves and several disease classes. The research process includes data preprocessing, labeling, model Training, and evaluation using precision, recall, mean Average Precision (mAP), and F1-Score metrics. The experimental results show that the proposed YOLOv11 model achieved a precision of 0,901, recall of 0,974, mAP50 of 0,982, and an F1-Score of 0,936 on the test data. These findings indicate that YOLOv11 demonstrates strong performance in Detecting and classifying Rice plant diseases, making it a promising approach for the development of an automated Detection system to assist farmers in making faster and more accurate disease management decisions.*

*Keywords: YOLOv11, Deep Learning, Rice plant diseases, classification.*

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, taufik, dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “**Implementasi Deep Learning Berbasis YOLOv11 pada Klasifikasi Penyakit Tanaman Padi**” dengan baik. Shalawat dan salam semoga senantiasa tercurah kepada junjungan Nabi Muhammad SAW, beserta keluarga dan para sahabat beliau yang telah menuntun umat menuju jalan kebenaran.

Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry Darussalam Banda Aceh. Dalam proses penyusunannya, penulis mendapatkan dukungan, bantuan, serta doa dari berbagai pihak yang telah memberikan semangat, bimbingan, dan motivasi sehingga karya ini dapat terselesaikan.

Dengan penuh ketulusan dan rasa hormat, penulis ingin menyampaikan ucapan terima kasih kepada:

1. **Allah SWT**, atas segala rahmat, hidayah, dan pertolongan-Nya, sehingga penulis diberikan kesehatan, kekuatan, serta kemudahan dalam menyelesaikan setiap tahapan penelitian hingga skripsi ini dapat terselesaikan dengan baik.
2. **Kedua Orang Tua dan Keluarga Tercinta**, Penulis menyampaikan Rasa terima kasih yang tulus atas segala doa, dukungan, dan semangat yang tak pernah putus. Kehadiran mereka menjadi sumber kekuatan yang berarti dalam setiap langkah, serta memberikan motivasi yang besar bagi penulis untuk menyelesaikan Skripsi ini dengan sebaik-baiknya.
3. **Bapak Dr. Ir. M. Dirhamsyah, M.T., IPU**, selaku Dekan Fakultas Sains dan Teknologi, atas dukungan, fasilitas, dan lingkungan belajar yang mendukung selama penulis menempuh studi.
4. **Ibu Malahayati, M.T.**, Penulis juga menyampaikan rasa terima kasih yang tulus kepada Ibu selaku Ketua Program Studi Teknologi Informasi. Sikap

beliau yang begitu pengertian dan terbuka memberikan kenyamanan tersendiri bagi penulis dalam menjalankan Perkuliahan hingga akhir. Dukungan serta arahan dari beliau menjadi motivasi yang sangat berarti dan memberikan ruang bagi penulis untuk belajar dan berkembang.

5. **Bapak Khairan AR, M.Kom**, selaku Pembimbing I, yang telah memberikan bimbingan, arahan, masukan, dan motivasi secara sabar serta penuh dedikasi sejak awal penyusunan hingga tahap akhir skripsi ini.
6. **Bapak Dr. Hendri Ahmadian, S.Si., M.Sc**, selaku Pembimbing II, yang telah dengan tulus meluangkan waktu di tengah kesibukan untuk memberikan saran, kritik membangun, dan arahan yang sangat membantu dalam penyempurnaan penelitian ini.
7. **Ibu Cut Ida Rahmadiana, S.Si.**, selaku Admin Program Studi yang telah banyak membantu penulis dalam berbagai keperluan akademik sejak awal perkuliahan hingga saat ini. Beliau adalah sosok yang sangat peduli, ramah, dan selalu berusaha memudahkan urusan mahasiswa dengan penuh kesabaran. Kehangatan dan kebaikan, beliau memberikan kesan yang mendalam dan menciptakan suasana komunikasi yang menyenangkan di lingkungan program studi.
8. Seluruh Bapak dan Ibu Dosen Program Studi Teknologi Informasi, yang telah membekali penulis dengan ilmu pengetahuan, wawasan, dan pengalaman berharga sebagai bekal di masa depan.
9. Seseorang yang selalu hadir memberi dukungan dan bantuan di berbagai kesempatan. Meski namanya tidak tertulis, penulis tetap menyimpan rasa terima kasih atas kontribusi yang telah diberikan.
10. Cleo, kucing kesayangan yang bukan sekadar hewan peliharaan, melainkan sahabat dan keluarga yang selalu menghibur, menghilangkan stres, dan menemani di setiap momen proses penyusunan skripsi ini.
11. Adhe Mera Ulfa, sahabat seperjuangan sejak awal kuliah, yang selalu ada di segala kondisi, memberikan dukungan, tawa, dan semangat yang tak ternilai.

12. Seluruh teman-teman Prodi Teknologi Informasi angkatan 2021, atas kebersamaan, kerja sama, dan dukungan yang menjadi warna dalam perjalanan studi ini.
13. Diri sendiri, yang telah bertahan, berjuang, dan terus melangkah meskipun rintangan datang silih berganti. Perjalanan menuju tahap ini bukanlah hal yang mudah penuh dengan lelah, kebingungan, rasa ingin menyerah, dan tantangan yang tak terhitung jumlahnya. Namun, setiap langkah kecil yang diambil, setiap usaha yang dikerahkan, dan setiap air mata yang jatuh, semuanya mengantarkan penulis pada titik ini. Ada rasa bangga dan syukur yang tak bisa diungkapkan dengan kata-kata, karena akhirnya penulis bisa berdiri di tahap ini, melihat sejauh mana perjuangan telah membawa, dan menyadari bahwa semua itu layak dijalani.

Penulis menyadari bahwa skripsi ini masih memiliki kekurangan, baik dari segi isi maupun penyajiannya. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Penulis berharap karya ini dapat memberi manfaat bagi pembaca, peneliti, serta pihak yang bergerak di bidang pertanian dan teknologi, khususnya dalam penerapan *Deep Learning* untuk deteksi penyakit tanaman padi. Semoga penelitian ini menjadi langkah kecil yang memberi dampak positif bagi perkembangan ilmu pengetahuan dan inovasi di masa depan. Penulis memohon semoga Allah SWT senantiasa meridhai setiap usaha, menguatkan langkah, dan memberkahi setiap niat baik kita semua. Aamiin ya Rabbal 'alamin.

Banda Aceh, 13 Agustus 2025  
Penulis,

Mindarina

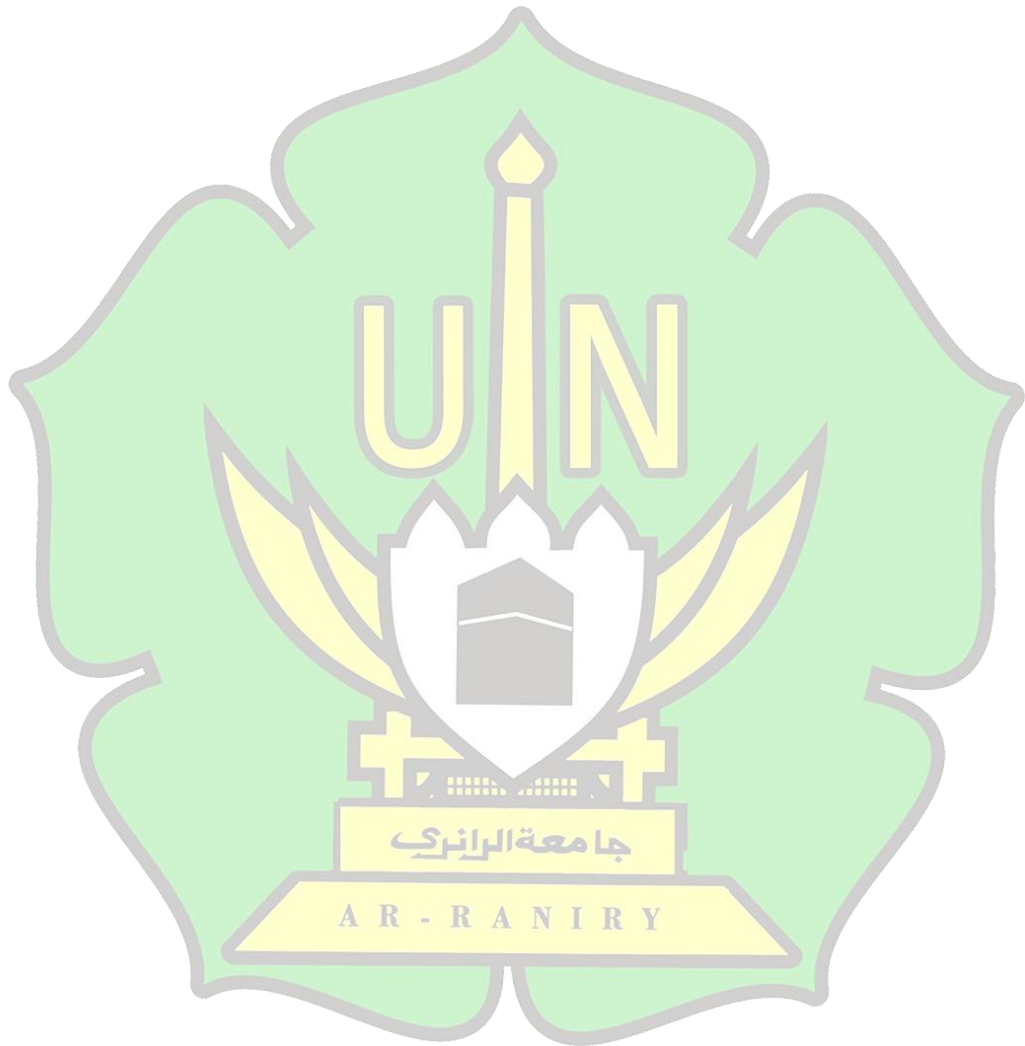
## DAFTAR ISI

ABSTRAK.....	IV
<i>ABSTRACT</i> .....	V
KATA PENGANTAR.....	VI
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian .....	2
1.4 Batasan Penelitian.....	2
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan .....	4
BAB II TINJAUAN PUSTAKA .....	5
2.1 Penelitian Terdahulu .....	5
2.2 Penyakit pada Tanaman Padi.....	8
2.3 <i>Deep Learning</i> .....	11
2.4 <i>Computer Vision</i> .....	12
2.5 Predictive Model.....	13
2.6 <i>YOLOv11</i> (You Look Only One version 11) .....	14
2.6.1 <i>YOLOv11</i> dalam Klasifikasi Penyakit pada Tanaman Padi .....	15
2.6.2 Perbandingan Performa <i>YOLOv11</i> dengan Generasi Sebelumnya.....	18
2.7 Roboflow.....	18
2.8 <i>Google Colaboratory</i> .....	19
2.9 <i>Python</i> .....	20
2.10 <i>Confusion matrix</i> .....	21
2.11 <i>mAP</i> (mean Average Precision) .....	22
2.12 Kerangka Berpikir .....	23
BAB III METODE PENELITIAN.....	24
3.1 Tahapan Penelitian.....	24
3.2 Pengumpulan Data.....	25
3.3 Labeling dan Pembuatan Bounding Box.....	25
3.4 Preprocessing data .....	26
3.5 Split dataset (pembagian dataset) .....	29
3.5 Implementasi <i>YOLOv11</i> .....	30

3.6	Training, Validation dan Testing Model .....	32
3.6.1	Proses <i>Training</i> dan <i>Validation</i> .....	32
3.6.2	Proses Testing Model .....	33
3.7	Evaluasi Kinerja Model .....	34
3.8	Analisis Hasil .....	35
3.9	Waktu dan Lokasi Penelitian .....	35
3.10	Alat dan Bahan .....	35
3.11	Perangkat Keras .....	36
3.12	Perangkat Lunak .....	36
BAB IV HASIL DAN PEMBAHASAN .....		37
4.1	Hasil Evaluasi Dataset <i>Training</i> dan <i>Validation</i> .....	37
4.6.1	Identifikasi Peningkatan Akurasi pada Beberapa Epoch .....	37
4.6.2	Hasil <i>Loss</i> dan Evaluasi Model .....	38
4.6.3	Visualisasi <i>Train Batch</i> .....	41
4.6.4	Visualisasi Prediksi dari <i>Validation Batch</i> .....	42
4.6.5	Kurva dan Metrik Evaluasi Dataset Training dan Validation .....	47
4.6.6	Visualisasi <i>Confusion matrix</i> pada Data Dataset Training dan Validation ....	50
4.7	Hasil Evaluasi Dataset <i>Testing</i> .....	52
4.7.1	Visualisasi Prediksi Dataset <i>Testing</i> .....	52
4.7.2	Kurva dan Metrix Evaluasi .....	55
4.7.3	Visualisasi <i>Confusion matrix</i> Dataset Testing .....	58
BAB V PENUTUP .....		60
5.1	Kesimpulan .....	60
5.2	Saran .....	60
DAFTAR PUSTAKA .....		62

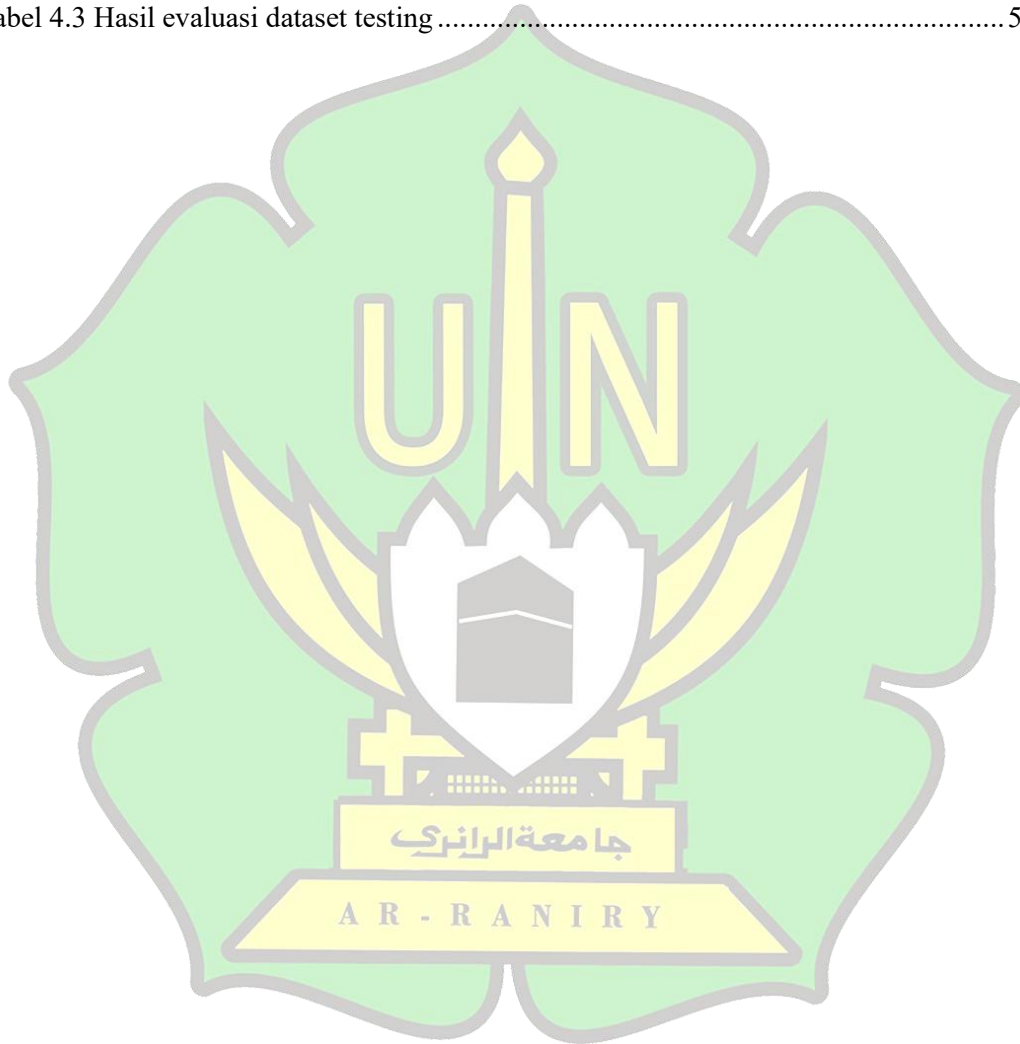
## DAFTAR GAMBAR

Gambar 2.1 Bacterial Leaf Blight .....	9
Gambar 2.2 Brown spot .....	10
Gambar 2.3 Leaf blast .....	10
Gambar 2.4 Healty Rice Leaf.....	11
Gambar 2.5 Arsitektur YOLOv11 .....	17
Gambar 2.6 Perbandingan versi YOLO.....	18
Gambar 2.7 Kerangka Berpikir .....	23
Gambar 3.1 Tahapan Penelitian.....	24
Gambar 3.2 Sumber pengumpulan dataset .....	25
Gambar 3.3 Proses Bounding Box .....	26
Gambar 3.4 Proses Resize Gambar .....	27
Gambar 3.5 Proses Augmentasi Dataset.....	28
Gambar 3.6 Proses Split Dataset di Google Colab .....	30
Gambar 3.7 Source code Untuk Train dan Validation Model .....	31
Gambar 3.8 Source code Untuk Melanjuttkkan Proses Training dan validation.....	32
Gambar 3.9 Tampilan Hasil Train dan Validation Model di Terminal .....	33
Gambar 3.10 Source code untuk proses Testing.....	34
Gambar 4.1 Gambar Grafik Loss, Precission, Recall, dan mAP .....	39
Gambar 4.2 Visualisasi Train Batch .....	42
Gambar 4.3 Hasil Prediksi Kelas Brown spot .....	43
Gambar 4.4 Hasil Prediksi Kelas Healty Rice.....	44
Gambar 4.5 Hasil Prediksi Kelas Bacterial Leaf Blight .....	45
Gambar 4.6 Hasil Prediksi Kelas Leaf blast.....	46
Gambar 4.7 Grafik Recall Confidence .....	47
Gambar 4.8 Grafik Precision Recall.....	48
Gambar 4.9 Grafik Precison-Confidence .....	49
Gambar 4.10 Tampilan Confution Matrix .....	51
Gambar 4.11 Hasil prediksi pada kelas Brown spot .....	53
Gambar 4.12 Hasil prediksi pada kelas Healthy Rice.....	54
Gambar 4.13 Hasil prediksi pada kelas Bacterial Leaf Blight .....	54
Gambar 4.14 Hasil prediksi pada kelas Leaf blast.....	55
Gambar 4.15 Grafik Recall Confidence Curve.....	56
Gambar 4.16 Grafik Precision Recall Curve .....	57
Gambar 4.17 Grafik Precision Confidence Curve .....	58



## DAFTAR TABEL

Tabel 2.1 State of Art Penelitian.....	6
Tabel 3.1 Pembagian Dataset .....	29
Tabel 3.2 Spesifikasi Perangkat Keras .....	36
Tabel 3.3 Spesifikasi Perangkat Lunak.....	36
Tabel 4.1 Tabel Hasil Evaluasi Training dan Validation.....	37
Tabel 4.2 Peningkatan Akurasi di Beberapa Epoch.....	38
Tabel 4.3 Hasil evaluasi dataset testing.....	52



# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Padi (*Oryza sativa*) merupakan komoditas pangan utama di banyak negara Asia dan sangat berperan pada ketahanan pangan. Produktivitas padi rentan terhadap serangan berbagai penyakit yang disebabkan jamur, bakteri, maupun virus; penyakit seperti *leaf blast*, *Brown spot*, dan *tungro* dapat menyebabkan penurunan hasil yang signifikan jika tidak dideteksi dan ditangani sejak dini. Karena itu, deteksi dini penyakit padi sangat penting untuk meminimalkan kerugian produksi dan meningkatkan efisiensi pengendalian hama/penyakit (Y. Li et al., 2024). Perkembangan *Deep Learning* (khususnya model-model CNN dan *ensemble*) telah terbukti efektif untuk diagnosis otomatis penyakit padi dari citra daun. Salah satu studi besar mengembangkan model *ensemble* (menggabungkan *DenseNet*, *SE-ResNet*, dsb.) pada dataset >33.000 gambar dan melaporkan performa tinggi untuk beberapa jenis penyakit padi, sekaligus mengimplementasikannya dalam aplikasi untuk mendukung diagnosis lapangan. Temuan-temuan seperti ini menunjukkan potensi kuat *Deep Learning* untuk deteksi penyakit padi secara otomatis (Deng et al., 2021).

Di ranah deteksi objek *real-time*, keluarga model YOLO banyak digunakan karena kemampuan deteksi dan klasifikasi sekaligus dalam satu langkah (*one-stage*), dengan kecepatan inferensi yang cocok untuk aplikasi lapangan atau perangkat edge. Penelitian terbaru telah mengadaptasi varian YOLO (mis. YOLOv5 dan modifikasi-nya) untuk mendeteksi penyakit padi di kondisi lapangan, beberapa studi melaporkan peningkatan *mAP*, akurasi, dan ukuran model yang tetap kecil sehingga cocok untuk *deployment*. Contoh penerapan YOLO pada kasus penyakit padi dan optimasinya untuk kondisi lapangan tersedia di literatur terkini (P. Li et al., 2025). Seiring evolusi keluarga YOLO, muncul juga penelitian yang mengembangkan *YOLOv11* dan turunannya khusus untuk deteksi penyakit daun; salah satu studi RLDD-*YOLOv11n* (2025) mengadaptasi arsitektur *YOLOv11* untuk tugas deteksi penyakit daun padi dan melaporkan performa kuat pada dataset yang diuji. Karena itu, memilih arsitektur *YOLOv11* untuk studi ini relevan: selain menargetkan akurasi klasifikasi, penelitian ini juga perlu mengevaluasi kinerja

(*precision, recall, mAP*) dan aspek *real-time/inference speed* yang penting untuk aplikasi lapangan (Fang et al., 2025).

Berdasarkan uraian di atas, penelitian ini difokuskan pada implementasi arsitektur *YOLOv11* untuk klasifikasi penyakit pada tanaman padi, pengukuran kinerja model menggunakan metrik standar (akurasi, *precision, recall, mAP*), serta pelaporan tingkat akurasi yang dicapai, sehingga hasil penelitian ini dapat menjadi dasar pengembangan alat deteksi penyakit padi yang dapat digunakan secara praktis di lapangan.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana cara menerapkan arsitektur *YOLOv11* untuk klasifikasi pada penyakit tanaman padi?
2. Bagaimana cara mengukur kinerja model dalam klasifikasi pada penyakit tanaman padi menggunakan *YOLOv11*?
3. Bagaimana tingkat akurasi klasifikasi penyakit pada tanaman padi menggunakan arsitektur *YOLOv11*?

## 1.3 Tujuan Penelitian

1. Menerapkan arsitektur *YOLOv11* untuk klasifikasi penyakit pada tanaman padi.
2. Mengevaluasi kinerja model dalam klasifikasi penyakit pada tanaman padi menggunakan *YOLOv11*.
3. Mengetahui tingkat akurasi klasifikasi penyakit pada tanaman padi menggunakan arsitektur *YOLOv11*.

## 1.4 Batasan Penelitian

1. Klasifikasi penyakit pada penelitian ini hanya mencakup empat kelas yaitu:
  - *Brown spot (Cochliobolus miyabeanus)* – Bercak Coklat
  - *Healthy Rice Leaf* – Daun Padi Sehat
  - *Bacterial Leaf Blight (Xanthomonas oryzae pv. oryzae)* – Hawar Daun Bakteri
  - *Leaf blast (Pyricularia oryzae)* – Blas Daun

2. Menggunakan dataset citra daun tanaman padi yang tersedia secara publik, dengan dominasi gambar original. Proses augmentasi dilakukan secara otomatis melalui *Roboflow* setelah tahap *labeling* selesai, dan hasil anotasi telah disesuaikan dengan format pelatihan YOLO.
3. Penelitian ini menggunakan arsitektur *YOLOv11* untuk klasifikasi penyakit pada citra daun tanaman padi berdasarkan fitur visual yang teridentifikasi oleh model.
4. Evaluasi performa model dilakukan dengan menggunakan metrik *Precision*, *Recall*, mean Average *Precision* pada IoU 0.5 (*mAP50*) dan *F1-Score*. Selain itu, metrik *Bounding box Loss*, *Objectness Loss*, dan *Classification Loss* juga dianalisis selama proses pelatihan untuk mengukur kemampuan model dalam mendeteksi dan mengklasifikasikan penyakit pada daun padi secara akurat.

#### 1.5 Manfaat Penelitian

1. Penelitian ini memberikan wawasan mengenai penerapan serta implementasi metode *Deep Learning*, khususnya arsitektur *YOLOv11*, dalam klasifikasi penyakit pada tanaman padi melalui citra digital. Hal ini berkontribusi dalam memperkenalkan teknologi kecerdasan buatan (AI) ke dalam bidang pertanian modern.
2. Hasil penelitian ini menyediakan dasar teknis yang efisien dan akurat dalam mendeteksi serta mengklasifikasikan penyakit padi secara otomatis menggunakan pendekatan *Deep Learning* berbasis *YOLOv11*. Meskipun model belum diimplementasikan ke dalam bentuk aplikasi, hasil ini dapat menjadi acuan awal bagi pengembangan sistem bantu diagnosis penyakit padi yang praktis bagi petani, penyuluh pertanian, dan pihak terkait lainnya di masa mendatang.
3. Penelitian ini dapat menjadi referensi bagi pengembang dan peneliti dalam mengintegrasikan model *Deep Learning* berbasis deteksi objek ke dalam sistem lain yang membutuhkan analisis berbasis citra, khususnya dalam bidang pertanian presisi dan bioteknologi.

4. Hasil penelitian ini diharapkan dapat menjadi dasar bagi penelitian lanjutan atau pengembangan sistem berbasis AI untuk memantau kesehatan tanaman dan mendukung keputusan pertanian yang lebih cepat dan akurat.

## 1.6 Sistematika Penulisan

Untuk dapat mengetahui isi penelitian ini, Sistematika penulisan skripsi ini disusun dengan urutan sebagai berikut:

1. BAB I PENDAHULUAN

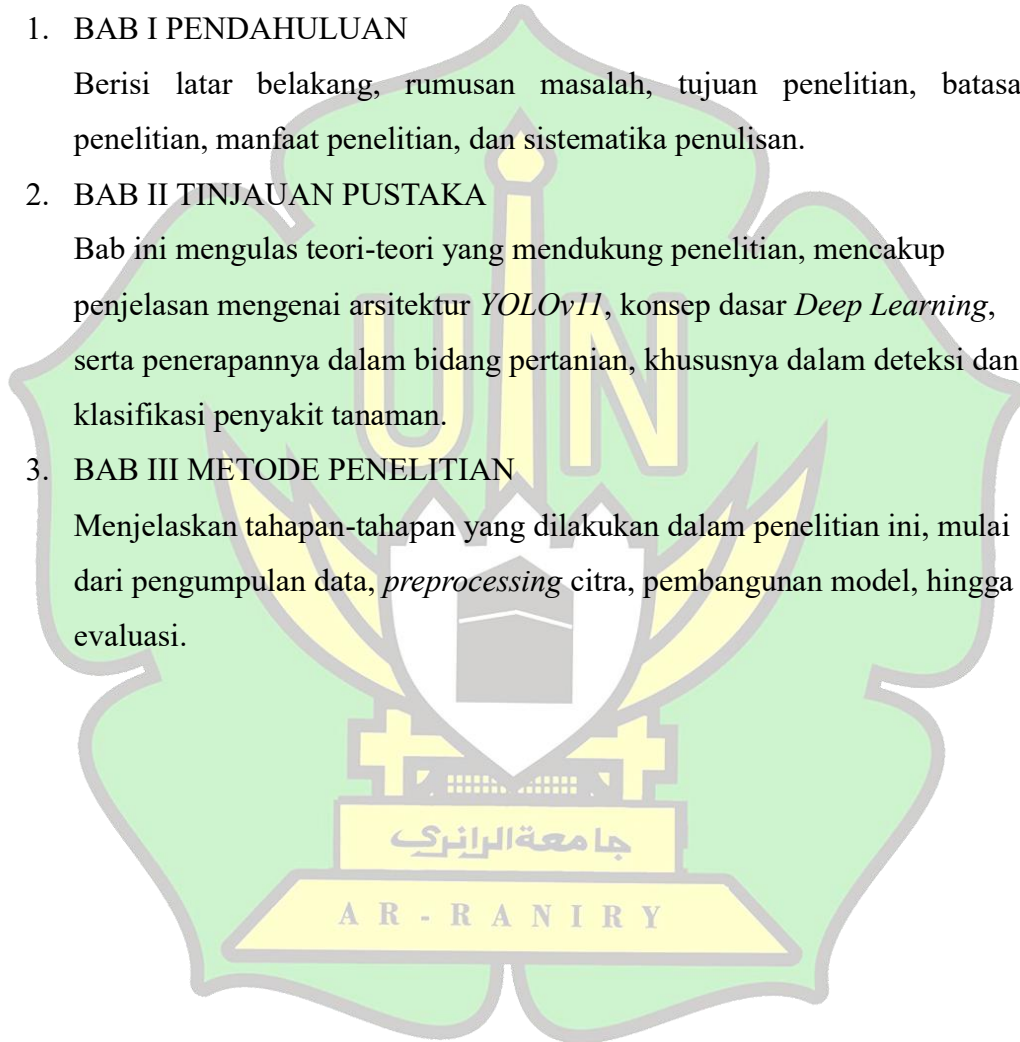
Berisi latar belakang, rumusan masalah, tujuan penelitian, batasan penelitian, manfaat penelitian, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Bab ini mengulas teori-teori yang mendukung penelitian, mencakup penjelasan mengenai arsitektur *YOLOv11*, konsep dasar *Deep Learning*, serta penerapannya dalam bidang pertanian, khususnya dalam deteksi dan klasifikasi penyakit tanaman.

3. BAB III METODE PENELITIAN

Menjelaskan tahapan-tahapan yang dilakukan dalam penelitian ini, mulai dari pengumpulan data, *preprocessing* citra, pembangunan model, hingga evaluasi.



## BAB II TINJAUAN PUSTAKA

### 2.1 Penelitian Terdahulu

Penelitian mengenai deteksi penyakit tanaman padi berbasis *Deep Learning* telah menjadi fokus penting dalam pengembangan sistem pertanian cerdas. Pendekatan visual berbasis analisis warna daun, tekstur, dan pola bercak pada citra tanaman telah terbukti efektif dalam identifikasi otomatis jenis penyakit. Salah satu metode yang naik daun adalah YOLO (*You Only Look Once*), karena kemampuannya melakukan deteksi dan klasifikasi secara *real-time* dengan akurasi tinggi. Sebagai contoh pada penelitian (Krisdianto et al., 2024) menggunakan YOLO untuk mendeteksi empat kondisi daun padi (*leaf smut*, *Bacterial Leaf Blight*, *Brown spot*, dan sehat) dengan hasil akurasi rata-rata sebesar 77% pada pelatihan dan testing, menunjukkan potensi nyata metode tersebut untuk implementasi praktis di lapangan.

Dengan keberhasilan penerapan YOLO versi sebelumnya ini, penggunaan arsitektur *YOLOv11* diharapkan menawarkan peningkatan performa dan kecepatan prediksi tetap efisien namun lebih presisi, sehingga sangat layak untuk diaplikasikan dalam monitoring penyakit tanaman padi, terutama di perangkat dengan sumber daya terbatas. Oleh karena itu, penerapan *YOLOv11* dalam klasifikasi penyakit tanaman padi berpotensi mendukung peningkatan ketahanan pangan dan produktivitas melalui deteksi yang lebih cepat dan akurat.

Penelitian pertama yaitu deteksi rambu lalu lintas *Real-time* di Indonesia dengan penerapan *YOLOv11* solusi untuk keamanan berkendara telah dilakukan oleh (Pradana et al., 2024). Metode yang digunakan adalah *YOLOv11*, sebuah algoritma *Deep Learning* yang dirancang untuk deteksi objek dengan kecepatan tinggi dan akurasi tinggi. dengan nilai *mAP50* yang stabil di sekitar 0,99 setelah beberapa *epoch* pelatihan.

Penelitian kedua yaitu Identifikasi Mangga Berdasarkan Tingkat Kematangan Menggunakan Yolo 11 oleh (Huda et al., 2025) dimana Penelitian ini membuktikan bahwa metode yang digunakan mampu mendeteksi kematangan buah dengan akurasi tinggi. Hasil evaluasi seperti *mAP* menunjukkan performa model

yang stabil, sementara nilai *Precision* dan *Recall* menandakan ketepatan klasifikasi serta minimnya kesalahan dalam pendeteksian.

Penelitian ketiga oleh (Junervin, 2025) tentang Deteksi dan klasifikasi otomatis lapisan grafena berbasis *YOLOv11*. dimana Penelitian ini menggunakan *YOLOv11* yang seimbang antara akurasi dan kecepatan inferensi. Model ini mampu mengenali beberapa kelas dengan baik dan menunjukkan performa deteksi objek yang efektif. Waktu inferensi yang cepat (40.4ms per gambar) menjadikannya cocok untuk aplikasi *real-time*.

Penelitian ke empat yaitu Deteksi dan Klasifikasi Sel Darah Otomatis dalam Gambar Mikroskopis Menggunakan *YOLOv11* dan Bobot yang Dioptimalkan Oleh (Sazak & Kotan, 2024) Studi ini menyelidiki arsitektur *YOLOv10* dan *YOLOv11* yang baru dikembangkan untuk deteksi dan klasifikasi sel darah. diamati bahwa *YOLOv11* menunjukkan tingkat keberhasilan yang tinggi sebesar 93,8%. *YOLOv11* memberikan hasil yang lebih baik. terutama dalam mendeteksi sel darah putih dan trombosit, dan mencapai tingkat akurasi yang lebih tinggi daripada *YOLOv10*.

Penelitian kelima yaitu Model baru yang Efisien untuk mendeteksi penyakit daun jagung berdasarkan peningkatan Model *YOLOv11* Oleh (He et al., 2025) dimana penelitian ini mengusulkan pengembangan arsitektur *YOLOv11* yang telah ditingkatkan untuk menghasilkan performa deteksi yang lebih baik. Hasilnya menunjukkan bahwa model ini memiliki akurasi yang unggul dibandingkan dengan model lainnya, dengan nilai *precision* mencapai 92,6%, *recall* sebesar 85,4%, dan skor F1 sebesar 88,9%.

Tabel 2.1 State of Art Penelitian

No	REFERENSI	METODE	DESKRIPSI SINGKAT	HASIL
1	(Pradana et al., 2024)	Deteksi Rambu Lalu Lintas <i>Real-time</i> di Indonesia dengan Penerapan <i>YOLOv11</i> : Solusi untuk Keamanan Berkendara	Model <i>YOLOv11</i> digunakan untuk deteksi rambu lalu lintas di Indonesia dan menunjukkan akurasi tinggi dengan <i>mAP</i> 0,99, cocok	Model mencapai akurasi yang tinggi, dengan nilai <i>mAP50</i> yang stabil di

			untuk sistem transportasi cerdas.	sekitar 0,99 setelah beberapa <i>epoch</i> pelatihan
2	(Huda et al., 2025)	Identifikasi Mangga Berdasarkan tingkat kematangan menggunakan <i>YOLOv11</i>	Penelitian ini bertujuan untuk mendeteksi tingkat kematangan buah mangga menggunakan algoritma <i>YOLOv11</i> , guna mendukung efisiensi dalam pengelolaan hasil panen di sektor pertanian.	Penelitian ini menunjukkan bahwa metode tersebut mampu mendeteksi tingkat kematangan buah secara akurat dan efektif.
3	(Junervin, 2025)	Deteksi dan Klasifikasi Otomatis Lapisan Grafena Berbasis <i>YOLOv11</i>	Penelitian ini bertujuan untuk mengembangkan model deteksi berbasis <i>YOLOv11</i> yang mampu mendeteksi dan mengklasifikasi grafena dalam berbagai ketebalan lapisan.	Penelitian ini membuktikan bahwa model <i>YOLOv11</i> sangat baik dalam mendeteksi objek grafena dengan berbagai layer. Dan kecepatan yang efisien yaitu 40.4ms per gambar yang dapat digunakan

				pada aplikasi <i>real-time</i>
4	(Sazak & Kotan, 2024)	Deteksi dan Klasifikasi Sel Darah Otomatis dalam Gambar Mikroskopis Menggunakan <i>YOLOv11</i> dan Bobot yang Dioptimalkan	Studi ini mengembangkan dan mengevaluasi model <i>YOLOv10</i> dan <i>YOLOv11</i> untuk mendeteksi serta mengklasifikasikan sel darah merah, sel darah putih, dan trombosit pada citra mikroskopis sebagai bagian dari proses awal pemeriksaan darah lengkap (CBC).	<i>YOLOv11</i> menunjukkan hasil yang sangat baik dalam mendeteksi dengan tingkat keberhasilan yang tinggi yaitu 93,8%.
5	(He et al., 2025))	<i>YOLOv11</i> -RCDWD: Model Efisien Baru untuk Mendeteksi Penyakit Daun Jagung Berdasarkan Peningkatan Model <i>YOLOv11</i>	penelitian ini memperkenalkan metode deteksi ringan berdasarkan model <i>YOLOv11</i> yang telah disempurnakan untuk Mendeteksi hama dan penyakit pada daun jagung	Model <i>YOLOv11s</i> -RCDWD menunjukkan performa yang terbaik dengan presisi 92,6%, <i>recall</i> 85,4%, dan <i>F1-Score</i> 88,9%, lebih unggul dari model lainnya.

## 2.2 Penyakit pada Tanaman Padi

Tanaman padi (*Oryza sativa*) adalah komoditas pangan strategis dunia, terutama di Asia. Namun, tanaman ini rentan terserang berbagai penyakit seperti

bakterial agar daun (*Bacterial Leaf Blight*, BLB), bercak coklat (*Brown spot*), dan blas daun (*Leaf blast*) yang mampu menurunkan hasil panen dan kualitas biji secara signifikan. Deteksi dini penyakit padi menjadi sangat penting untuk menjaga produktivitas pertanian dan mengurangi kerugian ekonomi. Beberapa penyakit utama yang sering menyerang padi antara lain:

a. *Bacterial Leaf Blight (anthomonas oryzae pv. oryzae)*

Penyakit BLB merupakan salah satu penyakit penting pada padi dan dianggap sangat merugikan secara ekonomi. Infeksi oleh *Xanthomonas oryzae pv. oryzae* mengakibatkan penurunan hasil secara dramatis. Untuk mengatasi ini, pengembangan varietas tahan melalui teknik modern sangat penting, termasuk pendekatan genomik dan strategi perlawanan tanaman (Sanya et al., 2022).



Gambar 2.1 Bacterial Leaf Blight

b. *Brown spot (Cochliobolus miyabeanus)*

*Brown spot* atau Penyakit bercak coklat adalah penyakit yang disebabkan oleh cendawan *Cochliobolus miyabeanus*, yang menyerang daun padi terutama pada kondisi kelembapan tinggi dan tanah miskin nutrisi. Penyakit ini muncul sebagai bercak oval berwarna coklat kehitaman yang dapat berkembang dan menurunkan kesehatan tanaman (Leiwakabessy et al., 2020).



Gambar 2.2 Brown spot

c. *Leaf blast (Pyricularia oryzae)*

*Leaf blast* atau penyakit blas daun merupakan salah satu ancaman utama bagi produksi padi. Jamur *P. oryzae* dapat menyebabkan lesi khas yang mengganggu fotosintesis dan sangat cepat menyebar dalam kondisi lembap. Studi terbaru menunjukkan bahwa penyakit ini bisa menyebabkan penurunan hasil hingga 50% pada beberapa varietas (Astriawati & Anfa, 2025).



Gambar 2.3 Leaf blast

Selain ke tiga kelas penyakit yang telah dijelaskan sebelumnya, penelitian ini juga menyertakan satu kelas tambahan yaitu kelas sehat, yang merepresentasikan daun tanaman padi tanpa gejala penyakit. Kelas ini digunakan sebagai pembandingan agar model dapat membedakan antara daun yang terinfeksi dan daun yang dalam kondisi normal. Dengan menyertakan kelas sehat, proses klasifikasi menjadi lebih akurat karena model dilatih untuk mengenali kondisi tanaman dalam keadaan tidak terpapar penyakit apapun.



Gambar 2.4 Healty Rice Leaf

### 2.3 *Deep Learning*

*Deep Learning*, khususnya dengan arsitektur *Convolutional Neural Network (CNN)*, telah menjadi salah satu teknologi inti dalam analisis citra digital karena kemampuannya mempelajari representasi fitur secara otomatis dari data mentah. Proses pembelajaran ini dilakukan secara bertingkat, mulai dari fitur sederhana seperti garis, sudut, dan tekstur, hingga pola kompleks yang merepresentasikan objek atau gejala penyakit tertentu pada tanaman. Keunggulan *CNN* terletak pada kemampuannya mempertahankan informasi spasial pada citra, sehingga sangat efektif untuk tugas klasifikasi dan deteksi objek. Karakteristik ini membuat *CNN* menjadi fondasi berbagai arsitektur modern seperti YOLO (*You Only Look Once*), yang menggabungkan proses deteksi dan klasifikasi secara *real-time* dalam satu kerangka kerja *end-to-end*.

Dalam bidang pertanian, penerapan *CNN* telah menunjukkan performa yang konsisten dan akurat. Penelitian yang dilakukan oleh (Novantara et al., 2025) berhasil mengembangkan aplikasi berbasis *CNN* untuk mendeteksi penyakit daun padi, meliputi *Brown spot*, *blast*, *hispa*, dan daun sehat, dengan menggunakan dataset 480 gambar. Hasilnya menunjukkan akurasi hingga 97,75 %, dengan *F1-Score* 0,97 untuk *Brown spot* dan 0,952 untuk daun sehat, membuktikan efektivitas *CNN* dalam mengidentifikasi gejala penyakit meskipun dengan variasi bentuk daun, pencahayaan, dan latar belakang yang berbeda.

Penelitian lain oleh (Maulana et al., 2025) memanfaatkan arsitektur *CNN ResNet50V2* melalui pendekatan *transfer learning* untuk mengklasifikasikan penyakit daun padi. Model ini dilatih menggunakan dataset citra beranotasi dan mencapai akurasi testing sebesar 94,14 %, yang menunjukkan bahwa pemanfaatan model *CNN pre-Trained* dapat menghemat waktu pelatihan sekaligus meningkatkan akurasi deteksi. Hasil ini juga menegaskan bahwa *CNN* dapat dioptimalkan untuk digunakan dalam kondisi lapangan dengan sumber daya komputasi terbatas tanpa mengorbankan kualitas prediksi.

Dengan bukti empiris dari kedua studi tersebut, *CNN* dipilih sebagai fondasi dalam penelitian ini sebelum diintegrasikan ke dalam arsitektur *YOLOv11*. Pendekatan ini diharapkan mampu memadukan keunggulan akurasi *CNN* dengan kecepatan deteksi *real-time YOLOv11*, sehingga sistem dapat melakukan identifikasi penyakit daun padi secara cepat, akurat, dan efisien, yang pada akhirnya berkontribusi pada peningkatan produktivitas dan ketahanan pangan.

#### **2.4 Computer Vision**

*Computer Vision* adalah cabang dari kecerdasan buatan yang bertujuan agar komputer dapat memperoleh, memproses, dan memahami informasi dari gambar digital. Dalam bidang pertanian, teknologi ini memiliki peran yang sangat penting, terutama dalam membantu proses identifikasi kondisi tanaman secara otomatis berdasarkan gejala visual pada daun maupun bagian tanaman lainnya.

Proses identifikasi citra umumnya meliputi beberapa tahap, yaitu akuisisi gambar, ekstraksi fitur, segmentasi objek, dan klasifikasi. Tahapan ini memungkinkan sistem mengenali pola visual seperti bentuk bercak, warna daun, atau distribusi lesi yang menjadi indikator adanya penyakit, misalnya bercak coklat, blas daun, atau hawar pelepah.

Penggunaan *YOLOv11 (You Only Look Once* versi 11) semakin memperkuat kemampuan deteksi otomatis tersebut. Model ini memungkinkan deteksi objek dan klasifikasi dilakukan dalam satu proses tunggal, sehingga lebih efisien dibandingkan metode tradisional yang memisahkan kedua tahap tersebut. *YOLOv11* secara simultan menentukan posisi objek (melalui *bounding box*) dan mengklasifikasikannya ke kelas penyakit yang sesuai. Pendekatan ini sangat cocok

digunakan dalam penelitian yang membutuhkan deteksi dan klasifikasi secara bersamaan.

Salah satu penelitian di Indonesia yang mengimplementasikan YOLO untuk deteksi penyakit padi menunjukkan bahwa model ini mampu memberikan solusi yang cepat, akurat, dan praktis bagi petani. Krisdianto et al. (2024) menggunakan YOLO untuk mendeteksi empat kondisi daun padi, yaitu *Bacterial Leaf Blight*, *leaf smut*, *Brown spot*, dan kondisi sehat, dengan akurasi pelatihan dan testing sebesar 77%. Hasil tersebut membuktikan bahwa pendekatan YOLO efektif untuk digunakan di lingkungan pertanian modern dan dapat mendukung pengambilan keputusan yang lebih cepat (Krisdianto et al., 2024).

Dengan dukungan teknologi *Computer Vision* dan *YOLOv11*, analisis penyakit pada tanaman padi tidak lagi bergantung pada pemeriksaan manual yang memerlukan waktu lama. Sebaliknya, pendekatan ini memungkinkan deteksi dan klasifikasi penyakit dilakukan secara otomatis dalam skala luas, sehingga mendukung efisiensi dan akurasi dalam praktik pertanian modern.

## 2.5 Predictive Model

Predictive model adalah metode untuk memperkirakan kejadian, nilai, atau kondisi di masa depan dengan mengandalkan pola dan informasi dari data historis. Dalam *machine learning*, model prediktif umumnya dikembangkan menggunakan pendekatan *supervised learning*, yaitu metode di mana model dilatih berdasarkan data yang telah diberi label. Ini memungkinkan model untuk memahami hubungan yang kompleks antara variabel input (*features*) dan output (*target*). Keberhasilan prediksi sangat tergantung pada kualitas dan representativitas data latih, karena data yang baik akan menghasilkan model yang akurat dan andal. Oleh sebab itu, bagian seperti pengumpulan data, pra-pemrosesan, dan teknik pelatihan yang tepat sangat penting untuk mencapai hasil prediksi yang optimal.

Di sektor pertanian Indonesia, penerapan model prediktif telah menunjukkan hasil yang menjanjikan. Salah satu penelitian membandingkan metode *Random Forest* dan *Gradient Boosting Regressor* dalam meramalkan harga bawang merah di kota Samarinda. Hasilnya menunjukkan bahwa model *Random Forest* memberikan nilai *Root Mean Square Error* (RMSE) dan *MAPE* yang lebih rendah,

sehingga diidentifikasi sebagai metode paling akurat untuk komoditas tersebut (Fitri & Nugraha, 2024). Selain itu, penelitian lain yang menggunakan arsitektur LSTM dan LSTM dengan *Attention Mechanism* untuk meramalkan harga cabai rawit merah di Provinsi Jawa Barat menunjukkan performa yang luar biasa. Model terbaik berhasil mencapai *Mean Absolute Percentage Error (MAPE)* sebesar hanya 2,74%, membuktikan bahwa teknik *Deep Learning* berbasis deret waktu dapat mengatasi fluktuasi harga komoditas secara efektif (Witanti et al., 2024).

## 2.6 YOLOv11 (You Look Only One version 11)

*YOLOv11* merupakan arsitektur deteksi objek mutakhir yang memanfaatkan pendekatan *Convolutional Neural Network (CNN)* secara *end-to-end*, sehingga mampu melakukan proses deteksi dan klasifikasi secara *real-time* dengan tingkat akurasi yang tinggi. Pada penelitian ini, *YOLOv11* digunakan untuk mendeteksi sekaligus mengklasifikasikan empat kondisi daun padi, yang mencakup daun sehat dan daun yang terinfeksi penyakit. Proses kerja model dimulai dengan membagi citra menjadi beberapa grid, kemudian secara bersamaan memprediksi *bounding box* dan label kelas. Kemampuan ini membuat *YOLOv11* sangat efisien, terutama untuk digunakan pada perangkat dengan keterbatasan sumber daya komputasi.

Keunggulan *YOLOv11* dibandingkan versi sebelumnya, seperti *YOLOv5* hingga *YOLOv8*, terletak pada peningkatan kemampuannya dalam mendeteksi objek berukuran kecil, termasuk bercak gelap akibat penyakit pada daun padi. Model ini juga menunjukkan stabilitas yang baik selama proses pelatihan meskipun data yang digunakan tidak seimbang (*imbalanced dataset*), serta dapat berjalan optimal di berbagai *platform* pengembangan seperti *Google Colab* dan *Roboflow*. Hasil keluaran model berupa *bounding box* yang mengelilingi area daun, disertai dengan skor kepercayaan (*confidence score*) dan label klasifikasi. Untuk menghindari prediksi ganda, digunakan teknik *Non-Maximum Suppression (NMS)*. Fitur-fitur tersebut menjadikan *YOLOv11* relevan untuk aplikasi pertanian presisi karena dapat memberikan visualisasi hasil deteksi yang akurat dan mudah diinterpretasikan.

Keefektifan *YOLOv11* dalam sektor pertanian telah dibuktikan oleh penelitian yang dilakukan di Universitas Siliwangi dan Bandung (2025) menggunakan dataset berisi 737 citra daun tomat dengan berbagai jenis penyakit.

Hasil testing menunjukkan akurasi sebesar 75,6 %, *precision* sebesar 0,80, *recall* sebesar 0,77, serta *mAP@0.5* sebesar 75,6 %, sehingga memperkuat relevansi *YOLOv11* dalam deteksi penyakit tanaman (Al Husaini et al., 2025).

Meskipun fokus penelitian ini adalah pada klasifikasi penyakit daun padi, tahap deteksi objek tetap dilakukan sebagai langkah awal untuk memastikan area daun yang relevan teridentifikasi dengan tepat. Dengan demikian, *YOLOv11* tetap dimanfaatkan secara penuh, meskipun evaluasi utama pada hasil penelitian difokuskan pada klasifikasi akhir dari setiap area daun yang telah terdeteksi.

### 2.6.1 *YOLOv11* dalam Klasifikasi Penyakit pada Tanaman Padi

Pada *YOLOv11*, arsitektur jaringan telah disempurnakan untuk meningkatkan akurasi dan efisiensi dalam mendeteksi dan mengklasifikasikan objek, termasuk objek kecil pada citra yang memiliki latar belakang yang kompleks.

#### 1. Kemampuan Deteksi Objek dalam Waktu Nyata

Salah satu keunggulan utama *YOLOv11* adalah kemampuannya dalam melakukan deteksi objek dalam waktu nyata. Hal ini dimungkinkan karena *YOLOv11* membagi gambar menjadi grid dan memprediksi *bounding box* serta kelas objek secara bersamaan, dalam satu forward pass. Pada penelitian ini, *YOLOv11* digunakan untuk mendeteksi berbagai penyakit pada tanaman padi secara langsung, mengidentifikasi apakah daun padi sehat atau terinfeksi berbagai penyakit seperti *Bacterial Leaf Blight* atau *Leaf blast*. Proses ini sangat menguntungkan dalam aplikasi pertanian modern, di mana kecepatan pemantauan sangat penting.

#### 2. Penggunaan *Bounding box* untuk Lokalisasi Penyakit

Dalam deteksi penyakit pada tanaman, *YOLOv11* menggunakan *bounding box* untuk menentukan lokasi objek pada citra. Pada penelitian ini, *bounding box* digunakan untuk menandai bagian daun padi yang terinfeksi penyakit. Setiap penyakit yang terdeteksi akan diberi kotak pembatas (*bounding box*) dengan label yang sesuai (misalnya: *Bacterial Leaf Blight*, *Leaf Scald*, dll.), yang memungkinkan identifikasi dan analisis yang lebih mudah. Dengan menggunakan teknik ini, *YOLOv11* dapat mengidentifikasi lokasi spesifik dari

setiap penyakit pada gambar tanaman, yang memberikan informasi yang lebih detail untuk pemantauan dan pengendalian penyakit.

### 3. Kemampuan untuk Mengklasifikasikan Multi-Kelas

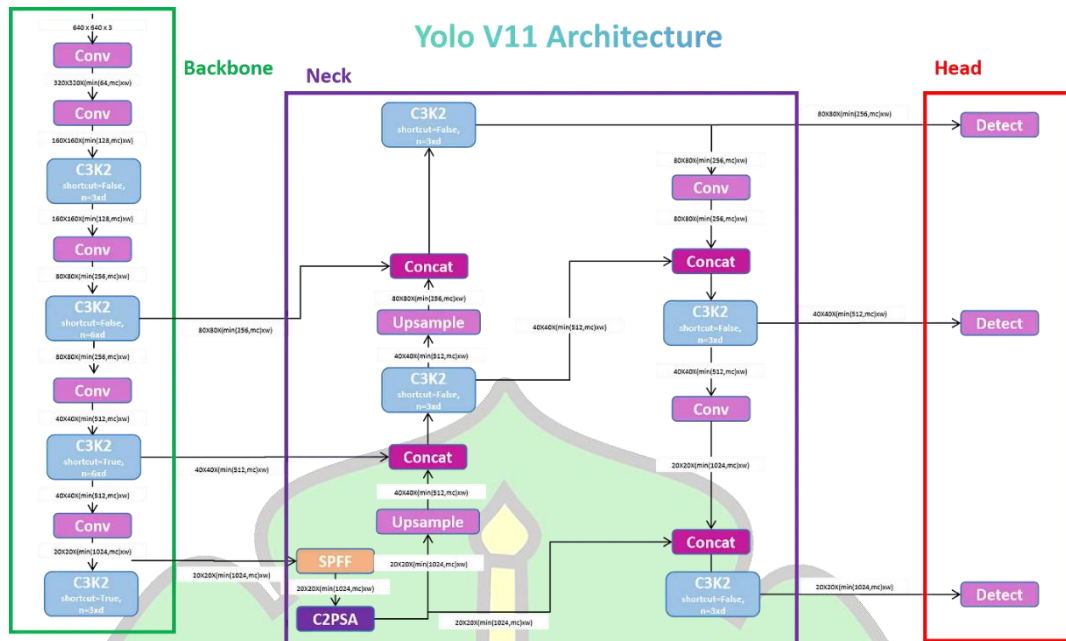
*YOLOv11* tidak hanya mampu mendeteksi objek, tetapi juga dapat mengklasifikasikan objek ke dalam berbagai kategori yang telah ditentukan. Dalam penelitian ini, *YOLOv11* mengklasifikasikan citra daun padi ke dalam empat kelas yang berbeda, termasuk *Bacterial Leaf Blight*, *Brown spot*, dan *Leaf blast*, serta kelas daun padi yang sehat. Sistem klasifikasi ini memungkinkan sistem untuk memberikan diagnosis yang lebih tepat dan spesifik terkait kondisi tanaman.

### 4. Peningkatan Akurasi dan Kecepatan dengan *YOLOv11*

Dalam penelitian ini, *YOLOv11* menawarkan keunggulan dalam hal akurasi dan kecepatan dibandingkan dengan metode tradisional. Dengan algoritma yang lebih efisien, *YOLOv11* dapat mengidentifikasi penyakit pada daun padi dengan tingkat akurasi yang tinggi, serta melakukannya dalam waktu yang relatif singkat. Hal ini sangat penting dalam aplikasi pertanian di dunia nyata, di mana diagnosis cepat dapat mengarah pada penanganan penyakit yang lebih efektif dan mencegah kerugian besar pada hasil panen.

### 5. Optimalisasi dan Penggunaan dalam *Platform Google Colaboratory*

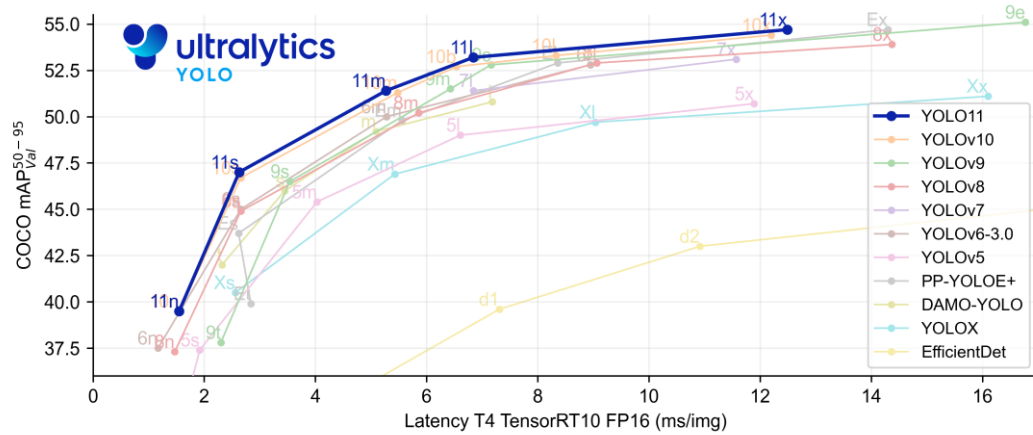
Dalam penelitian ini, pelatihan dan evaluasi model *YOLOv11* dilakukan menggunakan *platform Google Colaboratory*, yang memberikan akses gratis ke GPU untuk mempercepat proses pelatihan model. Dengan dukungan ini, *YOLOv11* dapat dilatih dengan lebih cepat dan efektif, menghasilkan model yang siap digunakan untuk deteksi dan klasifikasi penyakit pada tanaman padi dalam skala besar.



Gambar 2.5 Arsitektur YOLOv11

Arsitektur *YOLOv11* pada gambar tersebut terdiri dari tiga komponen utama, yaitu **Backbone**, **Neck**, dan **Head**. Pada tahap **Backbone**, citra masukan berukuran  $640 \times 640 \times 3$  diproses melalui serangkaian lapisan konvolusi (Conv) dan blok C3K2 untuk mengekstraksi ciri pada berbagai tingkat resolusi, dimulai dari resolusi tinggi hingga resolusi rendah. Selanjutnya, pada bagian **Neck** dilakukan penggabungan fitur (*Concat*) dan peningkatan ukuran fitur (*Upsample*) dari beberapa level resolusi, disertai penggunaan modul seperti SPFF dan C2PSA guna memperkaya representasi spasial serta konteks objek. Tahap terakhir, yaitu **Head**, memanfaatkan keluaran dari **Neck** untuk menghasilkan prediksi melalui lapisan deteksi (*Detect*) pada tiga skala berbeda, sehingga model dapat mengenali objek berukuran kecil, sedang, maupun besar dengan lebih optimal.

## 2.6.2 Perbandingan Performa YOLOv11 dengan Generasi Sebelumnya



Gambar 2.6 Perbandingan versi YOLO

Pada bagian ini dilakukan analisis perbandingan antara *YOLOv11* dan model pendahulunya, yaitu *YOLOv8*, *YOLOv9*, serta *YOLOv10*. Evaluasi dilakukan dengan mempertimbangkan beberapa indikator kinerja seperti *Mean Average Precision (mAP)*, kecepatan inferensi yang diukur dalam *Frames Per Second (FPS)*, dan efisiensi jumlah parameter yang digunakan. Perbandingan ini mencakup penerapan model pada berbagai tugas, baik deteksi maupun segmentasi objek, untuk melihat peningkatan akurasi, kecepatan, dan efisiensi yang ditawarkan oleh *YOLOv11* dibandingkan versi sebelumnya (Ramos & Sappa, 2025).

## 2.7 Roboflow

*Roboflow* adalah platform berbasis web yang mempermudah pengelolaan dataset dalam proyek *Computer Vision*, mencakup tahap pengolahan seperti anotasi gambar, augmentasi data, dan ekspor dataset ke format yang kompatibel dengan model deteksi objek seperti *YOLOv11*. Dalam penelitian ini, *Roboflow* memainkan peran penting di berbagai tahap praproses, termasuk:

### a. Anotasi Dataset

*Roboflow* menyediakan antarmuka grafis intuitif untuk pelabelan citra dengan *bounding box* serta klasifikasi kondisi seperti *Bacterial Leaf Blight* atau *Leaf blast*. Anotasi yang konsisten ini meningkatkan kualitas dataset dan mendukung model dalam mempelajari pola visual penyakit secara lebih akurat.

## **b. Augmentasi Citra**

Fitur augmentasi mencakup teknik seperti rotasi, flipping, blur, dan penyesuaian kecerahan. Studi oleh (Kukartsev et al., 2024) menunjukkan bahwa *Roboflow* memudahkan pipeline pelatihan YOLOv8 melalui anotasi dan augmentasi yang terstruktur, sehingga memperkuat robustnes model terhadap berbagai skenario visual(Kukartsev et al., 2024).

## **c. Ekspor Data Sesuai Format YOLOv11**

Setelah anotasi dan augmentasi, *Roboflow* mampu mengekspor dataset ke dalam format yang siap digunakan oleh *YOLOv11* termasuk file.txt untuk koordinat *bounding box* dan data.yaml untuk pengaturan label kelas, sehingga mempercepat integrasi dengan pipeline pelatihan dan evaluasi model.

## **d. Integrasi dengan Google Colab**

Dengan API-nya, *Roboflow* memungkinkan impor dataset langsung ke *Google Colaboratory*, memfasilitasi pelatihan model pada lingkungan GPU secara otomatis, efisien, dan tanpa masalah transfer data manual melalui kombinasi fitur-fitur tersebut, *Roboflow* mempercepat alur kerja penelitian *Deep Learning* dalam deteksi penyakit pada daun padi, menjadikan penelitian lebih efisien, *reproducible*, dan *scalable*.

## **2.8 Google Colaboratory**

*Google Colaboratory (Colab)* adalah *platform* cloud berbasis *Jupyter Notebook* yang menyediakan lingkungan pemrograman *Python* tanpa perlu instalasi lokal. *Platform* ini sangat populer di kalangan peneliti dan praktisi karena menyediakan akses gratis ke GPU, integrasi mulus dengan *Google Drive*, dan dukungan langsung terhadap pustaka-pustaka *Deep Learning* seperti *TensorFlow*, *PyTorch*, dan *Keras*. Selain itu, *Colab* juga memungkinkan kolaborasi secara *real-time*, mirip seperti *Google Docs* tetapi untuk coding, sehingga memudahkan tim melakukan kolaborasi dalam penelitian nyata.

- Dalam penelitian ini, *Google Colab* memiliki peran sentral dalam mendukung pelatihan model *YOLOv11* untuk klasifikasi penyakit pada daun padi, dengan manfaat berikut:

- Percepatan pelatihan model: Akses ke GPU cloud memungkinkan proses pelatihan *Deep Learning* yang kompleks menjadi lebih cepat tanpa perlu perangkat keras mahal.
- Kemudahan pengelolaan dataset: Dataset citra penyakit daun padi dapat diunggah dan diakses langsung dari *Google Drive*, sehingga menghindari proses transfer manual yang memakan waktu.
- Ketersediaan pustaka populer: *Colab* sudah dilengkapi dengan pustaka-pustaka penting seperti *TensorFlow* dan *OpenCV* sehingga peneliti dapat langsung fokus ke eksperimen tanpa ribet instalasi.
- Kolaborasi tim yang efisien: Dengan fitur berbagi *notebook*, seluruh tim dapat bekerja bersama secara *real-time*, melihat hasil, dan memberikan umpan balik dengan cepat.

Selain itu, *Colab* membantu penelitian melalui workflow modular untuk pemrosesan citra, seperti yang ditunjukkan oleh (Rippner et al., 2022), yang menggunakan *Colab* untuk segmentasi citra *X-ray* struktur tanaman dan tanah. Penelitian tersebut menunjukkan bahwa *platform* ini tidak hanya menghemat biaya dan meningkatkan efisiensi, tetapi juga memfasilitasi adopsi teknik *Deep Learning* dalam ilmu tanaman dan pertanian dengan sumber daya terbatas.

## 2.9 Python

*Python* adalah bahasa pemrograman tingkat tinggi yang terkenal dengan sintaksnya yang bersih dan mudah dipahami, sehingga menjadi pilihan utama dalam pengembangan aplikasi kecerdasan buatan, terutama di bidang *machine learning* dan *Deep Learning*. Fleksibilitas dan kemudahan penulisan kode menjadikannya populer di kalangan peneliti, ilmuwan data, dan praktisi teknologi.

*Python* dilengkapi dengan ekosistem pustaka yang sangat kaya. Sebagai contoh, *NumPy* dan *Pandas* menangani komputasi numerik dan manipulasi data; *Matplotlib* dan *Seaborn* mendukung visualisasi; *OpenCV* memfasilitasi pengolahan citra digital; serta *framework* seperti *TensorFlow*, *PyTorch*, dan *Ultralytics* memungkinkan implementasi model deteksi objek seperti *YOLOv11* secara efisien.

Dalam penelitian ini, *Python* menjadi pondasi untuk seluruh tahapan: pra-pemrosesan data, pelabelan gambar dengan *bounding box*, pelatihan model

*YOLOv11*, serta evaluasi performa dengan metrik seperti *mAP*, *precision*, *recall*, dan *IoU*, semuanya dilakukan dalam *notebook Python* yang terintegrasi dengan *Google Colab*. Kombinasi kemudahan scripting dan dukungan pustaka membuat *Python* sangat ideal untuk eksperimen berbasis GPU tanpa memerlukan hardware khusus.

Selain itu, dokumentasi yang lengkap dan komunitas pengguna yang besar memudahkan akses terhadap referensi teknis dan tutorial ketika menghadapi tantangan. Seperti dieksplorasi dalam kajian terkini, *Python* terus mendominasi dalam ekosistem AI dan data *science* berkat keterbukaannya, performa *prototyping* cepat, dan kompatibilitas lintas *platform* (Mantrala, 2025).

## 2.10 *Confusion matrix*

*Confusion matrix* adalah alat evaluasi utama dalam model klasifikasi yang membandingkan hasil prediksi model dengan label sebenarnya (*ground truth*). Dalam kasus klasifikasi multi-kelas, seperti deteksi penyakit daun padi menggunakan *YOLOv11*, *confusion matrix* berbentuk matriks  $n \times n$ , dengan  $n$  sebagai jumlah kelas. Setiap baris merepresentasikan kelas aktual, sedangkan kolom menunjukkan prediksi model. Nilai pada diagonal utama menandakan jumlah prediksi benar (*True Positive*) untuk tiap kelas, sedangkan nilai selain diagonal mencerminkan kesalahan prediksi berupa *False Positive* (FP) dan *False Negative* (FN). Penggunaan *confusion matrix* memungkinkan identifikasi kelas yang sering salah diprediksi, sehingga menjadi dasar perbaikan model melalui augmentasi data atau penyesuaian parameter (Sathyanarayanan, 2024).

Dari *confusion matrix*, beberapa metrik evaluasi dapat dihitung:

- Presisi (*Precision*)  
Mengukur seberapa banyak prediksi positif yang benar.

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

- *Recall*  
Mengukur kemampuan model dalam menemukan seluruh data positif.

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

- *F1-Score*

Merupakan rata-rata harmonis dari *precision* dan *recall*.

$$F1 = \frac{2x \text{ Precision } x \text{ Recall}}{\text{Precision } + \text{ Recall}}$$

- *mAP50*

Metrik ini menghitung rata-rata presisi dari seluruh kelas objek pada berbagai nilai IoU (*Intersection over Union*) threshold, yang biasanya diukur pada rentang tertentu seperti  $\text{IoU} = 0.5$ .

$$mAP50 = \frac{1}{K} \sum_{c=1}^K AP_c (\text{IoU} = 0.50)$$

Dengan:

- $K$  = Jumlah Kelas
- $AP_c$  = Average Precision untuk kelas ke- $c$  yang dihitung pada IoU threshold 0.50

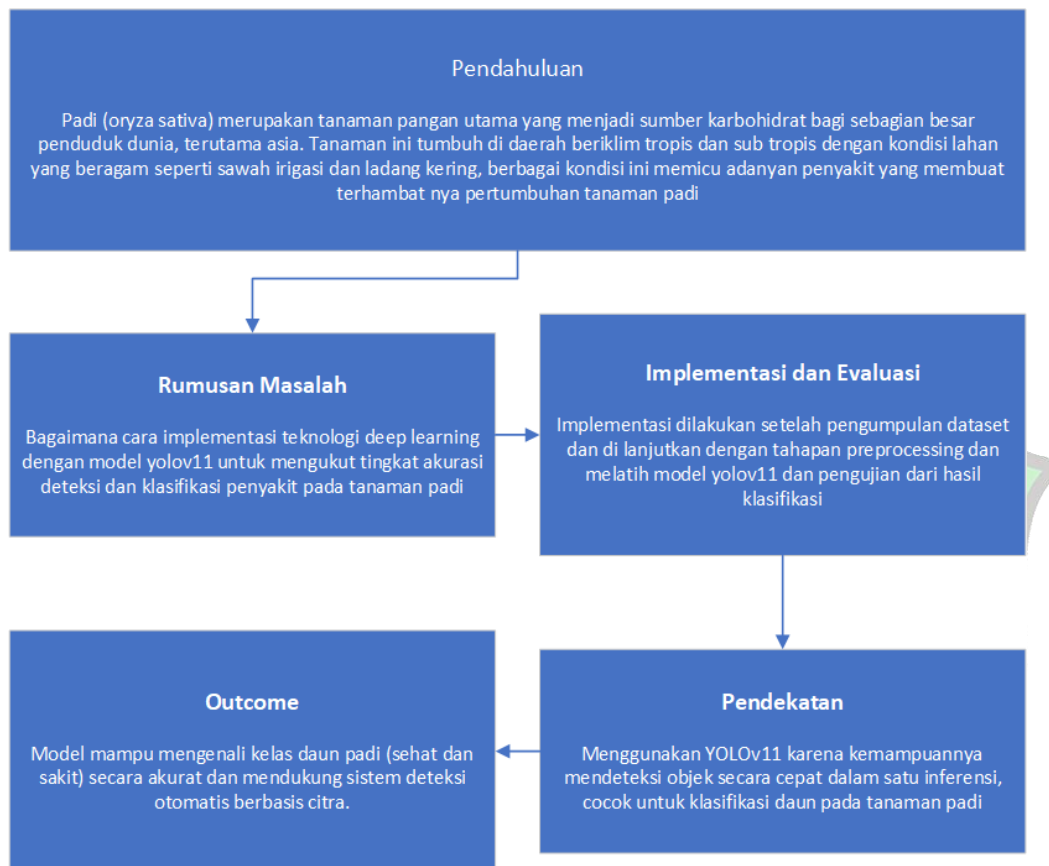
### 2.11 *mAP* (mean Average Precision)

*mAP* (*mean Average Precision*) adalah metrik evaluasi yang umum digunakan dalam object *Detection* seperti pada *YOLOv11*. Metrik ini menghitung rata-rata presisi dari seluruh kelas objek pada berbagai nilai IoU (*Intersection over Union*) *threshold*, yang biasanya diukur pada rentang tertentu seperti  $\text{IoU} = 0.5$ .

rata-rata dari AP semua kelas, dihitung pada ambang  $\text{IoU} = 0.5$  (apakah prediksi kotak cukup tumpang tindih dengan objek asli). Semakin tinggi nilai *mAP*, semakin baik model dalam menyeimbangkan antara *precision* dan *recall* pada semua kelas. Pada dasarnya, *mAP* memberikan gambaran menyeluruh mengenai kinerja deteksi model dengan mempertimbangkan tidak hanya apakah suatu objek terdeteksi, tetapi juga seberapa tepat letak *bounding box* yang dihasilkan. Dalam deteksi penyakit daun padi, *mAP* tinggi menunjukkan bahwa *YOLOv11* tidak hanya mampu mengenali jenis penyakit secara akurat, tetapi juga menempatkan *bounding box* dengan tepat sehingga meminimalkan kesalahan deteksi dan lokalisasi visual (*localization error*). Metrik ini sangat penting dalam aplikasi pertanian presisi karena akurasi deteksi yang tinggi memungkinkan pengambilan keputusan penanganan penyakit yang lebih cepat dan tepat, sebagaimana ditunjukkan dalam studi perbandingan performa berbagai detektor menggunakan *mAP* sebagai tolok ukur utama (Tian et al., 2024).

## 2.12 Kerangka Berpikir

Kerangka Pemikiran adalah struktur paradigma yang digunakan oleh penulis sebagai dasar untuk memahami, menganalisis, dan menyelesaikan masalah. Dalam kerangka ini, berbagai informasi dan konsep dirangkai secara logis menjadi satu kesatuan.



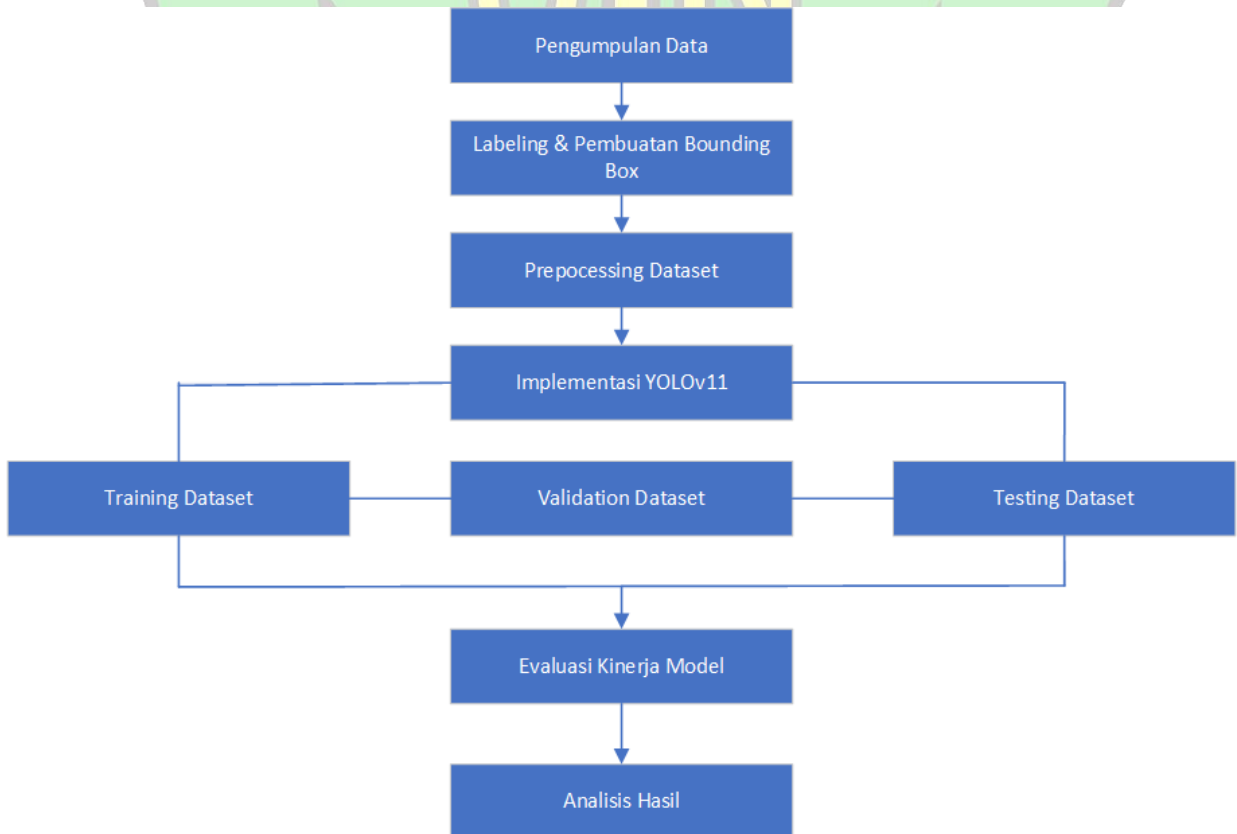
Gambar 2.7 Kerangka Berpikir

A R - R A N I R Y

## BAB III METODE PENELITIAN

### 3.1 Tahapan Penelitian

Tahapan penelitian merujuk pada serangkaian langkah sistematis yang harus ditempuh oleh peneliti dalam merancang, menjalankan, dan mengevaluasi suatu penelitian guna memperoleh hasil yang valid dan dapat dipertanggung jawabkan. Dalam penelitian ini, proses dilakukan secara bertahap mulai dari pengumpulan data citra daun padi, *preprocessing* data untuk memastikan kualitas input, pelatihan model *YOLOv11* untuk deteksi dan klasifikasi penyakit, hingga tahap evaluasi model menggunakan metrik yang sesuai seperti *mAP*, *precision*, *recall*, *F1-Score* dan *IoU*. Setiap tahapan dirancang agar saling berkesinambungan dan mendukung tercapainya tujuan penelitian.



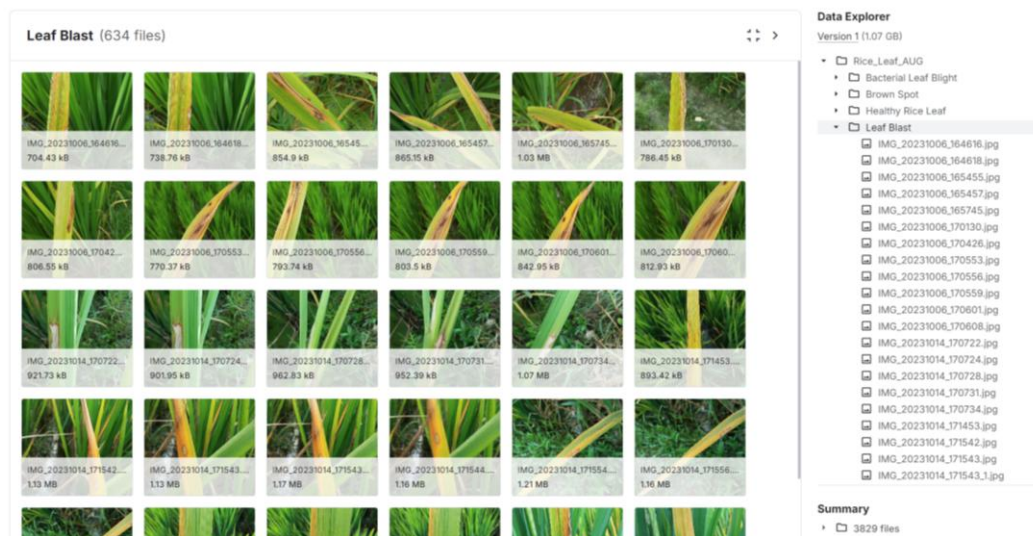
Gambar 3.1 Tahapan Penelitian

### 3.2 Pengumpulan Data

Pada tahap ini, Dataset citra daun padi yang digunakan dalam penelitian ini diperoleh melalui pengumpulan data sekunder dari beberapa sumber daring terpercaya. Sumber-sumber tersebut meliputi *Kaggle*, *GitHub*, *Mendeley Data*, dan *Google Dataset Search* ([Rice Disease Dataset](#)). Pemilihan dataset dari sumber daring ini bertujuan untuk memperoleh jumlah data yang memadai serta variasi visual yang mendukung proses pelatihan model deteksi berbasis *YOLOv11*. Pengumpulan data dilakukan dengan cara mengunduh kumpulan gambar yang telah tersedia secara publik dan relevan dengan empat kelas penyakit daun padi yang dikaji yaitu:

- *Brown spot* (*Cochliobolus miyabeanus*)
- *Healthy Rice Leaf* (Daun Padi Sehat)
- *Bacterial Leaf Blight* (*Xanthomonas oryzae pv. oryzae*)
- *Leaf blast* (*Pyricularia oryzae*)

Berikut merupakan salah satu contoh sumber pengumpulan dataset dapat dilihat pada Gambar 3.2.



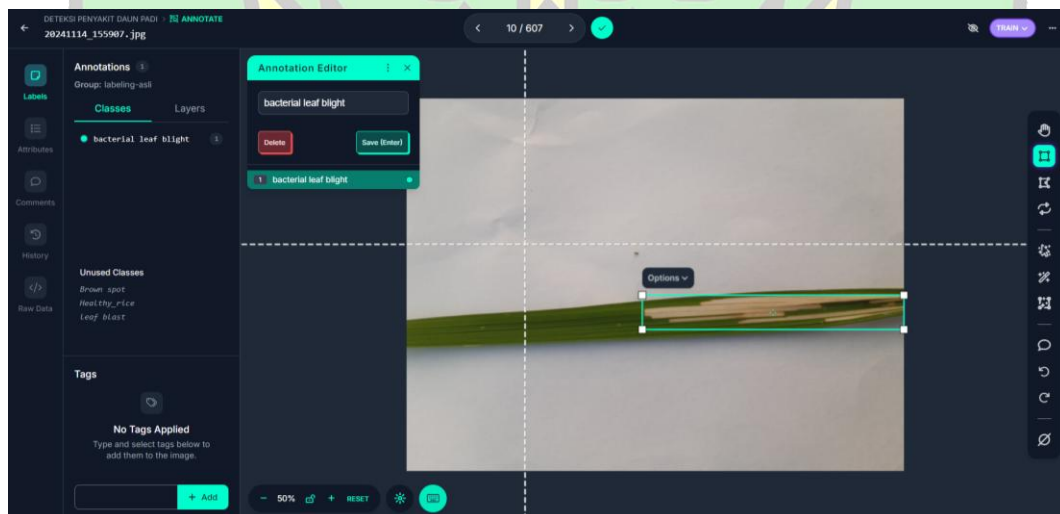
Gambar 3.2 Sumber pengumpulan dataset

### 3.3 Labeling dan Pembuatan Bounding Box

Setelah proses pengumpulan data selesai, tahap selanjutnya adalah melakukan proses *labeling* dan pembuatan *bounding box* pada setiap citra. Tujuan dari proses ini adalah untuk menandai bagian tertentu dari gambar yang

menunjukkan keberadaan penyakit pada daun padi, sehingga model dapat belajar mengenali lokasi serta kelas objek tersebut secara akurat. *Bounding box* adalah kotak persegi panjang yang mengelilingi area objek yang relevan pada gambar. Dalam penelitian ini, *bounding box* digunakan untuk menandai bagian daun yang mengalami gejala penyakit, seperti bercak, warna abnormal, atau pola kerusakan lainnya. Proses ini dilakukan secara manual menggunakan *tools* anotasi seperti *Roboflow*. *Roboflow* dipilih karena menyediakan antarmuka yang mudah digunakan untuk memberi anotasi pada citra daun padi yang menunjukkan gejala penyakit, serta memungkinkan penyimpanan otomatis dalam format yang kompatibel dengan *YOLOv11*. Setiap citra dianotasi dengan *bounding box* dan diberi label sesuai dengan empat kelas yaitu 3 kelas penyakit dan 1 kelas yang sehat sebagai perbandingannya dan hasil *labeling* ini akan digunakan sebagai data pelatihan.

Setiap gambar diberi *bounding box* secara manual untuk memastikan ketepatan posisi objek yang akan dipelajari oleh model *YOLOv11*. Contoh pelabelan dapat dilihat pada Gambar 3.3.



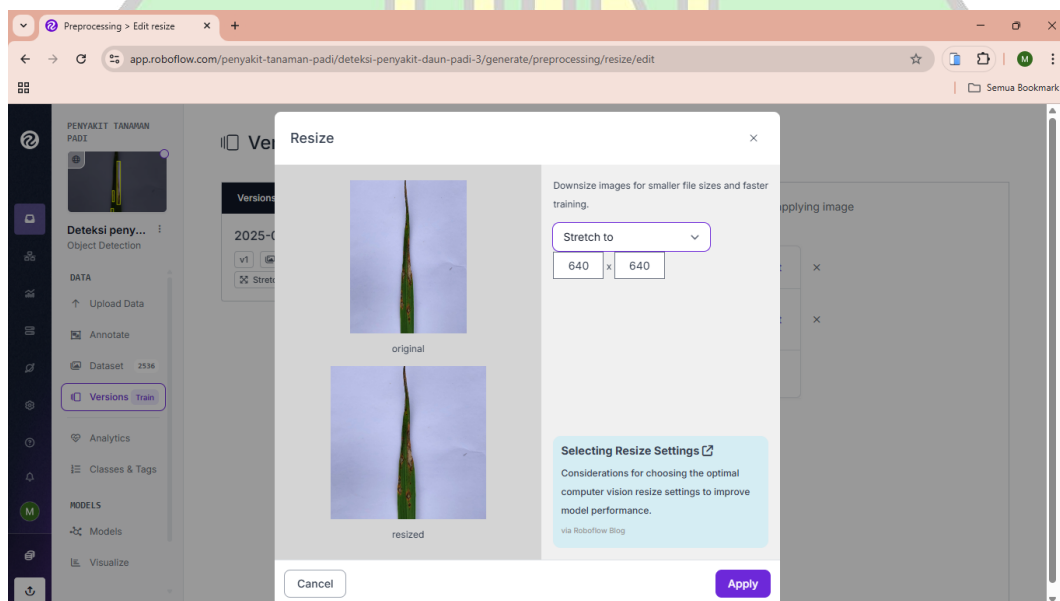
Gambar 3.3 Proses Bounding Box

### 3.4 Preprocessing data

Tahap *preprocessing* data dilakukan untuk menyiapkan citra agar sesuai dengan format dan kebutuhan model *YOLOv11*. Proses ini sangat penting karena berpengaruh langsung terhadap hasil pelatihan dan performa model dalam mendeteksi serta mengklasifikasikan penyakit pada daun padi. Langkah-langkah *preprocessing* yang dilakukan dalam penelitian ini mencakup:

## A. Resize Gambar

Tahapan *resize* gambar dilakukan penyeragaman dimensi seluruh dataset agar sesuai dengan ukuran input yang dibutuhkan oleh model *YOLOv11*, yaitu  $640 \times 640$  piksel. Proses ini menggunakan metode *stretch to 640x640*, yang berarti gambar akan ditarik atau disesuaikan ukurannya tanpa mempertahankan rasio aspek aslinya. Meskipun metode ini dapat menyebabkan sedikit distorsi pada citra, namun tetap dipilih karena kesesuaiannya dengan format input model serta kemudahan implementasinya melalui *platform Roboflow*. Penyeragaman ukuran ini penting agar model dapat memproses data secara konsisten dan mencegah error saat pelatihan. Contoh *resize* gambar dengan metode *stretch to 640x640*

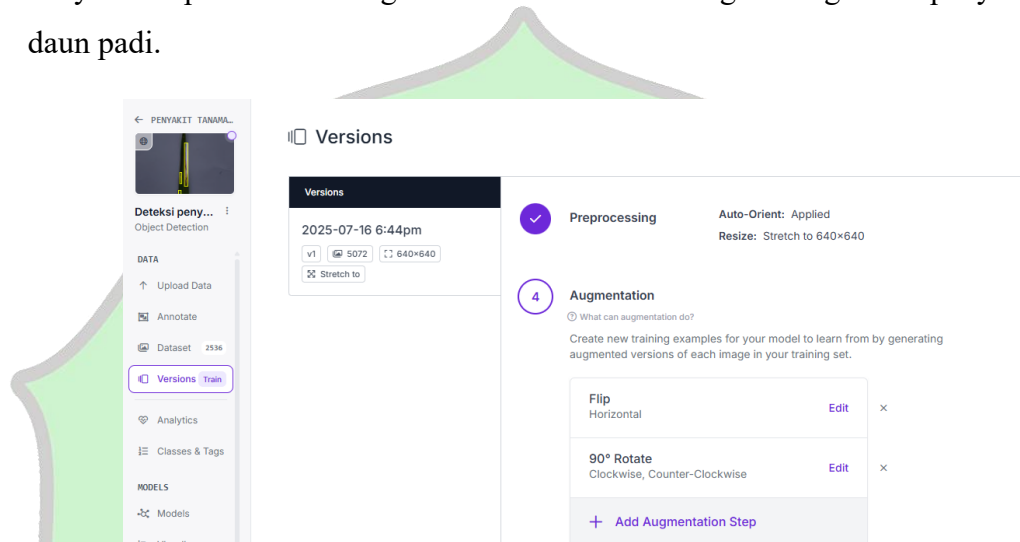


Gambar 3.4 Proses Resize Gambar

## B. Augmentasi Data

Setelah proses *resize*, tahap selanjutnya adalah melakukan augmentasi gambar untuk memperbanyak variasi data dan meningkatkan kemampuan generalisasi model saat pelatihan. Teknik augmentasi yang digunakan dalam penelitian ini meliputi *flip horizontal* dan *rotate 90 derajat*, yang diterapkan menggunakan fitur dari *platform Roboflow*. Teknik *flip horizontal* membalik

gambar secara mendatar, sementara *rotate* memutar gambar sebanyak 90° searah jarum jam. Kombinasi teknik ini menghasilkan variasi visual yang tetap relevan dengan label kelasnya, namun memiliki sudut pandang yang berbeda. Hasil dari proses augmentasi ini menyebabkan jumlah dataset bertambah menjadi dua kali lipat dari jumlah dataset asli. Penambahan ini dilakukan sebelum proses pelatihan model dimulai, sehingga model mendapatkan lebih banyak sampel dalam mengenali karakteristik masing-masing kelas penyakit daun padi.



Gambar 3.5 Proses Augmentasi Dataset

### C. Konversi Format Anotasi ke YOLO Format

Label dari setiap gambar dikonversi ke dalam format YOLO (.txt) yang memuat kelas objek serta koordinat *bounding box* dalam skala relatif (antara 0 sampai 1). Dalam penelitian ini, ukuran gambar ditetapkan pada 640×640 piksel, dan anotasi *bounding box* dipilih dalam format YOLO (dengan urutan: *class, x\_center, y\_center, width, height*). File hasil ekspor berupa folder lengkap dengan file .txt untuk label serta file data.yaml yang berisi informasi konfigurasi dataset seperti jumlah kelas, nama kelas, dan path direktori. dataset kemudian diekspor dengan format YOLOv5 *PyTorch* agar kompatibel langsung dengan format input dari *Ultralytics YOLOv11*. *Roboflow* menyediakan berbagai opsi format saat proses ekspor, namun pada penelitian ini dipilih format YOLOv5 karena menggunakan struktur file dan format anotasi yang sama dengan arsitektur *YOLOv11*

### 3.5 Split dataset (pembagian dataset)

Pembagian dataset adalah langkah penting dalam pelatihan model *machine learning*, termasuk deteksi objek menggunakan *YOLOv11*. Tujuan dari pembagian dataset adalah untuk memastikan model dapat belajar dengan baik dan diuji dengan data yang tidak terlihat sebelumnya, sehingga dapat mengukur seberapa baik performa model dalam kondisi dunia nyata. Pada penelitian ini, proses split dilakukan secara manual menggunakan script *Python* di *Google Colab*, dengan proporsi pembagian sebesar 70% untuk data *Training* (latih), 15% untuk data *validation* (validasi), dan 15% untuk data *testing* (uji). Pembagian ini dilakukan setelah proses augmentasi selesai, sehingga jumlah data yang digunakan sudah mencakup gambar hasil augmentasi. Proses split dilakukan menggunakan pustaka *Python* seperti *os*, *shutil*, dan *random*, yang memungkinkan pemindahan file gambar secara acak ke direktori masing-masing. Pembagian ini dilakukan secara adil agar distribusi kelas tetap seimbang pada ketiga subset data. Berikut merupakan tabel split dataset.

Tabel 3.1 Pembagian Dataset

No	Jenis Dataset	Total gambar	Jumlah Dataset Training (70%)	Jumlah Dataset Validasi (15%)	Jumlah Dataset Testing (15%)
1	<i>Brown spot (Cochliobolus miyabeanus)</i> – Bercak Coklat	1378	964	207	207
2	<i>Healthy Rice Leaf</i>	1308	915	196	196
3	<i>Bacterial Leaf Blight (Xanthomonas oryzae pv. oryzae)</i> – Hawar Daun Bakteri	1214	849	183	183
4	<i>Leaf blast (Pyricularia oryzae)</i> – Blas daun	1172	820	176	176
Total		5072	3548	762	762

Proses dilakukan secara acak namun tetap mempertahankan proporsi setiap kelas, serta telah diverifikasi jumlah file-nya agar sesuai dengan rasio pembagian. Berikut merupakan skrip *Python* yang digunakan untuk split dataset

```
Atur path dan parameter

[ ] # Ganti dengan path dataset kamu
base_path = '/content/drive/MyDrive/untuk split dataset 70:15:15'
images_path = os.path.join(base_path, 'images')
labels_path = os.path.join(base_path, 'labels')

# Path output hasil split
output_path = os.path.join(base_path, 'Hasil_split_dataset_70:15:15')
os.makedirs(output_path, exist_ok=True)

# Folder tujuan
for split in ['train', 'val', 'test']:
    os.makedirs(os.path.join(output_path, 'images', split), exist_ok=True)
    os.makedirs(os.path.join(output_path, 'labels', split), exist_ok=True)

# Rasio split
train_ratio = 0.70
val_ratio = 0.15
test_ratio = 0.15

Lakukan split dan salin file ke folder

# Ambil semua file gambar
image_files = [f for f in os.listdir(images_path) if f.endswith('.jpg')]
random.shuffle(image_files)

total = len(image_files)
train_end = int(train_ratio * total)
val_end = train_end + int(val_ratio * total)

train_files = image_files[:train_end]
val_files = image_files[train_end:val_end]
test_files = image_files[val_end:]

def copy_files(file_list, split):
    for img_file in file_list:
        label_file = img_file.replace('.jpg', '.txt')

        shutil.copy(os.path.join(images_path, img_file),
                    os.path.join(output_path, 'images', split, img_file))

        shutil.copy(os.path.join(labels_path, label_file),
                    os.path.join(output_path, 'labels', split, label_file))

copy_files(train_files, 'train')
copy_files(val_files, 'val')
copy_files(test_files, 'test')
```

Gambar 3.6 Proses Split Dataset di Google Colab

### 3.5 Implementasi *YOLOv11*

Pelatihan model *YOLOv11* dilakukan menggunakan platform *Google Colaboratory* dengan memanfaatkan integrasi ke *Google Drive* sebagai media penyimpanan dataset, konfigurasi, serta hasil pelatihan. Selanjutnya, model *YOLOv11s* dimuat menggunakan perintah `model = YOLO('yolo11s.pt')` sebagai arsitektur awal. Proses pelatihan dilakukan dengan parameter `imgsz=640` untuk menyesuaikan ukuran input gambar, `epochs=200` sebagai jumlah total pelatihan, serta `batch=8` agar tidak membebani memori GPU *Colab*. File hasil pelatihan akan

otomatis disimpan setiap 10 *epoch* melalui parameter `save_period=10`, dan nama proyek ditentukan agar hasilnya tersimpan rapi di direktori yang telah ditentukan. Apabila pelatihan sempat berhenti akibat limitasi *runtime Colab*, proses dapat dilanjutkan kembali dari *checkpoint* terakhir (*last.pt*) dengan menggunakan parameter `resume=True`. Berikut merupakan potongan *Source code* untuk melakukan *Train* dan *Validation* model.

```
from ultralytics import YOLO

# Load YOLOv11s model
model = YOLO('yolo11s.pt')

# Training
model.train(
    data='/content/drive/MyDrive/untuk_split_dataset_70:15:15/Hasil_split_dataset_70:15:15/data.yaml',
    epochs=200,
    imgsz=640,
    batch=8, # Gunakan batch lebih kecil agar hemat GPU
    name='yolo11_split_fix', # Hasil simpan di runs/detect/yolo11_split_fix
    project='/content/drive/MyDrive/YOLOv11_Project/runs', # Simpan di Drive
    device='0', # Gunakan GPU
    save_period=10, # Simpan tiap 10 epoch
)
```

Gambar 3.7 Source code Untuk Train dan Validation Model

Apabila proses pelatihan terhenti secara tiba-tiba akibat kendala teknis seperti batas waktu penggunaan GPU pada *Google Colab*, maka pelatihan dapat dilanjutkan dari *checkpoint* terakhir tanpa harus mengulang dari awal. Hal ini dilakukan dengan memuat model terakhir yang tersimpan (*last.pt*) dan menambahkan parameter `resume=True` pada fungsi pelatihan. Dengan demikian, model akan melanjutkan pelatihan dari *epoch* terakhir yang tercatat sebelumnya, sehingga efisiensi waktu dan sumber daya tetap terjaga.

```

from ultralytics import YOLO

# Load YOLOv11s model
model = YOLO('/content/drive/MyDrive/YOLOv11_Project/runs/yolo11_split_fix2/weights/last.pt')

# Training
model.train(
    data='/content/drive/MyDrive/untuk_split_dataset_70:15:15/Hasil_split_dataset_70:15:15/data.yaml',
    epochs=200,
    imgsz=640,
    batch=8, # Gunakan batch lebih kecil agar hemat GPU
    name='yolo11_split_fix', # Hasil simpan di runs/detect/yolo11_split_fix
    project='/content/drive/MyDrive/YOLOv11_Project/runs', # Simpan di Drive
    device='0', # Gunakan GPU
    save_period=10, # Simpan tiap 10 epoch
    resume=True
)

```

Gambar 3.8 Source code Untuk Melanjutkan Proses Training dan validation

### 3.6 Training, Validation dan Testing Model

Setelah implementasi model *YOLOv11*, langkah selanjutnya adalah melakukan *training*, *validation*, dan *testing*, untuk memastikan bahwa model dapat mendeteksi dan mengklasifikasikan penyakit pada daun padi dengan akurat. Berikut penjelasan lebih detail mengenai masing-masing tahap ini:

#### 3.6.1 Proses *Training* dan *Validation*

Tahap pelatihan dilakukan menggunakan Dataset *Training* yang telah dilabeli sebelumnya dan Dataset *Validation*. Training model dilakukan menggunakan *YOLOv11s.pt*, yaitu varian ringan (small) dari *YOLOv11*. Pemilihan varian ini didasarkan pada pertimbangan jumlah dataset yang tersedia serta kebutuhan efisiensi pelatihan tanpa mengorbankan akurasi secara signifikan. Selama proses pelatihan, model secara otomatis melakukan evaluasi pada *validation* set di setiap *epoch*. Output yang ditampilkan di terminal mencakup progres pelatihan, nilai *loss* (*box loss*, *obj loss*, *cls loss*), serta metrik evaluasi seperti *Precision* (P), *Recall* (R) dan *mAP50*. Proses ini memudahkan pemantauan performa model dari waktu ke waktu, sekaligus memastikan bahwa model tidak mengalami *overfitting* terhadap data latih.

Selama proses pelatihan dan validasi yang dilakukan selama 200 *epoch*, model mencatat perkembangan metrik performa pada setiap *epoch*, termasuk nilai *loss* dan *mean Average Precision* (*mAP*). Berdasarkan hasil pelatihan yang

ditampilkan secara otomatis oleh library *Ultralytics*, model menunjukkan peningkatan kinerja yang signifikan. Untuk tampilan hasil *Train* dan *Validation* di terminal dapat dilihat pada gambar 3.9.

```

split dataset 70:15:15.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan
Q Perintah + Kode + Teks ▶ Jalankan semua ▼

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
194/200 2.83G 1.111 0.6371 1.057 10 640: 100% ██████████ 444/444 [01:37<00:00, 4.56it/s]
      Class Images Instances Box(P) R mAP50 mAP50-95): 100% ██████████ 48/48 [00:09<00:00, 4.85it/s]

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
195/200 2.83G 1.1 0.6373 1.054 7 640: 100% ██████████ 444/444 [01:36<00:00, 4.60it/s]
      Class Images Instances Box(P) R mAP50 mAP50-95): 100% ██████████ 48/48 [00:10<00:00, 4.53it/s]

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
196/200 2.83G 1.1 0.6345 1.049 11 640: 100% ██████████ 444/444 [01:37<00:00, 4.55it/s]
      Class Images Instances Box(P) R mAP50 mAP50-95): 100% ██████████ 48/48 [00:10<00:00, 4.60it/s]

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
197/200 2.83G 1.108 0.6317 1.053 10 640: 100% ██████████ 444/444 [01:36<00:00, 4.62it/s]
      Class Images Instances Box(P) R mAP50 mAP50-95): 100% ██████████ 48/48 [00:11<00:00, 4.32it/s]

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
198/200 2.83G 1.099 0.6296 1.058 13 640: 100% ██████████ 444/444 [01:35<00:00, 4.64it/s]
      Class Images Instances Box(P) R mAP50 mAP50-95): 100% ██████████ 48/48 [00:10<00:00, 4.53it/s]

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
199/200 2.83G 1.089 0.6271 1.048 10 640: 100% ██████████ 444/444 [01:34<00:00, 4.72it/s]
      Class Images Instances Box(P) R mAP50 mAP50-95): 100% ██████████ 48/48 [00:11<00:00, 4.33it/s]

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
200/200 2.83G 1.082 0.6208 1.044 17 640: 100% ██████████ 444/444 [01:37<00:00, 4.55it/s]
      Class Images Instances Box(P) R mAP50 mAP50-95): 100% ██████████ 48/48 [00:10<00:00, 4.41it/s]

```

Gambar 3.9 Tampilan Hasil Train dan Validation Model di Terminal

Gambar diatas juga menunjukkan bahwa Waktu yang dibutuhkan untuk menyelesaikan satu *epoch* bervariasi, di mana pada awal pelatihan, proses memerlukan waktu yang relatif lebih lama, yakni mencapai sekitar 25-30 menit. Namun, seiring berjalannya pelatihan, durasi per *epoch* menjadi lebih singkat dan stabil, rata-rata hanya memakan waktu sekitar 1 menit 36 detik. Hal ini disebabkan oleh proses inisialisasi awal dan pemuatan data yang lebih berat pada *epoch* pertama.

### 3.6.2 Proses Testing Model

Proses *Testing Dataset* dilakukan menggunakan model terbaik (*best.pt*) yang diperoleh dari proses *Training*, dengan tujuan untuk mengukur kemampuan model dalam melakukan prediksi pada data yang benar-benar baru dan belum pernah dilihat sebelumnya. Dataset *testing* ini mencakup 15% dari keseluruhan data, yang terpisah sepenuhnya dari data *Training* maupun *validation*, sehingga hasil evaluasi dapat memberikan gambaran objektif mengenai performa generalisasi model di luar data latih. Berikut merupakan *Source code* yang digunakan untuk menjalankan proses *testing* di terminal yang dapat dilihat pada gambar 3.3.

```
from ultralytics import YOLO

# Load model terbaik
model = YOLO('/content/drive/MyDrive/YOLOv11_Project/runs/yolo11_split_fix2/weights/best.pt')

# Jalankan evaluasi di dataset test, simpan langsung ke Drive
model.val(
    data='/content/drive/MyDrive/untuk_split_dataset_70:15:15/Hasil_split_dataset_70:15:15/data.yaml',
    split='test',
    imgsz=640,
    batch=8,
    device='0',
    project='/content/drive/MyDrive/HASIL TEST', # Folder tujuan di Drive
    name='val_test', # Nama subfolder
    save_json=True, # Simpan hasil COCO-format
    save_hybrid=True # Simpan confusion matrix dll.
)
```

Gambar 3.10 Source code untuk proses Testing

Pada gambar 3.10 testing model dilakukan menggunakan dataset *testing* yang berfungsi untuk mengevaluasi performa akhir model pada data yang benar-benar baru dan tidak pernah digunakan selama proses pelatihan maupun validasi. Proses ini penting untuk mengukur kemampuan model dalam mengenali data pada kondisi nyata. Evaluasi dilakukan menggunakan model hasil pelatihan terbaik (*best.pt*) dengan perintah yang ditampilkan pada *Source code*.

### 3.7 Evaluasi Kinerja Model

Evaluasi kinerja model dilakukan untuk mengetahui sejauh mana model *YOLOv11* mampu mendeteksi dan mengklasifikasikan penyakit pada daun tanaman padi secara akurat. Proses evaluasi ini penting agar hasil yang diperoleh dapat diukur secara objektif dan dapat dipercaya. Pada tahap ini, evaluasi dilakukan dengan menggunakan metrics seperti

- *Precision*: Mengukur akurasi deteksi, yaitu berapa banyak deteksi yang benar dibandingkan dengan semua deteksi yang dibuat oleh model.
- *Recall*: Mengukur sensitivitas model, yaitu berapa banyak objek yang terdeteksi dengan benar dibandingkan dengan semua objek yang sebenarnya ada dalam gambar.
- *mAP50* (mean Average Precision): Digunakan untuk mengukur keakuratan model dalam mendeteksi dan mengklasifikasikan objek (penyakit tanaman).

- *F1-Score*: Merupakan rata-rata harmonis antara *precision* dan *recall*, digunakan untuk menilai keseimbangan antara keduanya.

### 3.8 Analisis Hasil

Analisis hasil dilakukan untuk menafsirkan performa model *YOLOv11* berdasarkan hasil evaluasi yang telah diperoleh. Setelah model dilatih dan diuji menggunakan dataset yang telah melalui proses *preprocessing* dan augmentasi, diperoleh hasil berupa nilai *precision*, *recall*, *mAP*, dan *F1-Score*. Nilai-nilai ini menjadi acuan untuk menilai kekuatan dan kelemahan model dalam mendeteksi serta mengklasifikasikan penyakit pada daun padi. Proses analisis ini tidak hanya melihat seberapa tinggi nilai metrik yang diperoleh, tetapi juga mempertimbangkan kestabilan model terhadap data testing, seperti konsistensi dalam mengenali enam kelas penyakit yang telah ditetapkan. Jika ditemukan nilai yang kurang optimal pada salah satu metrik atau kelas tertentu, maka akan dicari kemungkinan penyebabnya, seperti jumlah data per kelas yang tidak seimbang atau kompleksitas visual dari gejala penyakit tersebut. Dengan adanya analisis ini, peneliti dapat memahami efektivitas model secara lebih menyeluruh, serta mendapatkan gambaran tentang potensi pengembangan atau perbaikan pada penelitian berikutnya.

### 3.9 Waktu dan Lokasi Penelitian

Waktu penelitian dilakukan dalam jangka waktu dari bulan Desember 2024 hingga bulan Agustus 2025. Penelitian ini dibagi menjadi beberapa fase, termasuk dari pengumpulan dataset, perancangan model, pengujian dan evaluasi hingga ke tahap *deployment*.

Penelitian ini dilaksanakan pada laboratorium program studi Teknologi Informasi Universitas Islam Negeri Ar-Raniry Banda Aceh.

### 3.10 Alat dan Bahan

Alat yang diperlukan untuk melaksanakan penelitian ini terdiri dari perangkat keras dan perangkat lunak. Analisis mengenai alat dan spesifikasi sistem yang dibutuhkan untuk penelitian ini mencakup:

### 3.11 Perangkat Keras

Perangkat keras yang digunakan berupa satu buah unit Laptop Lenovo Idea Pad 3 14IML05 dengan spesifikasi yang dipaparkan pada tabel 3.2

Tabel 3.2 Spesifikasi Perangkat Keras

Komponen	Spesifikasi
Processor	Intel Core i3-10110U
Ram	4GB Soldered DDR4
Storage	512 GB SSD
<i>Graphics Processing Unit</i>	Intel UHD Graphics 2GB

### 3.12 Perangkat Lunak

Perangkat lunak yang dipakai dalam proses implementasi berupa sistem operasi *Microsoft Windows Windows 11 Home Single Language 23H2* juga menggunakan *tools* seperti, *Google Colab*, *Python*, *TensorFlow* dan *Chrome.Tools* tersebut akan dipakai dalam penelitian yang akan dilakukan oleh penulis, dapat dilihat dari tabel 3.3.

Tabel 3.3 Spesifikasi Perangkat Lunak

Perangkat Lunak	Version
Windows 11 Home Single Language	23H2
<i>Google Colab</i>	3.10
<i>TensorFlow</i>	2.17
<i>Python</i>	3.10.11
Chrome	132.0.6834.196

## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Hasil Evaluasi Dataset *Training* dan *Validation*

Berikut merupakan hasil evaluasi pada data Train dan Validation yang digunakan untuk menilai performa model pada setiap kelas. Evaluasi ini mencakup metrik utama seperti precision, recall, F1-score, serta mAP@50 yang secara keseluruhan memberikan gambaran mengenai kemampuan model dalam mendeteksi dan mengklasifikasikan penyakit tanaman padi. Hasil lengkap evaluasi ditunjukkan pada Tabel 4.1.

Tabel 4.1 Tabel Hasil Evaluasi Training dan Validation

No	Kelas	<i>mAP50</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
1	<i>Brown spot</i>	0,726	0,79	0,632	0,702
2	<i>Healthy Rice</i>	0,98	0,93	0,952	0,940
3	<i>Bacterial Leaf Blight</i>	0,745	0,774	0,711	0,741
4	<i>Leaf blast</i>	0,795	0,732	0,717	0,724

Berdasarkan Tabel tersebut, dapat disimpulkan bahwa model *YOLOv11* mampu memberikan performa deteksi yang sangat baik pada kelas *Healthy Rice* dengan nilai *mAP50*, *precision*, dan *recall* yang tinggi, menunjukkan kemampuan model mengenali kelas ini secara konsisten dan akurat. Sementara itu, performa pada kelas *Brown spot*, *Bacterial Leaf Blight*, dan *Leaf blast* menunjukkan hasil yang cukup baik, meskipun nilai *recall* dan *F1-Score* masih berada di bawah kelas *Healthy Rice*. Secara keseluruhan, performa model dapat dikategorikan baik dan layak digunakan, dengan catatan perbaikan lebih lanjut pada kelas-kelas dengan nilai *recall* yang relatif rendah

#### 4.6.1 Identifikasi Peningkatan Akurasi pada Beberapa Epoch

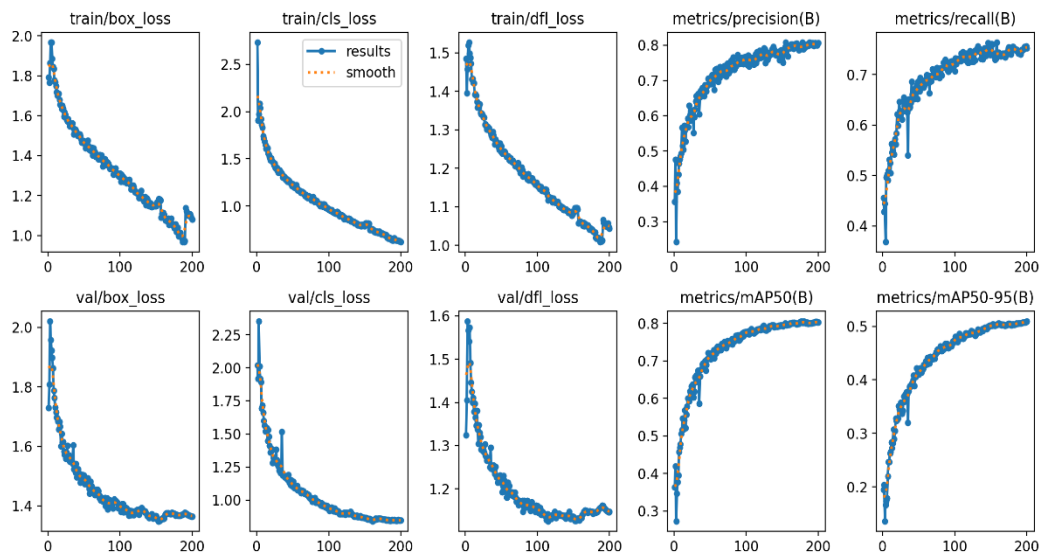
Berdasarkan hasil *Train* dan *validation*, akurasi model menunjukkan peningkatan signifikan antara *epoch* ke-1 hingga *epoch* ke-200 yang dapat dilihat pada tabel berikut.

Tabel 4.2 Peningkatan Akurasi di Beberpaa Epoch

No	<i>Epoch</i>	<i>mAP50</i>	keterangan
1	1	0.363	Awal pelatihan dan akurasi masih rendah
2	20	0.600	Terjadi peningkatan yang signifikan
3	46	0.705	Akurasi terus meningkat dengan stabil
4	161	0.800	Peningkatan mencapai nilai tertinggi sementara
5	200	0.803	Akurasi yang naik sedikit dari sebelumnya

#### 4.6.2 Hasil *Loss* dan Evaluasi Model

Berdasarkan hasil *Train* dan *Validation* model *YOLOv11* dengan proses pelatihan sebanyak 200 *epoch*. Grafik ini memvisualisasikan perkembangan nilai *loss* dan metrik evaluasi global dari keseluruhan kelas yang digunakan. Dengan menampilkan grafik ini, dapat diketahui bagaimana model belajar dalam mendeteksi serta mengklasifikasikan objek pada dataset penyakit daun padi pada setiap *epoch*. Analisis ini juga penting untuk melihat kestabilan model, mendeteksi potensi *overfitting*, dan memahami tren performa model secara menyeluruh sebelum melangkah ke evaluasi detail per kelas.



Gambar 4.1 Gambar Grafik Loss, Precision, Recall, dan mAP

**Loss (box\_loss, cls\_loss, dfl\_loss)**, yaitu sumbu Y merepresentasikan nilai error yang dihitung antara prediksi model dan label sebenarnya. Nilai loss yang tinggi di awal (misalnya  $>2.0$ ) menunjukkan bahwa prediksi masih banyak kesalahan, namun seiring bertambahnya epoch nilainya menurun (mendekati 1.0 atau lebih kecil), yang berarti model semakin mampu mengenali pola data dengan benar. Semakin rendah loss, semakin baik kemampuan model dalam melakukan prediksi.

**Metrics (precision, recall, mAP50, mAP50-95)**, yaitu sumbu Y merepresentasikan tingkat performa model dalam rentang 0–1. Nilai 0 berarti performa sangat buruk, sedangkan nilai mendekati 1 berarti performa sangat baik. Misalnya, precision = 0.8 berarti 80% deteksi yang dihasilkan model benar. Peningkatan kurva pada metrik ini menunjukkan bahwa model tidak hanya mengurangi kesalahan (loss rendah), tetapi juga semakin akurat dalam melakukan deteksi dan klasifikasi.

Dengan demikian, hubungan antara sumbu X dan Y pada grafik dapat menjelaskan proses pembelajaran model: semakin banyak epoch, loss cenderung menurun, sedangkan nilai metrik cenderung meningkat, menandakan model menjadi lebih stabil dan akurat.

- **Train/box\_loss** menunjukkan nilai *loss* untuk prediksi *bounding box* pada data latih. Terlihat nilai *loss* menurun secara konsisten dari awal pelatihan

(1.9) hingga akhir (1.0). Ini menandakan bahwa model semakin baik dalam menyesuaikan *bounding box* dengan posisi objek sebenarnya di gambar.

- **Val/box\_loss** pada data validasi juga mengalami penurunan dari sekitar 2.0 ke 1.3. Penurunan ini relatif stabil seiring *epoch* bertambah, menandakan kemampuan generalisasi model pada data yang tidak dilatih cukup baik.
- **Train/cls\_loss** Grafik ini menggambarkan nilai *loss* klasifikasi pada data latih. Dari awal 2.6 turun ke 1.0, menunjukkan model belajar semakin akurat dalam mengklasifikasikan objek ke kelas yang tepat.
- **Val/cls\_loss** *Loss* klasifikasi pada validasi turun dari 2.2 ke sekitar 1.1. Ini mendukung kesimpulan bahwa model tidak hanya menghafal data latih, tetapi juga mampu mengklasifikasikan data baru
- **Train/df\_l\_loss & Val/df\_l\_loss (Distribution Focal Loss)** berkaitan dengan prediksi distribusi *bounding box*. Baik grafik *Train* maupun *val* menunjukkan penurunan *loss* yang konsisten ke nilai 1.1, menandakan model semakin presisi dalam memperkirakan letak objek.
- **Precision** meningkat drastis dari sekitar 0.3 pada awal pelatihan ke mendekati 0.8 di *epoch* akhir, menandakan penurunan jumlah *False Positive*. Ini memperlihatkan bahwa model semakin jarang salah mendeteksi objek yang tidak ada.
- **Recall** juga meningkat dari 0.4 ke 0.8, menunjukkan bahwa model makin baik dalam menemukan semua objek yang benar-benar ada di gambar (mengurangi *False Negative*).
- **mAP50** Nilai mean Average *Precision* pada IoU 0.5 naik tajam dari 0.3 ke 0.8, memperlihatkan akurasi tinggi dalam mendeteksi objek pada *threshold* IoU yang umum digunakan.
- **mAP50-95** Metrik ini mengukur rata-rata *precision* di berbagai *threshold* IoU (0.5 sampai 0.95), yang naik dari 0.2 ke 0.5. Ini menunjukkan model mampu mempertahankan akurasi meski pada kriteria evaluasi yang lebih ketat.

Grafik loss (box\_loss, cls\_loss, dfl\_loss) pada tahap training menunjukkan pola penurunan seiring bertambahnya epoch. Hal ini terjadi karena selama proses pelatihan model melakukan optimisasi parameter dengan backpropagation, sehingga bobot model diperbarui secara bertahap untuk meminimalkan error. Penurunan nilai loss mengindikasikan bahwa model semakin mampu menyesuaikan prediksi dengan data latih. Sementara itu, peningkatan pada grafik metrik seperti precision, recall, dan mAP menunjukkan bahwa model tidak hanya belajar mengurangi kesalahan, tetapi juga meningkatkan kemampuan generalisasi dalam mendeteksi dan mengklasifikasi objek sesuai target.

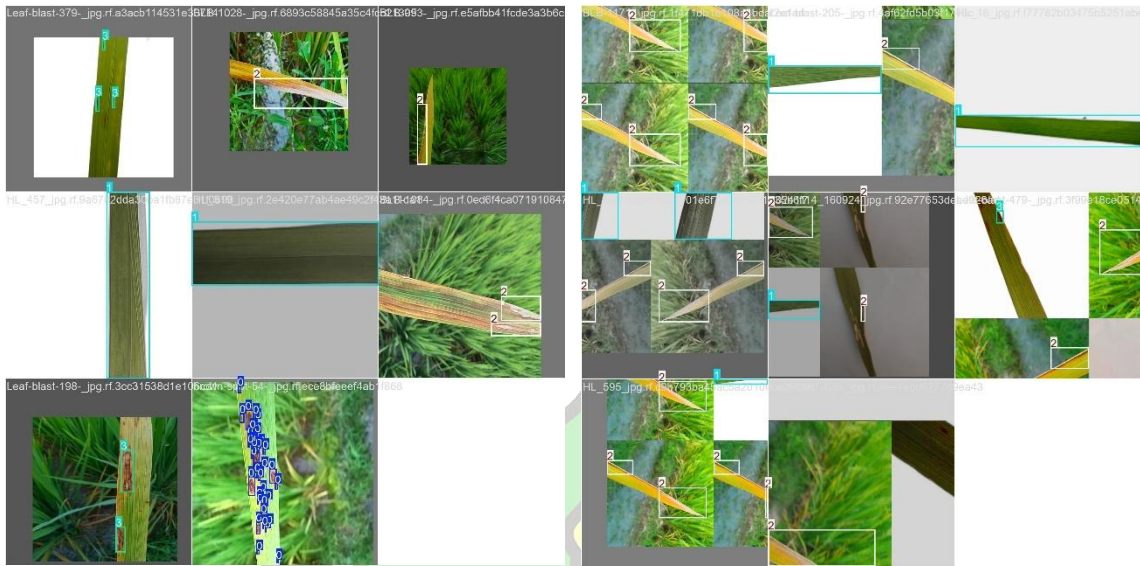
Grafik loss tidak ditampilkan pada data uji (testing) karena pada tahap ini model tidak lagi melakukan proses optimisasi parameter. Model hanya melakukan forward pass, yaitu menghitung prediksi dari input gambar tanpa propagasi balik (backpropagation). Oleh sebab itu, evaluasi pada data uji lebih relevan ditunjukkan melalui metrik performa seperti precision, recall, F1-score, mAP, confusion matrix, dan PR curve, bukan melalui loss function.

#### 4.6.3 Visualisasi *Train Batch*

Visualisasi *Train batch* merupakan hasil keluaran dari proses pelatihan model yang menampilkan citra latih beserta *bounding box* yang sudah dilabeli sesuai dengan kelas masing-masing objek. Dalam penelitian ini, terdapat empat kelas penyakit daun padi, yaitu:

- 0 = *Brown spot*
- 1 = *Healthy Rice*
- 2 = *Bacterial Leaf Blight* - R A N I R Y
- 3 = *Leaf blast*

Setiap *bounding box* pada gambar pelatihan dilengkapi dengan label angka yang merepresentasikan kelas sesuai daftar di atas. Warna *bounding box* biasanya dibedakan antar kelas untuk memudahkan identifikasi visual, sehingga memudahkan peneliti memastikan bahwa proses pelabelan dan anotasi berjalan sesuai target.



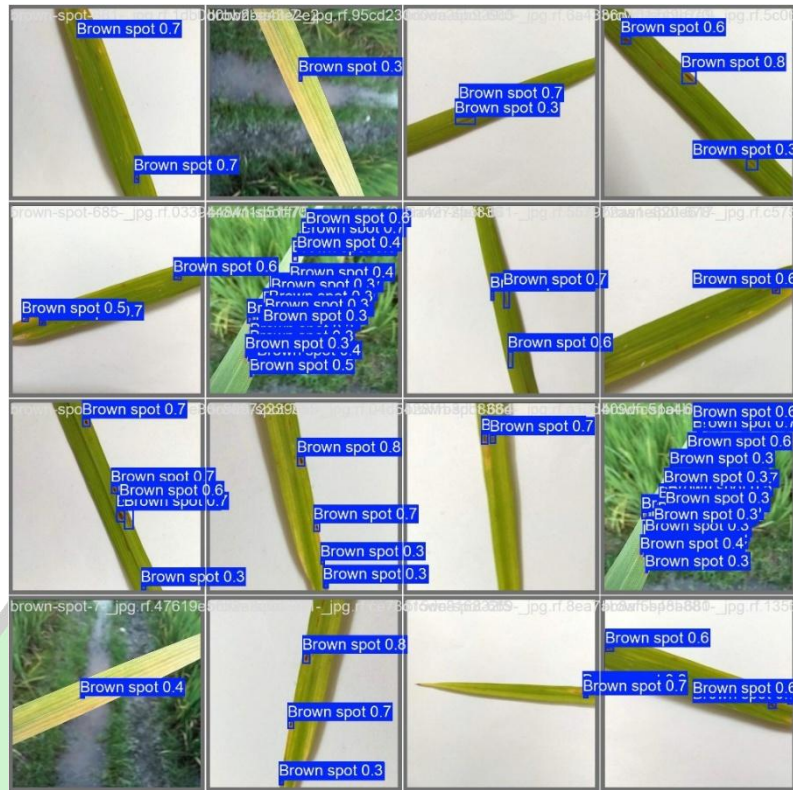
Gambar 4.2 Visualisasi Train Batch

#### 4.6.4 Visualisasi Prediksi dari *Validation Batch*

Berikut ditampilkan hasil prediksi model *YOLOv11* pada dataset validasi. Setiap gambar memperlihatkan *bounding box* yang mengelilingi area daun padi dengan label kelas hasil prediksi beserta nilai *confidence score* yang menunjukkan tingkat keyakinan model terhadap prediksi tersebut. Visualisasi ini memberikan gambaran nyata mengenai kemampuan model dalam mendeteksi dan mengklasifikasikan keempat kelas daun padi yang telah dilatih, yaitu *Leaf blast*, *Brown spot*, *Rice Hispa*, dan *Healthy Rice Leaf*. Contoh-contoh prediksi ini sekaligus menjadi bukti implementasi model *Deep Learning* secara praktis pada gambar asli, sehingga mempermudah dalam menilai keakuratan dan kehandalan sistem dalam mendeteksi penyakit pada tanaman padi. Berikut merupakan gambar prediksi dari setiap kelas:

A R - R A N I R Y

## 1. *Brown spot*



Gambar 4.3 Hasil Prediksi Kelas *Brown spot*

Gambar 4.3 menunjukkan hasil visualisasi prediksi model *YOLOv11* pada *batch* validasi untuk kelas *Brown spot*. Dimana setiap daun padi yang terdeteksi diberikan *bounding box* berwarna biru beserta nilai *confidence* dari prediksi model. Nilai *confidence* berkisar antara 0,3 hingga 0,8, yang menunjukkan tingkat keyakinan model terhadap keberadaan penyakit *Brown spot* pada area tertentu di daun. Beberapa gambar menunjukkan prediksi yang lebih jelas dengan *confidence* tinggi ( $\geq 0,7$ ), sementara sebagian area lain memiliki *confidence* yang lebih rendah (0,3–0,5), menandakan kemungkinan ambiguitas atau variasi visual pada gejala penyakit. Visualisasi ini membantu mengevaluasi kemampuan model dalam mendeteksi *Brown spot* secara lokal pada daun, sekaligus mengidentifikasi area yang mungkin memerlukan perbaikan dataset atau penyesuaian *threshold* prediksi.

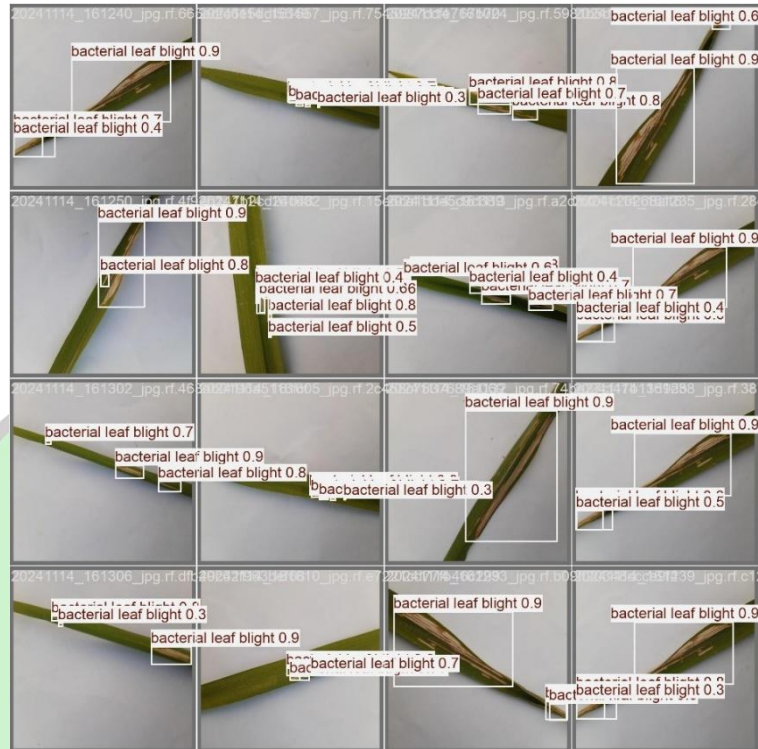
## 2. *Healthy Rice*



Gambar 4.4 Hasil Prediksi Kelas Healty Rice

Gambar 4.4 menampilkan hasil visualisasi prediksi model *YOLOv11* pada *batch* validasi untuk kelas *Healthy Rice*. Setiap daun yang terdeteksi sehat diberi *bounding box* berwarna hijau beserta nilai *confidence* prediksi. Nilai *confidence* berada pada rentang 0,78 hingga 0,88, menunjukkan tingkat keyakinan model yang relatif tinggi dan konsisten dalam mengenali daun sehat. Hampir seluruh deteksi berhasil mengidentifikasi ciri visual khas daun tanpa gejala penyakit, meskipun terdapat satu prediksi dari kelas lain yang muncul sebagai *False Positive*. Visualisasi ini mengindikasikan bahwa model memiliki kinerja yang baik pada kategori ini, namun tetap memerlukan perhatian pada kasus yang dapat memicu kesalahan deteksi.

### 3. Bacterial Leaf Blight



Gambar 4.5 Hasil Prediksi Kelas Bacterial Leaf Blight

Gambar 4.5 memperlihatkan hasil prediksi model *YOLOv11* pada data validasi untuk kelas *Bacterial Leaf Blight*. Setiap bagian daun yang terdeteksi diberi *bounding box* berwarna putih lengkap dengan label kelas serta nilai *confidence*. Rentang nilai *confidence* berada antara 0,3 hingga 0,9, mencerminkan variasi tingkat keyakinan model dalam mengidentifikasi gejala penyakit pada area tertentu. Deteksi dengan nilai di atas 0,7 menunjukkan keyakinan tinggi, sedangkan nilai di kisaran 0,3–0,5 mengindikasikan adanya kemungkinan ambiguitas atau kemiripan visual dengan kondisi lain tapi masih tergolong baik. Visualisasi ini memudahkan analisis performa model, sekaligus memberikan gambaran bagian mana yang masih berpotensi ditingkatkan melalui penambahan data atau penyesuaian parameter prediksi.

#### 4. *Leaf blast*



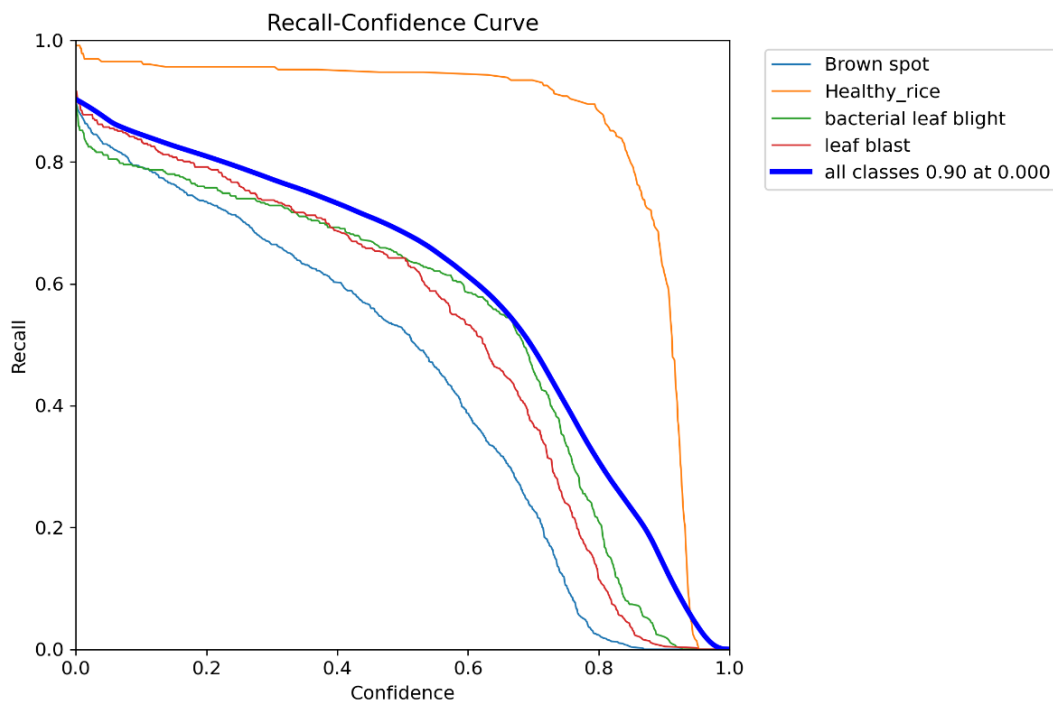
Gambar 4.6 Hasil Prediksi Kelas Leaf blast

Gambar 4.6 memperlihatkan hasil visualisasi prediksi model *YOLOv11* pada *batch* validasi untuk kelas *Leaf blast*. Setiap area daun yang terdeteksi memiliki gejala penyakit diberi *bounding box* berwarna oranye beserta label kelas dan nilai *confidence*. Nilai *confidence* berkisar antara 0,3 hingga 0,9, yang menunjukkan variasi tingkat keyakinan model dalam mengenali gejala pada setiap citra. Deteksi dengan nilai tinggi (0,9) umumnya ditemukan pada bagian daun dengan gejala yang jelas terlihat, sedangkan nilai yang lebih rendah mengindikasikan adanya ambiguitas atau gejala yang kurang tegas yang memungkinkan adanya kemiripan secara visual dengan kelas *Brown spot*. Visualisasi ini bermanfaat untuk menilai kinerja model pada kelas ini sekaligus mengidentifikasi area yang berpotensi ditingkatkan melalui penambahan data atau penyesuaian parameter pelatihan.

#### 4.6.5 Kurva dan Metrik Evaluasi Dataset Training dan Validation

##### A. Recall confidence curve

Pada Visualisasi *Recall-Confidence* berikut memperlihatkan kurva yang menunjukkan hubungan antara nilai *confidence* model dengan tingkat *recall* pada data validasi. Kurva ini merepresentasikan kemampuan model *YOLOv11* dalam mendeteksi objek (daun padi) pada berbagai *threshold confidence*. Berdasarkan gambar, terlihat bahwa nilai *recall* menurun seiring meningkatnya *threshold confidence*, yang berarti semakin tinggi kepercayaan prediksi yang ditetapkan, semakin sedikit objek yang berhasil terdeteksi oleh model. Kurva ini penting untuk mengevaluasi *trade-off* antara akurasi deteksi dan jumlah objek yang berhasil dikenali, sehingga dapat membantu dalam menentukan *threshold confidence* optimal yang menghasilkan keseimbangan terbaik antara *False Positive* dan *False Negative*.



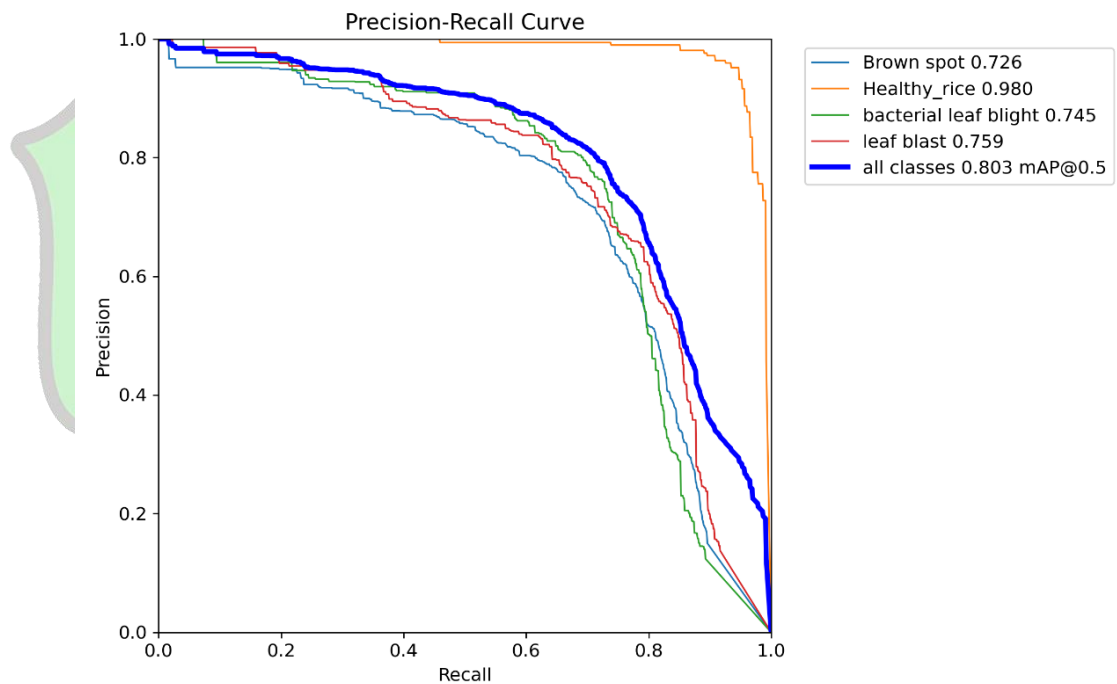
Gambar 4.7 Grafik Recall Confidence

Pada gambar 4.7 di atas menunjukkan nilai *recall* tertinggi mencapai sekitar 0,98 pada *threshold confidence* rendah (0,001–0,05), menunjukkan sensitivitas model yang tinggi terhadap mendeteksi objek. Namun, nilai *recall* menurun

signifikan ke kisaran 0,6–0,7 saat *threshold* mendekati 0,5, dan turun mendekati 0 pada *threshold* di atas 0,9. Hal ini menggambarkan adanya *trade-off* antara tingkat keyakinan prediksi dengan kemampuan mendeteksi seluruh objek yang ada.

### B. Precision-Recall Curve

Selanjutnya pada *Precision-Recall Curve* berikut memperlihatkan hubungan antara nilai *precision* dan *recall* pada data validasi untuk setiap kelas yang digunakan dalam pelatihan model *YOLOv11*. Kurva ini menunjukkan performa model dalam mendeteksi objek secara tepat (*precision*) sekaligus kemampuannya untuk menemukan sebanyak mungkin objek yang ada (*recall*) pada berbagai *threshold confidence*.



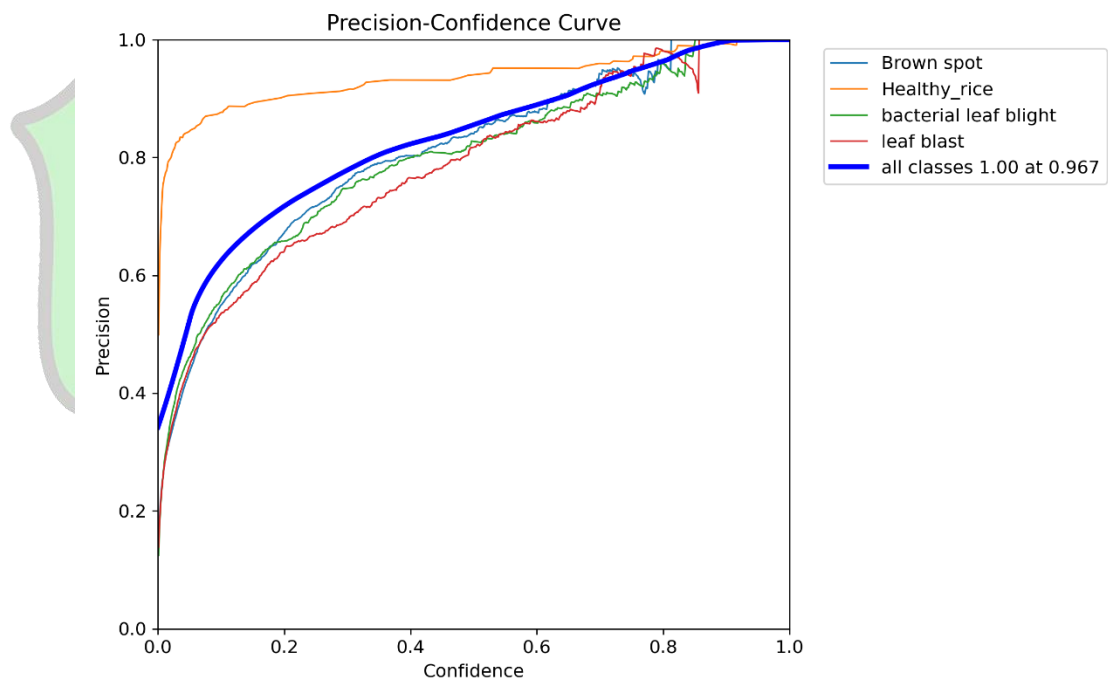
Gambar 4.8 Grafik Precision Recall

Dari gambar 4.8 tersebut Terlihat bahwa setiap kelas memiliki kurva dengan karakteristik yang berbeda. Kelas *Healthy\_Rice* memiliki *precision-recall* yang paling stabil dengan area mendekati nilai 1 pada sebagian besar rentang *recall*, menunjukkan performa deteksi yang sangat baik ( $mAP@0.5 = 0.980$ ). Sementara kelas lainnya, seperti *Brown spot*, *Bacterial Leaf Blight*, dan *leaf blast*, memiliki area kurva yang lebih rendah dengan  $mAP@0.5$  masing-masing sebesar 0.726, 0.745, dan 0.759 dengan nilai rata-rata presisi pada semua kelas (*all classes*)

mencapai  $mAP@0.5$  sebesar 0.803, yang menunjukkan model ini mampu mendeteksi objek dengan tingkat ketepatan yang cukup tinggi secara keseluruhan. Kurva ini penting untuk mengevaluasi kemampuan model dalam menghadapi *trade-off* antara mendeteksi objek secara akurat dan mendeteksi sebanyak mungkin objek yang ada pada berbagai tingkat keyakinan.

### C. Precision-Confidence Curve

*Precision-Confidence Curve* berikut menunjukkan hubungan antara nilai *precision* dengan *confidence threshold* pada data validasi. Kurva ini menggambarkan seberapa tepat model *YOLOv11* dalam melakukan prediksi pada berbagai tingkat kepercayaan (*confidence*) yang diterapkan saat mendeteksi objek.



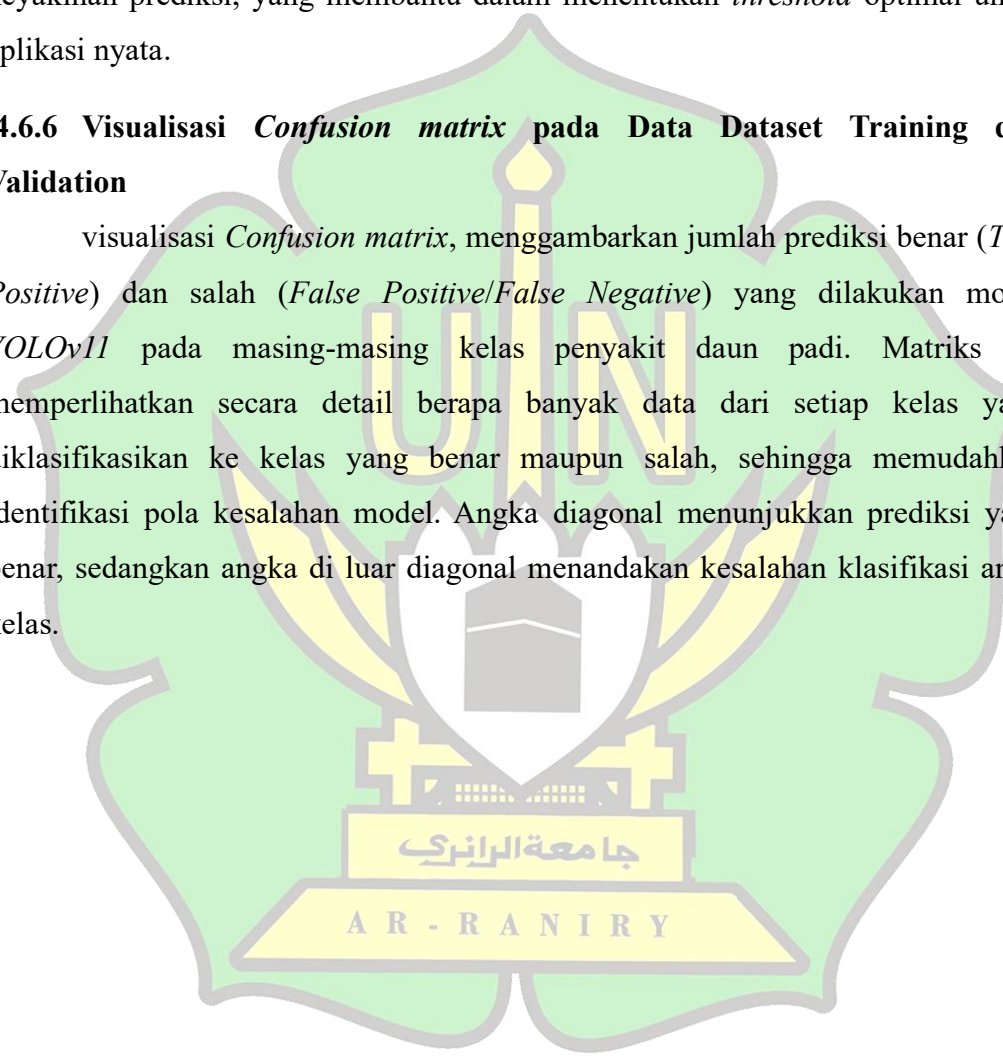
Gambar 4.9 Grafik Precision-Confidence

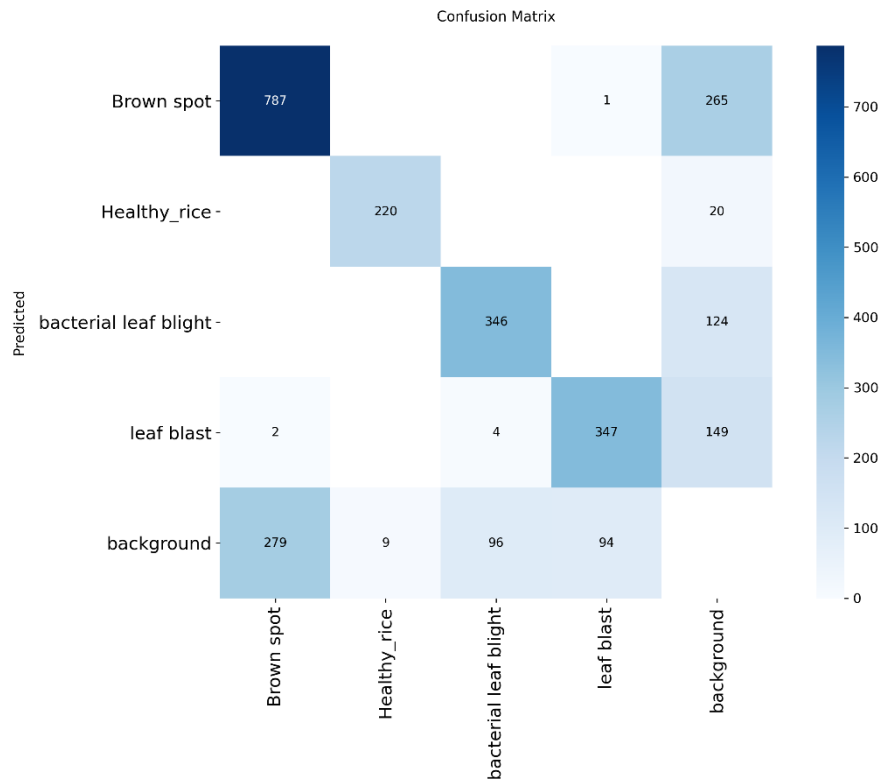
Terlihat pada gambar 4.9 bahwa *precision* untuk kelas *Healthy\_Rice* mendominasi dengan nilai yang tinggi dan stabil pada seluruh rentang *confidence*, menunjukkan kemampuan model mendeteksi kelas ini dengan sangat sedikit kesalahan (*False Positive*). *Precision* untuk kelas lainnya, yaitu *Brown spot*, *Bacterial Leaf Blight*, dan *leaf blast*, meningkat secara konsisten seiring dengan kenaikan *confidence threshold*, namun tetap menunjukkan gap dibanding *Healthy\_Rice*. Secara keseluruhan, *precision* rata-rata pada semua kelas mendekati

nilai 1,0 pada *confidence threshold* di atas 0,96, sesuai dengan nilai yang ditunjukkan pada kurva garis tebal biru (all *classes* 1.00 at 0.967). Hal ini menunjukkan bahwa ketika hanya prediksi dengan *confidence* tinggi yang dipertahankan, kesalahan prediksi menurun drastis, meskipun dapat menyebabkan hilangnya beberapa deteksi objek yang memiliki *confidence* rendah. Visualisasi ini penting untuk memahami bagaimana *precision* model berubah terhadap tingkat keyakinan prediksi, yang membantu dalam menentukan *threshold* optimal untuk aplikasi nyata.

#### 4.6.6 Visualisasi *Confusion matrix* pada Data Dataset Training dan Validation

visualisasi *Confusion matrix*, menggambarkan jumlah prediksi benar (*True Positive*) dan salah (*False Positive/False Negative*) yang dilakukan model *YOLOv11* pada masing-masing kelas penyakit daun padi. Matriks ini memperlihatkan secara detail berapa banyak data dari setiap kelas yang diklasifikasikan ke kelas yang benar maupun salah, sehingga memudahkan identifikasi pola kesalahan model. Angka diagonal menunjukkan prediksi yang benar, sedangkan angka di luar diagonal menandakan kesalahan klasifikasi antar kelas.





Gambar 4.10 Tampilan Confution Matrix

Dari Gambar 4.10 menunjukkan *confusion matrix* hasil prediksi model *YOLOv11s* terhadap lima kelas, yaitu *Brown spot*, *Healthy Rice*, *Bacterial Leaf Blight*, *Leaf blast*, dan *Background*. Nilai diagonal (dari kiri atas ke kanan bawah) merepresentasikan jumlah prediksi benar (*True Positive*) untuk masing-masing kelas, sedangkan nilai di luar diagonal menunjukkan jumlah kesalahan prediksi (*False Positive* atau *False Negative*). Dari hasil tersebut terlihat bahwa Kelas *Brown spot* memiliki prediksi benar sebesar 787 sampel, namun masih terdapat kesalahan klasifikasi ke kelas *Leaf blast* (265 sampel) dan *Background* (279 sampel). kemudian Kelas *Healthy Rice* terdeteksi benar sebanyak 220 sampel, namun terdapat kesalahan ke kelas *Bacterial Leaf Blight* (346 sampel) dan *Leaf blast* (124 sampel). Selanjutnya pada Kelas *Bacterial Leaf Blight* terdeteksi benar sebanyak 346 sampel, dengan kesalahan terbesar ke kelas *Leaf blast* (149 sampel). dan terakhir Kelas *Leaf blast* memiliki 347 prediksi benar, namun juga terdapat kesalahan cukup besar ke kelas *Background* (94 sampel). Kelas *Background* juga terdeteksi benar sebanyak 94 sampel, namun sering diklasifikasikan keliru sebagai *Brown spot* (279 sampel).

Secara umum, nilai *True Positive* yang cukup tinggi pada setiap kelas menunjukkan bahwa model mampu mengenali sebagian besar objek dengan benar. Namun, masih terdapat *misclassification* terutama pada kelas yang memiliki kemiripan visual, seperti antara *Brown spot* dan *Leaf blast* atau antara *Healthy Rice* dan *Bacterial Leaf Blight*. Hal ini mengindikasikan perlunya peningkatan kualitas dan variasi dataset, khususnya pada kondisi pencahayaan dan latar belakang yang berbeda, untuk mengurangi tingkat kesalahan prediksi.

#### 4.7 Hasil Evaluasi Dataset *Testing*

Untuk hasil evaluasi performa model pada dataset *testing* yang memuat informasi jumlah gambar dan *instance* perkelas, serta nilai *Precision*, *Recall*, *mAP50*, dan *F1-Score* untuk masing-masing kelas dapat dilihat pada Tabel 4.3.

Tabel 4.3 Hasil evaluasi dataset testing

No	Kelas	<i>mAP50</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
1	<i>Brown spot</i>	0,677	0,746	0,599	0,664
2	<i>Healthy Rice</i>	0,982	0,901	0,974	0,936
3	<i>Bacterial Leaf Blight</i>	0,757	0,714	0,711	0,712
4	<i>Leaf blast</i>	0,772	0,73	0,743	0,736

##### 4.7.1 Visualisasi Prediksi Dataset *Testing*

Bagian ini menyajikan visualisasi hasil prediksi model pada dataset *testing*, di mana setiap gambar menampilkan objek yang berhasil terdeteksi oleh model lengkap dengan label kelas dan nilai kepercayaan (*confidence score*). Visualisasi ini menjadi pelengkap dari hasil evaluasi kuantitatif, karena memberikan bukti visual mengenai sejauh mana model mampu mengenali dan mengklasifikasikan objek sesuai kategori yang telah ditentukan.

Dalam hasil ini, ditampilkan contoh-contoh prediksi dari setiap kelas, yaitu *Brown spot*, *Healthy Rice*, *Bacterial Leaf Blight*, dan *Leaf blast*. Setiap prediksi menunjukkan posisi objek pada gambar melalui *bounding box* yang diberi warna berbeda untuk tiap kelas, sehingga memudahkan identifikasi. Selain itu, nilai *confidence score* membantu menunjukkan tingkat keyakinan model terhadap prediksi yang dibuat. Penyajian visual ini diharapkan dapat memberikan

pemahaman yang lebih mudah dipahami secara visual terkait performa model, khususnya dalam membedakan objek antar kelas di kondisi nyata. Berikut merupakan hasil dari prediksi pada setiap kelas:

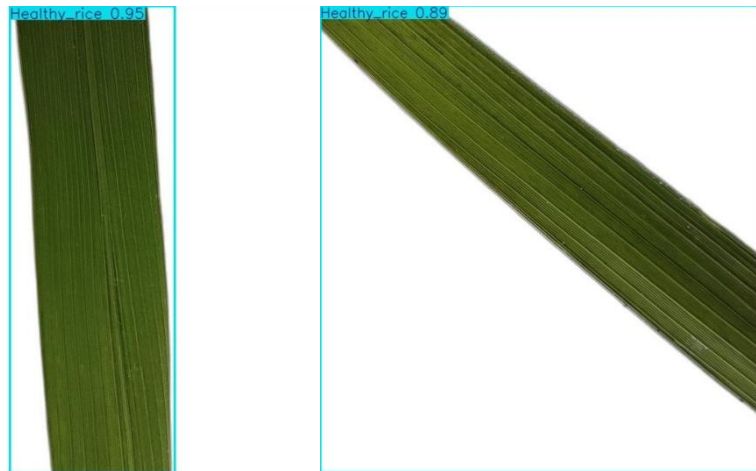
1. *Brown spot*



Gambar 4.11 Hasil prediksi pada kelas *Brown spot*

Gambar 4.11 menunjukkan hasil visualisasi prediksi model *YOLOv11* pada dataset *testing* untuk kelas *Brown spot*. Setiap daun padi yang terdeteksi ditandai dengan *bounding box* beserta nilai *confidence* dari prediksi model. Nilai *confidence* bervariasi antara 0,26 hingga 0,73, yang merepresentasikan tingkat keyakinan model terhadap keberadaan penyakit *Brown spot* pada area tertentu di daun. Kemungkinan terjadi ambiguitas atau variasi visual dari gejala. Visualisasi ini memudahkan dalam mengevaluasi sejauh mana model dapat mengenali penyakit *Brown spot* pada data baru, sekaligus memberikan wawasan tentang area yang memerlukan peningkatan kualitas dataset

## 2. Healthy Rice



Gambar 4.12 Hasil prediksi pada kelas Healthy Rice

Gambar 4.12 memperlihatkan hasil prediksi pada kelas *Healthy Rice*. *Bounding box* menandakan area daun yang diklasifikasikan sebagai sehat. Nilai *confidence* umumnya tinggi (0,89–0,95), yang menunjukkan model memiliki keyakinan kuat dalam membedakan daun sehat dari daun yang terkena penyakit. Visualisasi ini menegaskan kemampuan model dalam mengidentifikasi kondisi normal pada tanaman padi, yang penting untuk mengurangi *False Positive* pada deteksi penyakit.

## 3. Bacterial Leaf Blight



Gambar 4.13 Hasil prediksi pada kelas Bacterial Leaf Blight

Gambar 4.3 menunjukkan prediksi pada kelas *Bacterial Leaf Blight*. Area yang terdeteksi diberikan *bounding box*, dengan *confidence* yang bervariasi antara 0,26 hingga 0,85. Prediksi dengan *confidence* tinggi menunjukkan gejala yang jelas

seperti bercak memanjang di tepi daun, sementara *confidence* rendah mengindikasikan gejala yang samar atau kemiripan dengan penyakit lain.

#### 4. *Leaf blast*



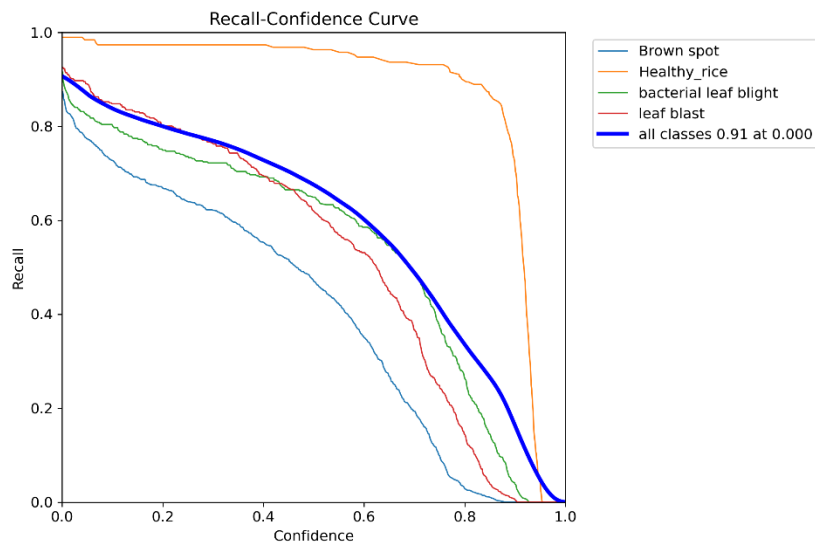
Gambar 4.14 Hasil prediksi pada kelas *Leaf blast*

Gambar 4.14 menampilkan hasil deteksi untuk kelas *Leaf blast*. *Bounding box* berwarna merah menandai area terinfeksi, dengan nilai *confidence* berkisar antara 0,5 hingga 0,89. Deteksi ini membantu memahami performa model dalam membedakan *Leaf blast* dari penyakit lain dengan gejala serupa, seperti *Brown spot*.

#### 4.7.2 Kurva dan Metrix Evaluasi

##### A. *Recall Confidence Curve*

Menunjukkan hubungan antara nilai *recall* dan tingkat keyakinan (*confidence*) prediksi model. Kurva ini membantu melihat bagaimana kemampuan model dalam mendeteksi semua objek yang benar berubah ketika ambang batas *confidence* dinaikkan.

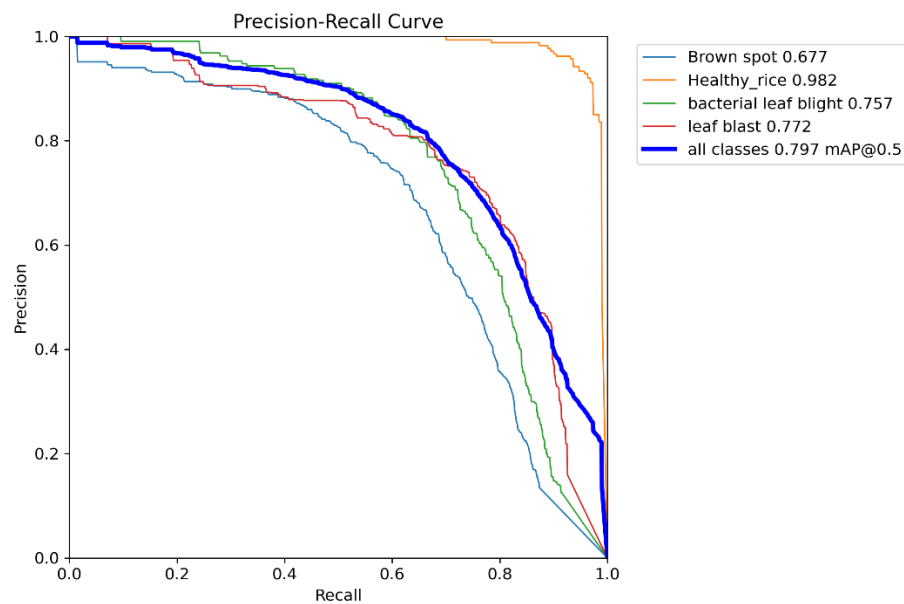


Gambar 4.15 Grafik Recall Confidence Curve

Pada *Recall-Confidence Curve* untuk dataset *testing*, terlihat bahwa kelas *Healthy\_Rice* memiliki *recall* yang tinggi dan stabil di berbagai tingkat *confidence*, menunjukkan kemampuan model yang konsisten dalam mengenali kelas ini meskipun *confidence threshold* dinaikkan. Sebaliknya, *Brown spot* menunjukkan penurunan *recall* yang cukup tajam ketika *confidence* melewati nilai 0,5, menandakan bahwa prediksi untuk kelas ini lebih sensitif terhadap perubahan *threshold*. Kelas *Bacterial Leaf Blight* dan *leaf blast* berada pada performa menengah, dengan *recall* yang cenderung stabil hingga *confidence* 0,4–0,5 sebelum mulai menurun. Secara keseluruhan, kurva ini menunjukkan bahwa model dapat mempertahankan *recall* di atas 0,8 hingga *confidence* sekitar 0,5, setelah itu terjadi penurunan yang lebih signifikan pada sebagian besar kelas.

#### B. *Precision Recall Curve*

Menggambarkan hubungan antara *precision* dan *recall* untuk berbagai ambang batas prediksi. Kurva ini berguna untuk mengevaluasi keseimbangan antara kemampuan model dalam memberikan prediksi yang benar (*precision*) dan kemampuan menemukan semua objek yang relevan (*recall*).

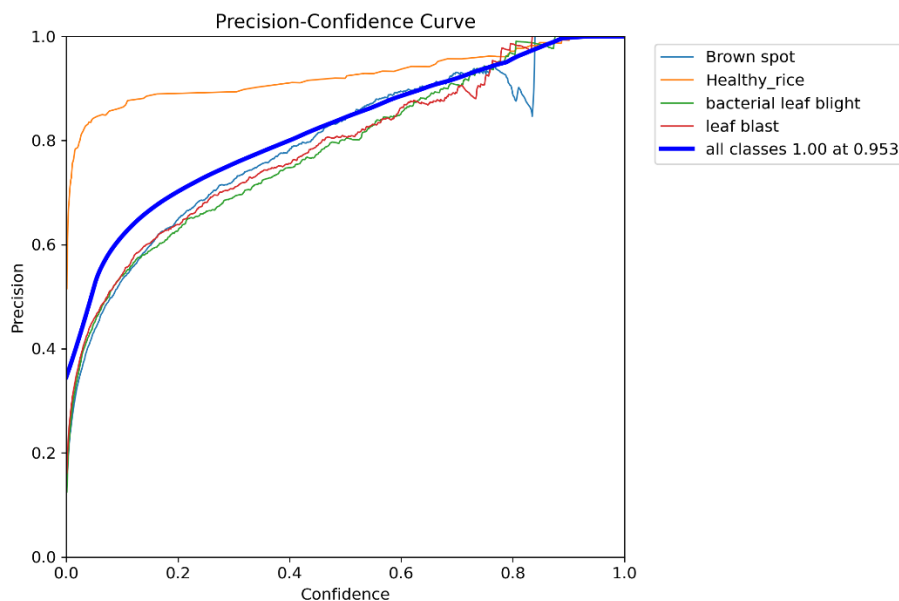


Gambar 4.16 Grafik Precision Recall Curve

Pada *Precision-Recall Curve* untuk dataset *testing*, kelas *Healthy\_Rice* menampilkan performa terbaik dengan *precision* mendekati 1 di hampir seluruh rentang *recall*, mencerminkan kemampuan model yang sangat konsisten dalam mengenali kelas ini tanpa banyak menghasilkan prediksi salah. Kelas *Leaf blast* dan *Bacterial Leaf Blight* menunjukkan keseimbangan yang cukup baik antara *precision* dan *recall*, dengan  $mAP@0.5$  masing-masing sebesar 0.772 dan 0.757. Sementara itu, kelas *Brown spot* memiliki performa paling rendah ( $mAP@0.5 = 0.677$ ) dengan penurunan *precision* yang lebih cepat ketika *recall* meningkat, menandakan tantangan model dalam menjaga akurasi prediksi pada kelas ini. Secara keseluruhan, rata-rata  $mAP@0.5$  semua kelas adalah 0.797, yang menunjukkan kinerja model yang cukup baik meskipun terdapat variasi antar kelas.

### C. Precision Confidence Curve

Menunjukkan hubungan antara nilai *precision* dan tingkat keyakinan prediksi. Grafik ini membantu melihat bagaimana ketepatan prediksi meningkat atau menurun ketika *confidence threshold* diubah.



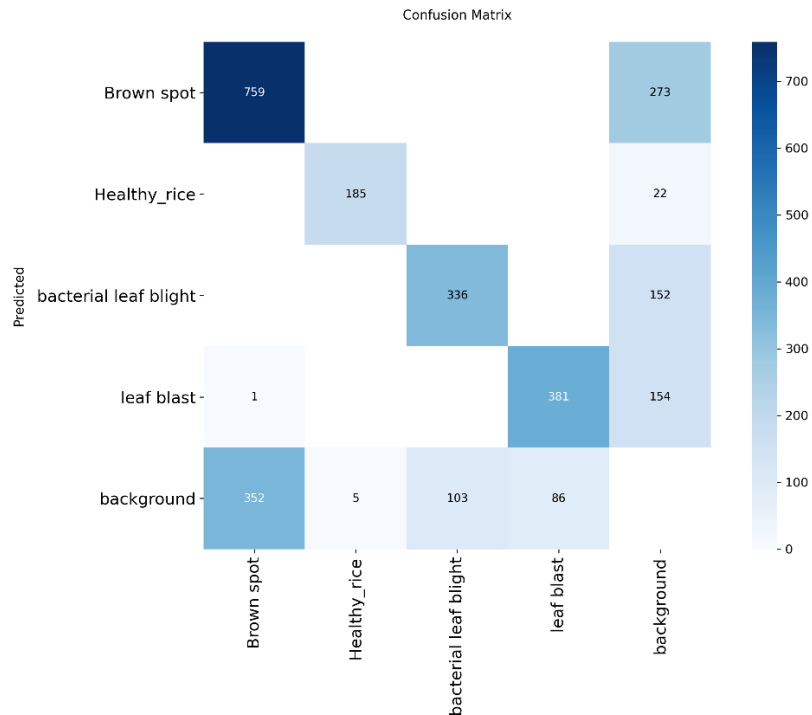
Gambar 4.17 Grafik Precision Confidence Curve

Pada *Precision–Confidence Curve* ini, terlihat bahwa peningkatan nilai *confidence* secara umum diikuti dengan kenaikan *precision* pada semua kelas. Kelas *Healthy\_Rice* kembali menunjukkan performa terbaik, mempertahankan *precision* tinggi bahkan pada *confidence* rendah dan mendekati 1 saat *confidence* mendekati maksimum. Kelas *Brown spot*, *Bacterial Leaf Blight*, dan *Leaf blast* memiliki pola peningkatan yang lebih lambat, namun tetap mencapai *precision* tinggi ketika *confidence* di atas 0.8. Garis tebal biru yang merepresentasikan rata-rata semua kelas mencapai *precision* 1.0 pada *confidence* sekitar 0.953, menandakan bahwa model sangat yakin dan akurat ketika hanya memprediksi dengan keyakinan tinggi.

#### 4.7.3 Visualisasi *Confusion matrix* Dataset Testing

grafik ini digunakan untuk menggambarkan performa model dalam mengklasifikasikan data testing dengan menampilkan jumlah prediksi benar dan salah untuk setiap kelas. Sumbu horizontal merepresentasikan kelas prediksi, sedangkan sumbu vertikal menunjukkan kelas sebenarnya. Nilai diagonal menunjukkan prediksi yang tepat, sementara nilai di luar diagonal menunjukkan kesalahan klasifikasi. Visualisasi ini memudahkan identifikasi pola kesalahan model, seperti kelas yang sering tertukar, serta membantu evaluasi akurasi dan perbaikan model di masa mendatang. Berikut akan menampilkan matrix yaitu *Confusion matrix*. Dimana Menampilkan jumlah absolut prediksi benar dan salah

untuk setiap kelas pada dataset Testing. Nilai pada diagonal menunjukkan jumlah prediksi yang benar, sedangkan nilai di luar diagonal menunjukkan kesalahan klasifikasi antar kelas yang di tunjukkan pada gambar 4.18.



Gambar 4.18 Visualisasi Confusion matrix

Gambar *confusion matrix* di atas menunjukkan performa model pada dataset testing dalam bentuk jumlah absolut prediksi. Nilai diagonal menggambarkan jumlah prediksi yang benar untuk setiap kelas, seperti *Brown spot* yang terdeteksi benar sebanyak 759 sampel dan *Healthy Rice* sebanyak 185 sampel. Sementara itu, nilai di luar diagonal menunjukkan kesalahan prediksi, misalnya 273 sampel *Brown spot* terklasifikasi sebagai *Background* dan 152 sampel *Bacterial Leaf Blight* salah dikenali sebagai *Background*. Pola ini membantu mengidentifikasi kelas yang sering tertukar dan potensi perbaikan pada model atau dataset.

## **BAB V PENUTUP**

### **5.1 Kesimpulan**

Penelitian ini menggunakan model *YOLOv11* dengan bobot model *YOLOv11s*.pt untuk mendeteksi empat kondisi daun padi: *Brown spot*, *Healthy Rice*, *Bacterial Leaf Blight*, dan *Leaf blast*. Variasi dipilih karena menawarkan keseimbangan antara akurasi dan kecepatan inferensi, berbeda dari variasi n yang lebih ringan namun kurang akurat, serta variasi m yang lebih akurat namun membutuhkan sumber daya komputasi lebih besar.

Evaluasi kinerja dilakukan menggunakan metrik mAP50, precision, recall, dan F1-score. Hasil evaluasi menunjukkan bahwa model memiliki performa sangat baik khususnya pada kelas *Healthy Rice* dengan nilai mAP50 sebesar 0,982, precision 0,901, recall 0,974, dan F1-score 0,936. Untuk kelas *Brown spot*, *Bacterial Leaf Blight*, dan *Leaf blast*, hasil deteksi juga cukup memuaskan dengan nilai mAP50 di atas 0,70, meskipun masih terdapat peluang peningkatan pada nilai recall dan F1-score.

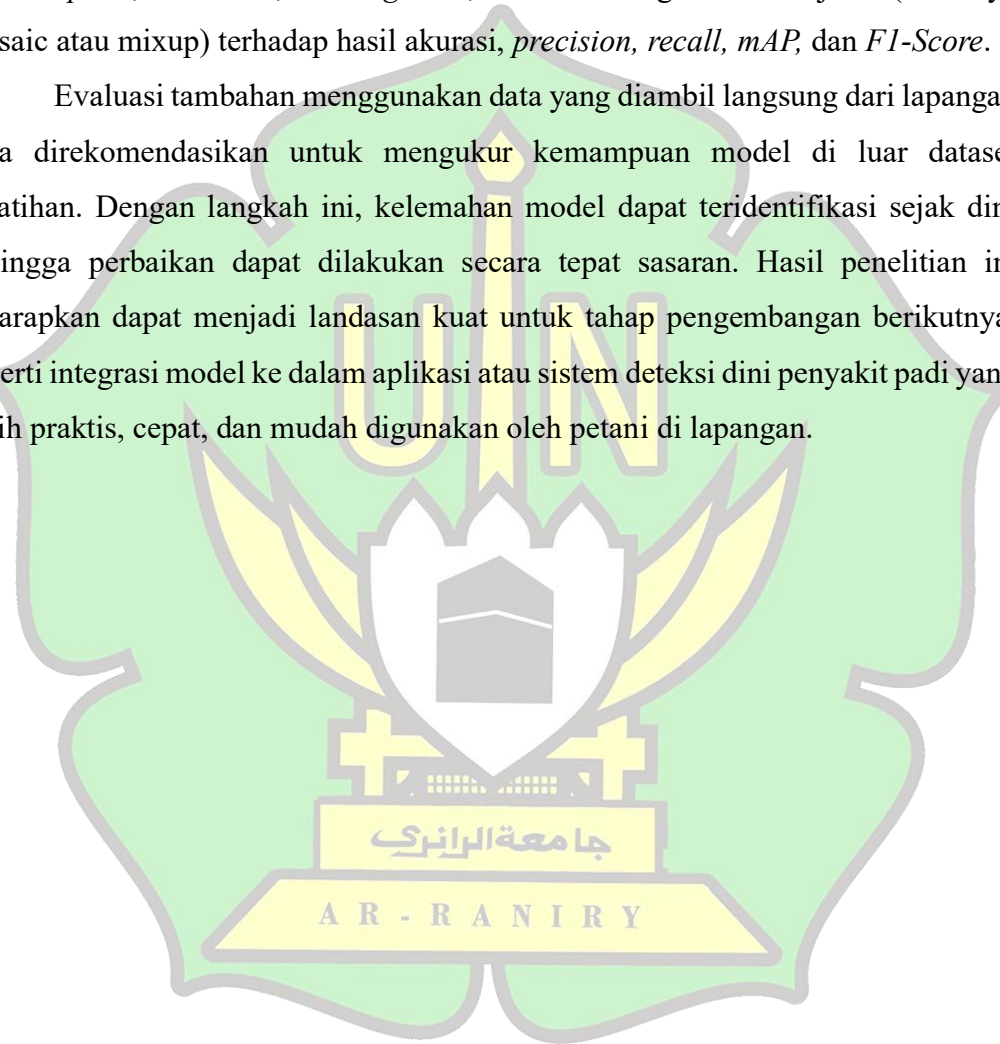
Secara keseluruhan, model *YOLOv11s* menunjukkan kemampuan klasifikasi yang baik pada penyakit daun padi. Dataset yang digunakan telah melalui proses preprocessing dan augmentasi untuk meningkatkan variasi data latih, sedangkan pelabelan bounding box mengikuti standar YOLO. Model hasil pelatihan dalam format .pt berpotensi untuk diintegrasikan ke dalam aplikasi atau perangkat lapangan sehingga dapat membantu petani mendeteksi penyakit tanaman padi secara cepat dan akurat. Dengan demikian, pendekatan Deep Learning berbasis *YOLOv11* terbukti efektif dan berpotensi dikembangkan lebih lanjut untuk mendukung pertanian presisi di Indonesia.

### **5.2 Saran**

Untuk penelitian selanjutnya, disarankan agar jumlah dan variasi dataset diperluas, terutama pada kelas penyakit dengan jumlah sampel terbatas seperti *Brown spot* dan *Leaf blast*. Pengumpulan data sebaiknya melibatkan berbagai kondisi pencahayaan, sudut pengambilan gambar, variasi ukuran objek, serta latar belakang yang beragam agar model dapat melakukan generalisasi lebih baik pada

kondisi nyata di lapangan. Penambahan data dari musim tanam yang berbeda dan lokasi pertanian yang bervariasi juga dapat meningkatkan representasi data sehingga model lebih adaptif. Selain itu, perlu dilakukan eksplorasi terhadap tipe model YOLO lainnya seperti *YOLOv11m* atau *YOLOv11l*, serta mencoba arsitektur alternatif seperti *EfficientDet* atau *Faster R-CNN* untuk membandingkan performa. Penelitian lanjutan juga dapat menguji pengaruh variasi parameter pelatihan seperti jumlah *epoch*, *batch size*, *learning Rate*, dan teknik augmentasi lanjutan (misalnya mosaic atau mixup) terhadap hasil akurasi, *precision*, *recall*, *mAP*, dan *F1-Score*.

Evaluasi tambahan menggunakan data yang diambil langsung dari lapangan juga direkomendasikan untuk mengukur kemampuan model di luar dataset pelatihan. Dengan langkah ini, kelemahan model dapat teridentifikasi sejak dini sehingga perbaikan dapat dilakukan secara tepat sasaran. Hasil penelitian ini diharapkan dapat menjadi landasan kuat untuk tahap pengembangan berikutnya, seperti integrasi model ke dalam aplikasi atau sistem deteksi dini penyakit padi yang lebih praktis, cepat, dan mudah digunakan oleh petani di lapangan.



## DAFTAR PUSTAKA

- Al Husaini, M., Rachmat Raharja, A., Cahaya Putra, V. H., & Lukmana, H. H. (2025). Enhanced Plant Disease Detection Using Computer Vision YOLOv11: Pre-Trained Neural Network Model Application. *Journal of Computer Networks, Architecture and High Performance Computing*, 7(1), 82–95. <https://doi.org/10.47709/cnahpc.v7i1.5146>
- Astriawati, F., & Anfa, Q. (2025). Isolasi dan Karakterisasi Fungi *Pyricularia oryzae* Penyebab Penyakit Blast pada Tanaman Padi. *Biospecies*, 18(1), 16–24. <https://doi.org/10.22437/biospecies.v18i1.38524>
- Deng, R., Tao, M., Xing, H., Yang, X., Liu, C., Liao, K., & Qi, L. (2021). Automatic Diagnosis of Rice Diseases Using Deep Learning. *Frontiers in Plant Science*, 12. <https://doi.org/10.3389/fpls.2021.701038>
- Fang, K., Zhou, R., Deng, N., Li, C., & Zhu, X. (2025). RLDD-YOLOv11n: Research on Rice Leaf Disease Detection Based on YOLOv11. *Agronomy*, 15(6), 1266. <https://doi.org/10.3390/agronomy15061266>
- Fitri, E., & Nugraha, S. N. (2024). OPTIMASI KINERJA LINEAR REGRESSION, RANDOM FOREST REGRESSION DAN MULTILAYER PERCEPTRON PADA PREDIKSI HASIL PANEN. *INTI Nusa Mandiri*, 18(2), 210–217. <https://doi.org/10.33480/inti.v18i2.5269>
- He, J., Ren, Y., Li, W., & Fu, W. (2025). YOLOv11-RCDWD: A New Efficient Model for Detecting Maize Leaf Diseases Based on the Improved YOLOv11. <https://doi.org/10.20944/preprints202503.1320.v1>
- Huda, M. M., Prasetyo, K. A., Vierli, M. A. R., Wulanningrum, R., & Dara, M. A. D. W. (2025, January 20). Identifikasi Mangga Berdasarkan Tingkat Kematangan Menggunakan YOLO 11. *Seminar Nasional Teknologi & Sains*. <https://proceeding.unpkediri.ac.id/index.php/stains/article/view/5773>
- Junervin. (2025). Deteksi dan Klasifikasi Otomatis Lapisan Grafena Berbasis YOLOv11. *Jurnal Tera*, 4(2), 33–45. <https://jurnal.undira.ac.id/jurnaltera/article/view/330>
- Krisdianto, K., Elta Sonalitha, & Yandhika Surya Akbar Gumilang. (2024). Deteksi penyakit padi menggunakan YOLO. *Uranus : Jurnal Ilmiah Teknik Elektro, Sains Dan Informatika*, 2(3), 125–134. <https://doi.org/10.61132/uranus.v2i3.259>
- Kukartsev, V. V., Ageev, R. A., Borodulin, A. S., Gantimurov, A. P., & Kleshko, I. I. (2024). *Deep Learning for Object Detection in Images Development and Evaluation of the YOLOv8 Model Using Ultralytics and Roboflow Libraries* (pp. 629–637). [https://doi.org/10.1007/978-3-031-70285-3\\_48](https://doi.org/10.1007/978-3-031-70285-3_48)
- Leiwakabessy, C., Inayatri, F., Jambormias, E., Patty, J., & Ririhena, R. E. (2020). Ketahanan Enam Varietas Padi Terhadap Penyakit Blas (*Pyricularia oryzae* Cav.) pada Lahan Sawah Irigasi dan Sawah Tadah Hujan. *JURNAL BUDIDAYA PERTANIAN*, 16(2), 147–156. <https://doi.org/10.30598/jbdp.2020.16.2.147>

- Li, P., Zhou, J., Sun, H., & Zeng, J. (2025). RDRM-YOLO: A High-Accuracy and Lightweight Rice Disease Detection Model for Complex Field Environments Based on Improved YOLOv5. *Agriculture*, 15(5), 479. <https://doi.org/10.3390/agriculture15050479>
- Li, Y., Chen, X., Yin, L., & Hu, Y. (2024). Deep Learning-Based Methods for Multi-Class Rice Disease Detection Using Plant Images. *Agronomy*, 14(9), 1879. <https://doi.org/10.3390/agronomy14091879>
- Mantrala, S. S. (2025). PYTHON'S PIVOTAL ROLE IN AI AND DATA SCIENCE. *INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY AND MANAGEMENT INFORMATION SYSTEMS*, 16(2), 1676–1686. [https://doi.org/10.34218/IJITMIS\\_16\\_02\\_105](https://doi.org/10.34218/IJITMIS_16_02_105)
- Maulana, M. F. A., Anggadimas, N. M., & Sani, D. A. (2025). Klasifikasi Citra Penyakit Daun Padi Dengan Metode CNN Menggunakan Arsitektur ResNet50V2. *CESS (Journal of Computer Engineering, System and Science)*, 10(2), 517–529. <https://doi.org/10.24114/cess.v10i2.66960>
- Novantara, P., Risteruw Leonardo Firmansyah, & Marrilyn Arismawati. (2025). Deteksi Hama Penyakit Daun Padi Dengan Menggunakan Teknik Optimasi Deep Learning Convolutional Neural Network. *Bit-Tech*, 7(3), 975–983. <https://doi.org/10.32877/bt.v7i3.2284>
- Pradana, A. I., Harsanto, H., & Wijiyanto, W. (2024). Deteksi Rambu Lalu Lintas Real-Time di Indonesia dengan Penerapan YOLOv11: Solusi Untuk Keamanan Berkendara. *Jurnal Algoritma*, 21(2), 145–155. <https://doi.org/10.33364/algoritma/v.21-2.2106>
- Ramos, L. T., & Sappa, A. D. (2025). A comprehensive analysis of YOLO architectures for tomato leaf disease identification. *Scientific Reports*, 15(1), 26890. <https://doi.org/10.1038/s41598-025-11064-0>
- Rippner, D. A., Raja, P. V., Earles, J. M., Momayyezi, M., Buchko, A., Duong, F. V., Forrestel, E. J., Parkinson, D. Y., Shackel, K. A., Neyhart, J. L., & McElrone, A. J. (2022). A workflow for segmenting soil and plant X-ray computed tomography images with deep learning in Google's Colaboratory. *Frontiers in Plant Science*, 13. <https://doi.org/10.3389/fpls.2022.893140>
- Sanya, D. R. A., Syed-Ab-Rahman, S. F., Jia, A., Onésime, D., Kim, K.-M., Ahohuendo, B. C., & Rohr, J. R. (2022). A review of approaches to control bacterial leaf blight in Rice. *World Journal of Microbiology and Biotechnology*, 38(7), 113. <https://doi.org/10.1007/s11274-022-03298-1>
- Sathyanarayanan, S. (2024). Confusion matrix-Based Performance Evaluation Metrics. *African Journal of Biomedical Research*, 4023–4031. <https://doi.org/10.53555/AJBR.v27i4S.4345>
- Sazak, H., & Kotan, M. (2024). Automated Blood Cell Detection and Classification in Microscopic Images Using YOLOv11 and Optimized Weights. *Diagnostics*, 15(1), 22. <https://doi.org/10.3390/diagnostics15010022>

Tian, J., Jin, Q., Wang, Y., Yang, J., Zhang, S., & Sun, D. (2024). Performance analysis of deep learning-based object detection algorithms on COCO benchmark: a comparative study. *Journal of Engineering and Applied Science*, 71(1), 76. <https://doi.org/10.1186/s44147-024-00411-z>

Witanti, W., Arie Anggara, S., & Melina, M. (2024). Peramalan Harga Cabai Rawit Merah Menggunakan Attention Mechanism Berbasis Long Short-Term Memory. *Journal of Applied Computer Science and Technology*, 5(2), 128–135. <https://doi.org/10.52158/jacost.v5i2.875>

