

**PENDETEKSI PENYAKIT PADA DAUN PADI
MENGUNAKAN ALGORITMA YOU ONLY LIVE ONCE
(YOLO) DI DESA JANGAN-JANGAN KECAMATAN
PUJANANTING KABUPATEN BARRU**

SKRIPSI

Diajukan Untuk Memenuhi Salah Satu Guna Memperoleh Gelar
Sarjana Komputer Prodi Informatika



**SUANDI ARITMAWIJAYA
105841102921**

**UNIVERSITAS MUHAMMADIYAH MAKASSAR
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA
2025**



MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
FAKULTAS TEKNIK



PENGESAHAN

Skripsi atas nama Suandi Aritmawijaya dengan nomor induk Mahasiswa 105 84 11029 21, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0004/SK-Y/55202/091004/2025, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu, 30 Agustus 2025.

Panitia Ujian :

1. Pengawas Umum

Makassar,

6 Rabi'ul Awa 1447 H

30 Agustus 2025 M

a. Rektor Universitas Muhammadiyah Makassar

Dr. Ir. H. Abd. Rakhim Nanda, ST., MT., IPU

b. Dekan Fakultas Teknik Universitas Hasanuddin

Prof. Dr. Eng. Muhammad Isran Ramli, S.T., M.T., ASEAN, Eng

2. Penguji

a. Ketua : Prof. Dr. Ir. Hafsa Nihwana, M.T.

b. Sekretaris : Desi Anggreani, S.Kom., M.T.

3. Anggota

1. Chyquitha Danuputri, S.Kom., M.Kom

2. Ir. Ida, S.Kom., M.T.

3. DARNIATI, S.Kom., M.T.

Mengetahui :

Pembimbing I

Pembimbing II

Fahrim Irhamna Rachman S.Kom., M.T.

Rizki Yushana Bakti, S.T., M.T.

Dekan



Ir. Muh. Syaifatt S. Kuba, S.T., M.T.

NEM : 795288

Gedung Menara Iqra Lantai 3

Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221

Web: <https://teknik.unismuh.ac.id/>, e-mail: teknik@unismuh.ac.id





MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
FAKULTAS TEKNIK



HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : **PENDETEKSI PENYAKIT PADA DAUN PADI MENGGUNAKAN ALGORITMA YOU ONLY LIVE ONCE (YOLO) DI DESA JANGAN-JANGAN KECAMATAN PUJANANTING KABUPATEN BARRU**

Nama : Suandi Aritmawijaya

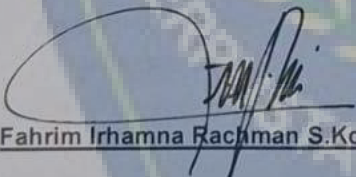
Stambuk : 105 84 11029 21

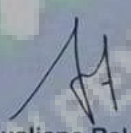
Makassar, 30 Agustus 2025

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing;

Pembimbing I

Pembimbing II


Fahrir Irhamna Rachman S.Kom., M.T


Rizki Yustiana Bakti, S.T., M.T

Mengetahui,

Ketua Prodi Informatika



ABSTRAK

SUANDI ARITMAWIJAYA, Pendeteksi Penyakit Daun Padi Menggunakan Algoritma You Only Live Once (Yolo) Di Desa Jangan-Jangan Kec, Pujananting, Kab. Barru. (Di Bimbing Oleh Fahrin Irhamna Rachman S.Kom., M.T Dan Rizki Yusliana Bakti, S. T.,M. T)

Produksi padi di Indonesia sering kali terganggu oleh serangan penyakit daun yang mengancam hasil panen dan ketahanan pangan nasional. Penelitian ini bertujuan untuk membangun sistem pendeteksi otomatis penyakit pada daun padi dengan memanfaatkan algoritma *YOLOv8 (You Only Look Once versi 8)*, sebuah metode deep learning berbasis *Convolutional Neural Network (CNN)*. Dataset dikumpulkan langsung dari lahan pertanian di Desa Jangan-Jangan, Kabupaten Barru, yang terdiri dari tiga jenis penyakit utama: blast, bercak coklat, dan hawar daun bakteri. Proses penelitian meliputi anotasi data menggunakan Roboflow, pelatihan model dengan Google Colab, dan evaluasi menggunakan metrik seperti confusion matrix, precision, recall, F1-score, dan mAP. Hasil pelatihan menunjukkan bahwa model mampu mendeteksi penyakit daun padi dengan akurasi tinggi dan waktu inferensi yang cepat, menjadikannya layak digunakan sebagai solusi real-time bagi petani dalam upaya deteksi dini dan pencegahan penyakit tanaman. Dengan demikian, sistem ini diharapkan dapat meningkatkan produktivitas pertanian dan mendukung ketahanan pangan lokal.

KATA KUNCI : YOLOv8, Deteksi Penyakit Padi, Deep Learning, Citra Digital, Pertanian Presisi, Roboflow, CNN.

ABSTRACT

SUANDI ARITMAWIJAYA, *Detecting Rice Leaf Disease Using the You Only Live Once (Yolo) Algorithm in the village of janngan-jangan Subdistrict, Pujananting, Barru Regency. (Guided by Fahrir Irhamna Rachman S.Kom., M.T and Rizki Yusliana Bakti, S. T., M. T)*

Rice production in Indonesia is often disrupted by leaf disease attacks that threaten crop yields and national food security. This research aims to build an automatic disease detection system on rice leaves using the YOLOv8 (You Only Look Once version 8) algorithm, a deep learning method based on Convolutional Neural Network (CNN). The dataset was collected directly from agricultural fields in Janggan-Janggan Village, Barru Regency, which consists of three main types of diseases: blast, brown spot, and bacterial leaf blight. The research process included data annotation using Roboflow, model training with Google Colab, and evaluation using metrics such as confusion matrix, precision, recall, F1-score, and mAP. The training results showed that the model is able to detect rice leaf diseases with high accuracy and fast inference time, making it suitable for use as a real-time solution for farmers in early detection and prevention of plant diseases. Thus, this system is expected to increase agricultural productivity and support local food security.

Keywords: YOLOv8, Rice Disease Detection, Deep Learning, Digital Imagery, Precision Farming, Roboflow, CNN.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji bagi Allah Subhanallahu Wa Ta'ala atas limpahan rahmat dan karunianya saya sebagai penulis dapat merampungkan Laporan Tugas Akhir yang berjudul **“Pendeteksi Penyakit Pada Daun Padi Menggunakan You Only Once (Yolo) Di Desa Jangan-Jangan Kecamatan Pujananting Kabupaten Barru”**. Salawat serta Salam senantiasa penulis panjatkan kepada Nabi Muhammad SAW, yang sudah membawa kita asal Zaman jahiliah menuju Zaman yang serba modern seperti waktu ini.

Skripsi ini disusun sebagai salah satu syarat untuk memenuhi tugas akademik sekaligus langkah awal dalam penelitian guna mengembangkan solusi teknologi berbasis kecerdasan buatan, khususnya dalam meningkatkan layanan pendaftaran mahasiswa baru.

Penulis menyadari bahwa terselesaikannya skripsi ini tidak lepas dari dukungan, bimbingan, dan motivasi dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Allah Subhanahu Wa Ta'ala, sumber segala ilmu dan kemudahan.
2. Kedua Orang Tua yang Penulis mengucapkan terima kasih yang sebesar-besarnya atas semua doa, kasih sayang, dan bantuan moral dan materi.
3. Bapak IR. MUHAMMAD SYAFAAT S KUBA, ST., MT, selaku Dekan Fakultas Teknik Universitas Muhammadiyah Makassar.
4. Bapak RIZKI YUSLIANA BAKTI, S.T.,M.T selaku Ketua Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar dan selaku Dosen Pembimbing 2 yang telah memberikan arahan dan bimbingan serta saran yang sangat berarti dalam penulisan skripsi ini.
5. FAHRIM IRHAMNA RACHMAN S.KOM., M.T Dosen Pembimbing 1 yang telah memberikan arahan dan bimbingan serta saran yang sangat berarti dalam penulisan skripsi ini.

6. Seluruh Dosen Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar, yang telah memberikan ilmu pengetahuan, bimbingan, dan dorongan berharga selama masa studi hingga penyusunan skripsi ini.
7. Teman-teman kelas A Angkatan 2021 Program Studi Informatika Universitas Muhammadiyah Makassar.
8. Semua pihak yang tidak dapat disebutkan satu persatu yang telah berkontribusi dalam penyelesaian skripsi ini.

Penulisan menyadari bahwa dalam penulisan skripsi ini terdapat kekurangan untuk itu penulis mengharapkan kritik dan saran. Akhir kata penulis berharap semoga skripsi ini bermanfaat bagi semua pihak demi perkembangan dan kemajuan akademik

Makassar, 12 Juni 2025

SUANDI ARITMAWIJAYA

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR.....	iii
DAFTAR TABEL	vii
DAFTAR GAMBAR.....	viii
DAFTAR LAMPIRAN	ix
DAFTAR ISTILAH.....	x
BAB 1 PENDAHULUAN.....	1
A. Latar Belakang Masalah.....	1
B. Rumusan Masalah	2
C. Tujuan Penelitian.....	2
D. Manfaat Penelitianff	2
E. Ruang lingkup Penelitian	3
F. Sistem Penulisan.....	3
BAB II TINJAUAN PUSTAKA.....	5
A. Landasan Teori.....	5
B. Penelitian Terkait.....	13
C. Kerangka Pikir.....	18
BAB III METODE PENELITIAN	19
A. Tempat Dan Waktu Penelitian	19
B. Bahan.....	19
C. Perancangan Sistem.....	20

D. Teknik Pengujian Sistem	25
E. Teknik Analisis Data	28
BAB IV HASIL DAN PEMBAHASAN	30
A. Pengumpulan dataset	30
B. Anotasi Data Di Roboflow	31
C. Labeling data di roboflow	32
D. Pembagian atau spilt data training dan testing	32
E. Proses Training	33
F. Pengujian System	44
G. Testing	46
H. Hasil test	54
BAB V PENUTUP	51
A. Kesimpulan	51
B. Saran	51
DAFTAR PUSTAKA	52
LAMPIRAN	54

DAFTAR TABEL

Table 1 Penyakit Daun Padi.....	6
Table 2 Confusion Matrix.....	26
Table 3 Jumlah Data	31
Table 4 Hasil Deteksi Penyakit Bercak Coklat.....	46
Table 5 Penyakit Hawar Daun.....	47
Table 6 Hasil Deteksi Penyakit Padi Blast	49
Table 7 Real Time Deteksi Bercak Coklat.....	51
Table 8 Real Time Penyakit Hawar Daun	52
Table 9 Real Time Penyakit Blast	53



DAFTAR GAMBAR

Gambar 1 Arsitektur Yolo	9
Gambar 2 Arsitektur YOLOv8	11
Gambar 3 Kerangka Berpikir	18
Gambar 4 Flowchart Sistem Pendeteksi Penyakit Daun Padi	20
Gambar 5 Flowchart Algoritma YOLOv8 Deteksi Penyakit Pada Daun Padi.....	24
Gambar 6 Penyakit Blast.....	30
Gambar 7 Penyakit Hawar Bakteri.....	30
Gambar 8 Bercak Coklat	30
Gambar 9 Upload Dataset	31
Gambar 10 Jenis Penyakit	32
Gambar 11 Proses Pelabelan	32
Gambar 12 Split Dataset.....	33
Gambar 13 Proses Pembuatan Api	34
Gambar 14 Hasil Export Dataset Menjadi Bentuk Api	34
Gambar 15 Proses Pemanggilan Api	35
Gambar 16 Hasil Training	35
Gambar 17 Confusion Matriks	40
Gambar 18 Folder Hasil Training.....	44
Gambar 19 Folder Python Dalam Pengujian Sistem.....	44
Gambar 20 Sourcode Pendeteksi.....	45

DAFTAR LAMPIRAN

Lampiran 1 Dataset	55
Lampiran 2 Source Code	61
Lampiran 3 Dokumentasi Pengambilan Dataset	69
Lampiran 4 Hasil Turnitin	70



DAFTAR ISTILAH

Yolo	<i>You Only Look Once</i> , algoritma deteksi objek berbasis CNN.
Yolov8	versi terbaru YOLO yang lebih cepat dan akurat dengan pendekatan anchor-free.
CNN	<i>Convolutional Neural Network</i> , jaringan saraf khusus untuk pengolahan citra.
Deep Learning	Cabang machine learning dengan jaringan saraf dalam untuk data besar.
Machine Learning	Metode kecerdasan buatan yang memungkinkan sistem belajar dari data.
Roboflow	Platform untuk anotasi, labeling, dan manajemen dataset computer vision.
Pytorch	Framework open-source populer untuk deep learning.
Confusion Matrix	Tabel evaluasi berisi nilai TP, TN, FP, FN.
Precision	Tingkat ketepatan prediksi positif.
Recall	Tingkat keberhasilan menemukan data positif.
F1-score	Rata-rata harmonis antara precision dan recall.
mAP (<i>Mean Average Precision</i>)	Metrik utama pada deteksi objek.
Bounding Box	Kotak pembatas objek hasil deteksi.
Loss Function	Fungsi pengukur kesalahan model saat training.
Supervised Learning	Pembelajaran mesin dengan data berlabel.
Dataset	Kumpulan data untuk melatih dan menguji model.
Training Set	Data latih untuk model.

Validation Set	Data validasi saat training.
Testing Set	Data uji performa model akhir.
Blast	Penyakit daun padi berupa bercak berbentuk mata ikan.
Bercak Coklat	Penyakit daun padi berupa bercak coklat tidak beraturan.
Hawar Daun	Penyakit padi dengan bercak kekuningan/kehitaman.
Google Colab	Layanan cloud Google untuk menjalankan Python/AI.
API	<i>Application Programming Interface</i> , penghubung antar sistem.
Anotasi Data	Proses memberi tanda pada citra (bounding box/label).
Labeling	Pemberian label kelas pada objek citra.
Overfitting	Kondisi model terlalu hafal data latih.
Inference	Proses model membuat prediksi pada data baru.
Backbone	Bagian YOLOv8 yang mengekstraksi fitur.
Neck	Bagian YOLOv8 yang menggabungkan fitur dari backbone.
Head	Bagian YOLOv8 yang menghasilkan prediksi akhir.

BAB 1

PENDAHULUAN

A. Latar Belakang Masalah

Indonesia merupakan salah satu negara agraris terbesar di dunia, di mana sektor pertanian memiliki peran penting dalam kehidupan sosial dan ekonomi masyarakat. Salah satu komoditas utama yang sangat vital bagi masyarakat Indonesia adalah beras. Beras tidak hanya menjadi makanan pokok mayoritas penduduk, tetapi juga menjadi sumber utama pendapatan bagi jutaan petani di seluruh negeri. Namun, produksi beras sering menghadapi tantangan serius, salah satunya adalah serangan penyakit pada tanaman padi. Penyakit-penyakit ini dapat memberikan dampak yang merusak, menurunkan hasil panen secara signifikan, dan mengancam ketahanan pangan nasional. Oleh karena itu, proses diagnosis dan identifikasi penyakit tanaman secara dini sangat penting untuk memastikan produktivitas tetap tinggi, kualitas hasil pertanian terjaga, serta efisiensi dalam pengelolaan lahan pertanian dapat tercapai. Salah satu tantangan utama dalam pengembangan pertanian saat ini adalah deteksi penyakit pada daun padi, yang jika tidak ditangani dengan tepat dapat menghambat peningkatan hasil pertanian secara keseluruhan (Widyono & Pratama, 2022).

Penyakit padi ini merupakan satu musuh utama dari ancaman ini, menghipnotis kualitas beras yang didapatkan asal beras serta bahkan membuahkan fatal. tumbuhan yang terserang penyakit membagikan tanda-tanda berupa bercak dengan pola dan warna eksklusif. oleh sebab itu, daun padi dapat dipergunakan menjadi langkah awal pada mendeteksi penyakit padi dan dapat menaikkan yang akan terjadi padi dengan menerapkan teknologi yang mendukung pertanian buat menaikkan kualitas intensifikasi (Agustiani et al., 2022).

Mesin learning, sebagai bagian dari kecerdasan buatan, merupakan metode yang meniru cara manusia berpikir untuk membangun model yang mampu menyelesaikan berbagai permasalahan secara otomatis. Salah satu pendekatan yang telah terbukti efektif dalam mendeteksi objek adalah metode YOLO (*You Only Look Once*). YOLO merupakan pengembangan dari teknik *Convolutional Neural Network* (CNN) yang dirancang untuk mengenali objek dalam gambar maupun video secara cepat dan efisien. Berbeda dari metode deteksi tradisional, YOLO bekerja dengan satu jaringan konvolusional terpadu yang mampu

secara simultan memprediksi lokasi objek dalam bentuk bounding box serta menentukan kelas objek tersebut. (Putra Pranjaya et al., 2024)

Tujuan dari penelitian ini adalah untuk mengembangkan sistem deteksi dini penyakit pada daun padi dengan memanfaatkan metode *YOLO (You Only Look Once)* sebagai salah satu pendekatan deep learning berbasis CNN, guna membantu proses identifikasi penyakit tanaman secara cepat, akurat, dan efisien. Dengan penerapan teknologi ini, diharapkan dapat meningkatkan produktivitas hasil pertanian, menjaga kualitas beras, serta mendukung ketahanan pangan nasional melalui pengelolaan lahan pertanian yang lebih efektif.

B. Rumusan Masalah

Berdasarkan latar belakang tersebut, maka rumusan masalah dalam penelitian ini adalah:

1. Bagaimana langkah-langkah implementasi algoritma YOLOv8 dalam mendeteksi berbagai jenis penyakit pada daun padi?
2. Bagaimana tingkat akurasi sistem dalam mendeteksi penyakit pada daun padi?

C. Tujuan Penelitian

Sesuai latar belakang dan rumusan masalah penelitian di atas, penelitian dapat memberikan tujuan penelitian sebagai berikut:

1. Penelitian ini bertujuan untuk mengetahui bagaimana proses implementasi algoritma YOLOv8 dalam mendeteksi penyakit pada daun padi, mulai dari tahap pengumpulan dan pelabelan data, pelatihan model, hingga pengujian dan penerapan sistem secara real-time.
2. Menganalisis tingkat akurasi sistem deteksi penyakit daun padi berbasis algoritma yolo dengan menggunakan citra daun padi dari lingkungan pertanian di Desa jang-an-jang-an sebagai data uji

D. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah untuk memberikan solusi dalam mendeteksi penyakit pada daun padi secara otomatis dengan menggunakan algoritma *You Only Look Once (YOLO)*. Penelitian ini dilakukan di Desa Jang-an-Jang-an, Kecamatan Pujananting, Kabupaten Barru, dengan harapan dapat membantu petani dalam mengidentifikasi penyakit secara lebih cepat dan akurat

1) Bagi Peneliti

- a) Memberikan pengalaman langsung dalam penerapan algoritma deep learning, khususnya YOLOv8, untuk pengolahan citra digital di bidang pertanian.
- b) Peneliti dapat mengaplikasikan ilmu yang dimiliki untuk menyelesaikan permasalahan nyata di khususnya dalam membantu petani mengenali penyakit tanaman secara dini.

2) Bagi masyarakat

- a) Petani dapat mengetahui jenis penyakit yang menyerang tanaman padi, sehingga dapat melakukan tindakan pencegahan secara cepat dan tepat.
- b) Dapat mengurangi potensi kerusakan tumbuhan padi yang berujung di yang akan terjadi panen yang lebih baik.

E. Ruang lingkup Penelitian

Penelitian ini difokuskan pada pengembangan dan implementasi sistem deteksi penyakit daun padi menggunakan algoritma YOLOv8. Adapun ruang lingkup penelitian ini mencakup:

1. Objek yang digunakan adalah citra daun padi yang diambil dari lahan pertanian.
2. Penelitian ini berfokus pada 3 jenis penyakit pada daun padi yang umum dijumpai di wilayah penelitian, seperti bercak coklat, hawar daun bakteri, dan blast.
3. Ukuran gambar 640 x 640 pixel

F. Sistem Penulisan

1. BAB I - PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian (bagi peneliti dan masyarakat), serta ruang lingkup penelitian. Tujuannya adalah untuk memberikan gambaran umum tentang fokus dan urgensi penelitian.

2. BAB II - TINJAUAN PUSTAKA

Di bab ini dibahas teori-teori yang relevan menggunakan penelitian, termasuk teori tentang penyakit daun padi, pengolahan citra digital, algoritma YOLO, serta penelitian terdahulu yang mendukung. Bab ini bertujuan memberikan dasar teori yang bertenaga buat mendukung metode dan pendekatan yang dipergunakan.

3. BAB III- METODOLOGI PENELITIAN

Bab ini menjelaskan metode yang digunakan dalam penelitian, mulai dari tahapan pengumpulan dataset, praproses data, pelatihan model YOLOv8, hingga teknik evaluasi seperti confusion matrix dan mean Average Precision (mAP). Selain itu, bab ini juga memaparkan alur kerja sistem (kerangka pikir) serta metode analisis data yang digunakan.

4. BAB IV-HASIL PEMBAHASAN

Bab ini menjelaskan tentang proses selama penelitian berlangsung dan hasil dari penelitian yang telah dilakukan.

5. BAB V-PENUTUP

Bab ini menyajikan kesimpulan dari hasil penelitian serta memberikan saran bagi peneliti selanjutnya yang ingin mengembangkan penelitian dengan judul serupa



BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. Penyakit Tanaman Padi

Tanaman padi tumbuhan pangan yang sangat penting pada Indonesia. Struktur tumbuhan padi sendiri terdiri dari akar, batang, serta daun. tumbuhan ini termasuk kedalam tumbuhan budidaya yang akibat pengolahannya dapat dijadikan bahan pangan ketika telah diolah menjadi beras. ada banyak jenis penyakit di tumbuhan padi yang menyebabkan turunnya kualitas dan kuantitas yang akan terjadi panen tumbuhan padi yang secara tidak eksklusif hal ini menghipnotis produksi beras buat kebutuhan pangan pada Indonesia. asal semua jenis tanaman padi, pada ini akan berfokuskan di tiga klasifikasi penyakit pada tanaman padi, yaitu blast, bercak coklat dan hawar daun bakteri, kemudian penyakit ini menyerang pada bagian daun tumbuhan padi. Proses klasifikasi ini akan sebagai tujuan primer di penelitian ini (Santosa et al., 2023)

Pertanian merupakan kegiatan dengan memanfaatkan sumber daya hayati yang dilakukan oleh kebanyakan manusia dengan bertujuan untuk menghasilkan suatu bahan pangan. Kegiatan seperti ini merupakan sektor industri dan juga sumber energi untuk dapat mengelola lingkungan hidup. Dalam budi daya tanaman padi tidak akan terlepas dari ancaman penyakit yang sering menyerang. Penyakit yang menyerang tanaman padi ini tanpa disadari oleh petani dan sering kali para petani kurang mengerti mengenai penyakit yang menyerang tanaman padi sehingga terlambat dalam penanganan untuk didiagnosa gejala membuat produksi tanaman padi menurun (Purnamawati et al., 2020)

Terdapat tiga jenis penyakit utama yang sering menyerang tanaman padi, yaitu blast, bercak coklat, dan hawar daun bakteri. Penyakit *blast* ditandai dengan munculnya bercak-bercak pada daun yang berwarna keputihan di bagian tengah dan berwarna coklat tua pada bagian tepinya. Bercak ini awalnya kecil, namun lama kelamaan akan membesar dan membentuk pola khas menyerupai mata ikan. Sementara itu, penyakit *bercak coklat* menyebabkan permukaan daun terlihat kering dengan bercak-bercak berwarna coklat yang menyebar secara tidak teratur di seluruh daun. Gejala ini dapat menghambat proses fotosintesis dan memperlambat

pertumbuhan tanaman. Adapun penyakit *hawar daun bakteri* memiliki gejala berupa garis-garis kehitaman atau kekuningan yang muncul di bagian tepi atau ujung daun, kemudian menyebar sepanjang batang daun. Penyakit ini dapat dengan cepat menyebar terutama saat kondisi lingkungan lembab dan basah. Ketiga penyakit ini dapat menurunkan produktivitas padi secara signifikan jika tidak ditangani dengan cepat dan tepat.

Table 1 Penyakit Daun Padi

Nama penyakit	Gambar penyakit	Ciri penyakit	Rekomendasi Penanganan
blast		Bercak berwarna keputihan dan tepi berwarna coklat tua, awalnya berupa bercak dan bentuk seperti mata ikan	Gunakan fungisida berbahan aktif triazol atau strobilurin secara tepat
Bercak coklat		Permukaan daun tampak kering, bercak menyebar tidak teratur,	Pemupukan berimbang terutama kalium dan fosfor
Hawar daun bakteri		Berupa garis-garis kehitaman atau kuning Meuncul di tepi duan atau ujung daun menyebar sepanjang batang daun	Hindari penggunaan pupuk nitrogen berlebih

2. Pengolahan Citra Digital

Pengolahan citra digital adalah ilmu yang mempelajari hal-hal berkaitan dengan perbaikan kualitas terhadap suatu gambar (meningkatkan kontras, perubahan warna, restorasi citra), transformasi gambar (translasi, rotasi transformasi, skala, geometrik), melakukan pemilihan citra ciri (*feature images*) yang optimal untuk tujuan analisis, melakukan penyimpanan data yang sebelumnya dilakukan reduksi dan kompresi, transmisi data, dan waktu proses data. Adapun diagram sederhana dari proses pengolahan citra dapat dilihat pada gambar di bawah ini.(Munantri et al., 2020)

Citra atau gambar dalam bahasa latin image artinya suatu representasi, kemiripan, atau imitasi asal suatu obyek atau benda. gambaran bisa dikelompokkan sebagai gambaran tampak dan gambaran tidak tampak. model citra tampak pada kehidupan sehari-hari: foto, gambar, serta lukisan, sedangkan citra tidak tampak contohnya: data gambar pada file (citra digital), serta citra yang direpresentasikan menjadi fungsi matematis. di antara jenis citra tadi, hanya citra digital yang bisa diolah memakai personal komputer . Jenis citra lain, Bila hendak diolah dengan komputer, wajib diubah dulu menjadi citra digital, contohnya foto di scan menggunakan scanner, persebaran panas tubuh foto ditangkap dengan kamera infra merah dan diubah menjadi info numeris, isu densitas dan komposisi bagian dalam tubuh manusia ditangkap dengan bantuan pesawat sinar-x serta sistem deteksi radiasi sebagai isu digital. aktivitas buat mengubah isu citra fisik non digital menjadi digital dianggap menjadi pencitraan. Pengolahan citra ialah sebuah bentuk pemrosesan sebuah gambaran atau gambar dengan proses numerik asal gambar tersebut, dalam hal ini yang diproses merupakan masing-masing pixel atau titik asal gambar tadi. keliru satu teknik pemrosesan gambaran memanfaatkan komputer menjadi peranti lunak memproses masing-masing pixel dari sebuah gambar(Gazali et al., 2014)

3. Deep learning

Deep learning digunakan buat mendeteksi hama dan penyakit di daun tanaman melalui objek daun dan teknik pengolahan data. Metode ini murah serta mudah buat mendeteksi sebagai akibatnya sangat berguna dan membantu para petani(Yasen et al., 2023)

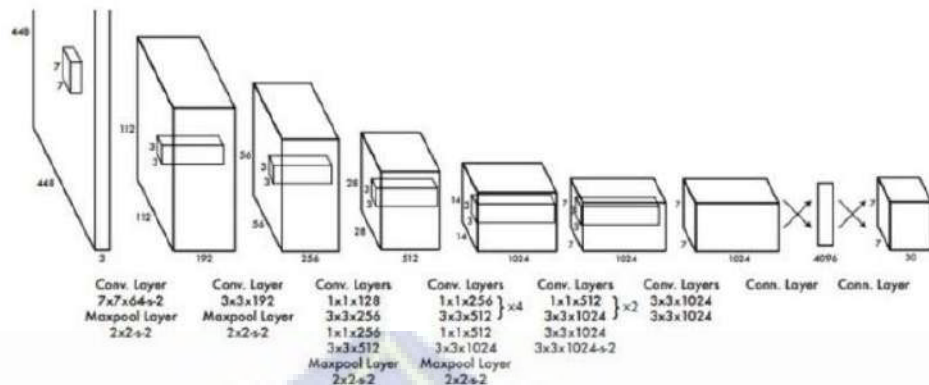
Deep learning artinya salah satu subset berasal machine learning atau pembelajaran mesin yang terinspirasi asal struktur dari otak insan. seperti halnya manusia membentuk keputusan dengan menganalisa memakai rangkaian pemikiran logis yang terstruktur, deep learning mengatasinya menggunakan sebuah prosedur pemecahan yang dinamakan neural network memungkinkan mesin untuk belajar asal data.(Baay et al., 2021)

Deep learning dapat mendeteksi penyakit, menganalisis pola warna, tekstur, dan bentuk pada daun padi, cukup beri data dan label

4. Algoritma YOLO (You Only Look Once)

You Only Look Once (YOLO) adalah sebuah metode mendeteksi objek yolo merupakan bagian dari metode *Convolution Neural Networks* (CNN) yang banyak di aplikasikan pada citra. untuk melakukan deteksi, sistem deteksinya menggunakan (*localizer* atau *classifier*). citra yang dianggap memiliki skor tertinggi dianggap sebagai hasil pilihan. metode yolo ini memisahkan gambar atau video yang dimasukkan menjadi grid berukuran $S \times S$. Jika titik tengah koordinat suatu objek jatuh dalam grid, grid itu akan mendeteksi objek tersebut. YOLO menyelesaikan semua masalah dalam satu langkah setelah setiap grid bagian memprediksi bounding box yang dapat menampung objek di dalamnya (Hastari et al., 2024)

Cara kerja YOLO adalah membagi gambar input menjadi grid $S \times S$ yang mana nilai s adalah 7 dengan input gambar 640×640 waktu ada gambar di dalam sel grid tersebut maka sel tersebut yang mempunyai tugas buat mendeteksi objek yang pada dalam sel. untuk mendapatkan nilai bounding. Masing masing sel grid memprediksi bounding box B serta nilai kepercayaan (*confidence*) dalam kotak. buat nilai prediksi kotak pembatas terdapat 5 variabel x, y, w, h dan box confidence score. buat letak x, y berisi nilai letak pusat kotak grid. seluruh atribut dalam sel grid akan dinormalisasi hingga nilai nya menjadi antara 0 sampai 1. Titik koordinat x dan y akan di sesuaikan dengan titik kiri dari atas grid yang berhubungan. Sedangkan buat tinggi serta lebar akan di sesuaikan dengan ukuran gambar (Alwie et al., 2020)



Gambar 1 Arsitektur Yolo

Fungsi Ekstraksi fitur awal dari gambar dengan filter besar (misal untuk mendeteksi tepi dan pola dasar). menggunakan gambar Dimensi 448 x 448 menggunakan tiga channel kemudian melalui suatu proses. Jaringan konvolusi didapatkan hasil bentuk (7, 7, 30), dimana 7 x 7 ialah ukuran jaringan sel ($S = 7$) serta 30 ialah nilai jumlah bujur sangkar B yang dikalikan dengan jumlah sapta-bilangan tersebut Kelas dan jumlah komponen pada kotak B ($B \times \text{lima} + C$, $B = \text{dua}$, $C = 20$) [19]. Setiap operasi konvolusi selain menjadi parameter filter berukuran kernel dan jumlah filter, terdapat pula parameter lain mensugesti format keluaran operasi Konvolusi, yaitu pad serta step. Padding artinya parameter yang memilih kuantitas penambahan border pada kurang lebih tepi gambaran input, yang meminimalkan kerugian Data pada tepian gambar (input). duduk perkara ini Dikarenakan proses pelipatan itu sendiri dimana umumnya tepi gambar diabaikan menggunakan kernel filter, kecuali kernel berukuran 1x1 Metode populer dan termudah buat mengatasi masalah ini, gunakan zero padding, yaitu. menambahkan nilai 0 buat masing-masing tepi gambar (input). Meskipun langkahnya parameter yang memilih jumlah lapisan (langkah) menjalankan kernel filter. Parameter ini sering digunakan buat perataan berukuran saluran keluar. buat menghitung berukuran output operasi konvolusi mampu dihitung menggunakan persamaan (Alwie et al., 2020)

a) Prediksi Bounding Box

YOLO membagi gambar menjadi $S \times S$ grid. Setiap grid bertanggung jawab untuk mendeteksi objek jika pusat objek berada di dalam grid tersebut.

Setiap grid memprediksi:

1) B bounding box

2) Untuk setiap bounding box:

- $(x,y)(x, y)(x,y) \rightarrow$ koordinat pusat bounding box (relatif terhadap grid)
- $w, hw, hw, h \rightarrow$ lebar dan tinggi bounding box (relatif terhadap ukuran gambar)

Confidence Score =

$$\text{Confidence} = \text{Pr}(\text{Object}) \times \text{IoUpred} \quad (1)$$

b) Confidence Score

$$\text{Confidence} = \text{Pr}(\text{Object}) \times \text{IoUpredtruth} \quad (2)$$

- $\text{Pr}(\text{Object})$ = probabilitas ada objek di bounding box
- IoUIoUIoU = Intersection over Union antara prediksi dan ground truth

c) Class Prediction

Untuk setiap grid, diprediksi juga probabilitas dari setiap kelas:

$$\text{Pr}(\text{Class}_i | \text{Object}) \quad (3)$$

d) Final Prediction

$$\text{Final Score} = \text{Pr}(\text{Class}_i | \text{Object}) \times \text{Confidence} \quad (4)$$

e) Loss Function (YOLOv8 versi sederhana) YOLO menggunakan kombinasi dari beberapa loss:

Localization Loss (Bounding Box Regression):

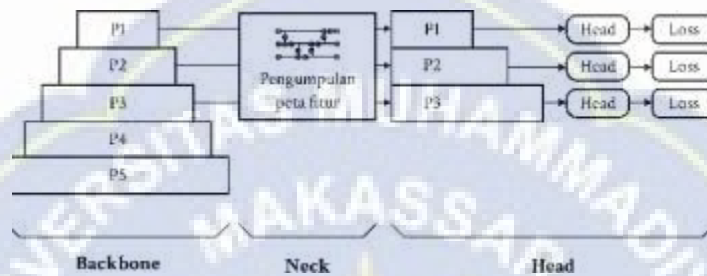
$$\text{Lcoord} = \lambda_{\text{coordi}} = 1 \sum S_2^j = 1 \sum B_{1ij} \text{obj} [(x_i - x^{\wedge}i)^2 (y_i - y^{\wedge}i)^2 + (w_i - w^{\wedge}i)^2 + (h_i - h^{\wedge}i)^2] \quad (5)$$

5. Yolov8

YOLOv8 (*You Only Look Once versi 8*) adalah versi terbaru dari algoritma deteksi objek YOLO yang sangat terkenal dalam bidang visi komputer. YOLO merupakan algoritma yang dirancang untuk mendeteksi objek dalam gambar atau video secara real-time dengan efisiensi tinggi dan akurasi yang baik. YOLOv8 adalah pengembangan dari YOLOv7 yang menghadirkan berbagai perbaikan dalam arsitektur dan kinerja.(Prayudi, 2024)

YOLOv8 memiliki keunggulan dibandingkan versi sebelumnya dalam beberapa aspek utama. Algoritma ini menggunakan arsitektur dasar yang lebih cepat dan akurat, sehingga mampu meningkatkan kinerja dalam mendeteksi objek. Salah satu inovasi penting dari YOLOv8 adalah penerapan metode deteksi objek tanpa

anchor (anchor-free), yang memungkinkan prediksi bounding box menjadi lebih presisi. Selain itu, efisiensi YOLOv8 juga didukung oleh penggunaan feature map yang lebih besar dan jaringan konvolusi yang lebih optimal, sehingga mempercepat proses pelatihan dan mengurangi risiko overfitting. YOLOv8 juga dilengkapi dengan antarmuka pengguna yang ramah, sehingga mudah diintegrasikan ke dalam berbagai perangkat lunak, menjadikannya pilihan unggulan dalam berbagai tugas deteksi objek.(Manurung et al., 2024)



Gambar 2 Arsitektur YOLOv8

a) Back bone

Fungsi backbone pada YOLOv8 adalah untuk meningkatkan efisiensi version dengan mengurangi jumlah parameter dan kompleksitas komputasi, sambil tetap mempertahankan atau meningkatkan akurasi

b) Neck

fungsinya ialah untuk memproses serta mengintegrasikan fitur yang dihasilkan oleh backbone sebelum diteruskan ke head buat prediksi akhir.

c) Head

Desain serta fungsi head ini berkontribusi pada performa keseluruhan YOLOv8 dalam aneka macam perangkat lunak global nyata, mulai berasal pengawasan sampai sistem tunggangan otonom.

Berikut beberapa keunggulan metode YOLOv8:

- 1) YOLOv8 memiliki performa yang lebih baik dibandingkan versi sebelumnya, dengan deteksi objek yang lebih cepat dan akurat.
- 2) Model YOLOv8 memiliki ukuran yang lebih kecil dibandingkan versi sebelumnya, sehingga lebih mudah untuk digunakan dan diimplementasikan.

- 3) YOLOv8 menggunakan metode ekstraksi fitur yang lebih baik dibandingkan versi sebelumnya, sehingga menghasilkan deteksi objek yang lebih akurat.
- 4) YOLOv8 memiliki kemampuan multi-skala, sehingga dapat menangani objek dengan ukuran berbeda pada gambar yang sama.
- 5) YOLOv8 dapat melakukan deteksi banyak objek pada gambar yang sama, mempermudah proses deteksi objek.

YOLOv8 dapat melakukan deteksi objek pada gambar berukuran besar dengan akurasi yang baik, mempermudah proses deteksi objek. Dengan keunggulan-keunggulan ini, YOLOv8 menjadi metode deteksi objek yang sangat populer dan banyak digunakan. Ini membantu mempermudah proses deteksi objek dan memastikan hasil yang akurat dan tepat.

6. Roboflow

Roboflow ialah *framework* pengembang computer vision buat pengumpulan data yang lebih baik ke pra pemrosesan, serta teknik pembinaan contoh. Roboflow mempunyai kumpulan data publik yang tersedia bagi pengguna dan juga memiliki akses bagi pengguna buat mengunggah data khusus mereka sendiri (Pokhrel, 2024).

Roboflow berfungsi untuk mengelola dataset gambar, memberi label, melatih model dan mendeteksi penyakit secara otomatis. Cara kerja roboflow ialah mengkonversi dataset yang sudah pada beri label ke pada YOLOv8.

7. Pytorch

PyTorch merupakan sebuah library yang dapat digunakan dalam machine learning deep learning, dan bahkan natural language processing yang dikembangkan oleh Facebook AI (sekarang Meta AI) pada tahun 2016. *PyTorch* pada dasarnya adalah compiled library /shared object yakni sebuah file yang isinya adalah fungsi-fungsi yang dapat dieksekusi pada masing-masing sistem operasi atau secara native. Format file berbeda-beda pada tiap sistem operasi, yaitu .dll pada windows dan .so pada linux. *PyTorch* dikembangkan menggunakan c++, namun PyTorch memiliki frontend API berbasis python yang sangat mempermudah penggunaan library tersebut (Qodryantha et al., 2023).

Pytorch adalah sebuah framework atau Pustaka pemrograman yang digunakan untuk membuat dan melatih model kecerdasan buatan, khususnya model deep learning

pembelajaran mendalam. pyTroch banyak digunakan oleh peneliti, praktisi AI, dan pengembangan karena fleksibel dan mudah di gunakan. pyTorch membuat model AI, mengelola data dalam bentuk tensor, melatih model (training), mendukung GPU dan flexible dan dinamis. PyTorch berfungsi sebagai kerangka kerja utama untuk mengembangkan sistem deteksi penyakit daun padi berbasis visi komputer dan deep learning. Ia memudahkan semua tahap, dari pembangunan model hingga penggunaannya untuk prediksi.

8. Confusion Matrix

Confusion Matrix merupakan alat pengukuran kinerja metode prediksi menggunakan menghitung taraf kebenaran proses penjabaran. akibat pengujian ini berupa perbandingan akurasi prediksi kemunculan hari baik keluaran ketiga metode fuzzy yang diimplementasikan pada software pengujian. Akurasi prediksi kemunculan hari baik dihitung memakai Confusion Matrix. Jumlah data prediksi dari sistem dan jumlah data prediksi asal ahli sebagai variabel masukan Confusion Matrix, selanjutnya oleh sistem memberikan masing-masing metode fuzzy nilai Accuracy, Precision, Recall dan F-1 Score (Hary Candana et al., 2021)

9. Supervised Learning

Supervised Machine Learning adalah salah satu metode machine learning yang memakai donasi training dataset asal user sebagai akibatnya mesin secara otomatis bisa mengidentifikasi sesuai pelatihan dataset buat melakukan prediksi maupun penjabaran (Aisyah et al., 2021)

Supervised learning berfungsi buat menciptakan prediktif dengan menggunakan data yang telah pada diberi label, sehingga contoh tersebut dapat di gunakan buat mengklasifikasikan data baru atau memprediksi nilai berdasarkan pola yang telah pada pelajari dari data latih.

B. Penelitian Terkait

Sebelumnya terdapat beberapa penelitian sebagai referensi sebagai referensi untuk pendeteksi menggunakan algoritma yolov8. beberapa penelitian yang terkait yaitu:

1. "Deteksi Dan Klasifikasi Hama Potato Beetle Pada Tanaman Kentang Menggunakan Yolov8" Daniel Geoffery Manurung, Mohammad Ryan Pinasthika, Muhammad Azka

Obila Vasya,Rania Aprilian Dwi Setya Putri,Agustina Parasian Tampubolon, Rakan Fadhil Prayata,Septia Khoirin Nisa,Novanta Yudistira (2024)

Dipengujian contoh YOLOv8-s, dilakukan pembinaan menggunakan 10 epoch serta total parameter mencapai 11.126.358 parameter. Secara holistik, model mencapai recall 83,3% serta precision 75,4%, yang mengidentifikasi kemampuan model dalam mendeteksi objek menggunakan akurasi yang baik. Nilai mean average precision buat mAP50 secara holistik 81,8% dan mAP50-95 38,9%. kemudian, penilaian kelas menggunakan objek guk mencapai precision 78,1%, recall 89,8%, mAP50 88,7%, mAP50-95 48%. buat kelas lich mencapai precision 72,8%, recall 76,8%, mAP50 74,9%, mAP50-95 29,8%. hasil pengujian model YOLOv8- s dapat digambarkan dengan confusion matrix di Terakhir di pengujian model YOLOv8-m, dilakukan pelatihan menggunakan 10 epoch dan total parameter mencapai 25.840.918 parameter. Secara keseluruhan, contoh mencapai recall 76,1% serta precision 88,2%, yang mengidentifikasi kemampuan contoh pada mendeteksi objek menggunakan akurasi yang baik. Nilai mean average precision buat mAP50 secara keseluruhan 82,lima% dan mAP50-95 39,7%.

2. Aplikasi Deteksi Ham Dan Penyakit Pada Buah Kakao Menggunakan Yolo V8 Berbasis Website” Alhita Dwi Lestari (2024)

Pada pengujian yang didapat dari Black Box Testing mendapatkan akibat 100% keberhasilan, sedangkan pengujian yang gagal menerima 0% dari tiga responden serta 12 pengujian dilengkapi menggunakan masukan asal responden yang tercantum pada lampiran. Berikutnya buat pengujian White Box menerima hasil yang sangat memuaskan dimana mendapatkan presentase 100% dengan pengujian yang tercapai, sedangkan buat pengujian gagal mendapatkan presentase 0% asal value test yang terdiri dari 4 path. Selain itu, ada pengujian User Acceptance Testing (UAT) membentuk taraf keberhasilan yang sangat baik, dengan persentase sebesar 90,8% asal lima responden yang menjawab 10 pertanyaan. Terakhir adapula pengujian lapangan yang menyampaikan hasil konklusi bahwa software dapat mendeteksi hama serta penyakit pada buah kakao menggunakan baik serta dapat bermanfaat bagi pengguna, khususnya petani.

3. "Aplikasi Deteksi Penyakit Pada Daun Tanaman Singkong Menggunakan Algoritma You Only Look Once (Yolo) V8 Berbasis Android" Tegar Ramadan (2024)

Sesuai pengujian Black Box, dari 7 pengujian di perangkat lunak yang didapat berasal tiga responden, berikut ini akibat pengujian Black Box: • Pengujian Pertama Tercapai: $7/7 \times 100\% = 100\%$ Gagal: $0/7 \times 100\% = 0\%$ • Pengujian kedua Tercapai: $7/7 \times 100\% = 100\%$ Gagal: $0/7 \times 100\% = 0\%$ • Pengujian Ketiga Tercapai: $7/7 \times 100\% = 100\%$ Gagal: $0/7 \times 100\% = 0\%$ Jumlah presentase homogen-homogen tercapai = $300\% / 3 = 100\%$ Jumlah presentase rata-rata gagal = $0\% / 3 = 0\%$ sesuai analisis tadi, berasal 7 pengujian yang dilakukan oleh tiga responden, hasilnya membagikan bahwa taraf keberhasilan pengujian Black Box mencapai 100%. Ad interim kegagalan tak terjadi sama sekali, sebagai akibatnya persentasenya artinya 0%. Kesimpulannya, aplikasi berjalan sinkron menggunakan fungsinya yang dibutuhkan.

4. "Klasifikasi Jenis Buah Pepaya Menggunakan YOLOv8" Egi Verdiansyah, Firman Nurdiyansyah, Istiadi (2024)

Berdasarkan uji coba yang telah dilakukan, klasifikasi buah pepaya ditunjukkan pada Gambar 3 dapat dijelaskan bahwa pepaya California memiliki hasil nilai sebesar 97%. Kemudian pepaya Bangkok memiliki hasil klasifikasi sebesar 96%. Buah pepaya Hawaii memiliki nilai 95%, dan buah pepaya Red Lady memiliki nilai 95%. Hasil keakuratan pada klasifikasi jenis buah pepaya menggunakan YOLOv8 cukup baik. Dalam melakukan klasifikasi dibutuhkan waktu yang relatif cepat, kecepatan: 0.0ms preprocess, 0.7ms inference, 0.0ms postprocess per image at shape. Untuk menerapkan ke dalam aplikasi nyata, inferensi kecepatan merupakan faktor penting dalam melakukan klasifikasi objek. Proses pelatihan dan pengujian dari klasifikasi jenis buah pepaya menggunakan YOLOv8, menggunakan kumpulan data berukuran 128 x 128 piksel yang diunggah ke Google Colab dan jenis runtime Python 3 dan GPU. Untuk pembagian data latihan sebesar 88% (1050 image), data validasi 8% (100 image), dan data uji 4% (50 image). Dataset dipisah pada setiap kelas buah pepaya. Pelatihan data dilakukan dengan menggunakan 300 epochs. Pada Tabel 3 adalah hasil percobaan terbaik dari model YOLOv8 untuk mengklasifikasi buah pepaya menggunakan 300 epochs.

5. “Implementasi Algoritma Deep Learning Yolo (You Only Look Once) Untuk Deteksi Kualitas Kentang Segar Dan Busuk Secara Real Time” Ocha Alfiana, Dan Santi Rahayu (2024)

Hasil pengujian model YOLOv8n menggunakan dataset yang terdiri dari 58 sampel segar dan 58 sampel busuk. Hasil pengujian menunjukkan bahwa model dapat mendeteksi kedua jenis kualitas dengan akurat, menegaskan efektivitasnya dalam mengidentifikasi objek pada data tes. Meskipun model ini menunjukkan tingkat kepercayaan yang tinggi dalam prediksi, ada beberapa gambar atau video, terutama dari Google, yang tidak terdeteksi dengan baik.

6. ”Pendeteksi Penyakit Pada Daun Cengkeh Menggunakan Algoritma You Only Live Once (Yolo)” Muh Ilham Kurniawan (2024)

Penelitian ini menunjukkan bahwa algoritma YOLOv8 mampu mendeteksi penyakit pada daun cengkeh dengan sangat baik. Model berhasil membedakan antara daun sehat, daun cacar, dan daun bukan cengkeh. Hasil pelatihan model mencapai akurasi 99,3%, sedangkan pengujian menggunakan confusion matrix menunjukkan akurasi 94%. Dalam uji coba menggunakan 10 gambar sampel baru, sistem mampu mendeteksi seluruhnya dengan tepat, dengan tingkat akurasi per kelas: cacar cengkeh 0,94, daun sehat 0,88, dan bukan cengkeh 0,75. Hasil ini membuktikan bahwa YOLOv8 efektif dalam mengenali gejala penyakit secara otomatis, dengan catatan bahwa kualitas dan jumlah data sangat memengaruhi hasil akurasi

7. “Real-time Identification of Rice Leaf Diseases using Convolutional Neural Networks” Juan Arcila-Diaz¹, Dilmer Altamirano-Chavez¹, Liliana Arcila-Diaz¹, Carlos Valdivia² (2024)

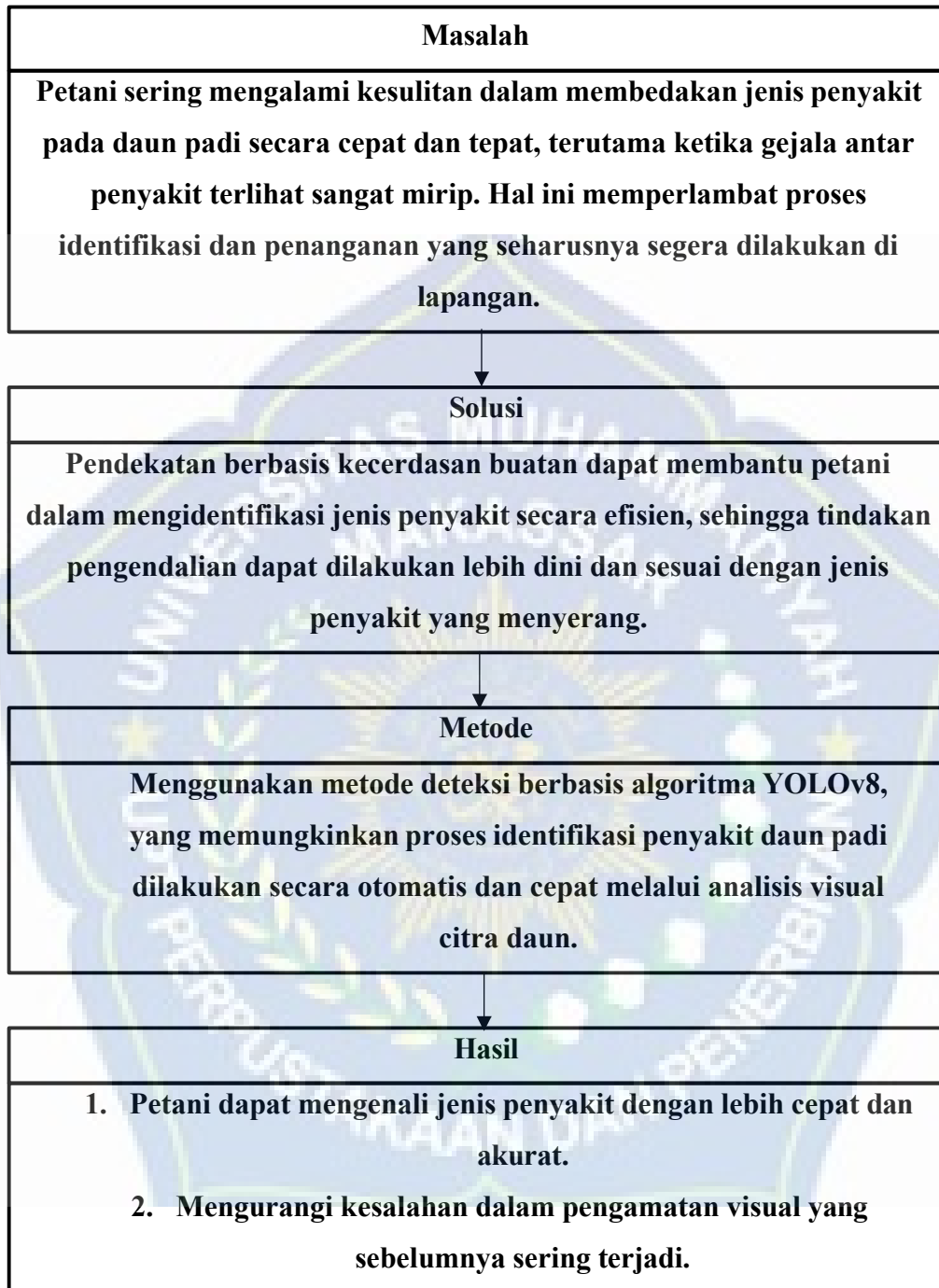
Berdasarkan hasil penelitian, model YOLO versi 8 berhasil mendeteksi dua jenis penyakit daun padi, yaitu *Brown Spot* dan *Blast*, dengan tingkat akurasi sebesar 92,98%, recall 92,45%, dan F1 Score 92,71%. Nilai precision tertinggi dicapai pada deteksi *Brown Spot*, yaitu 98,3%, sedangkan untuk *Blast* sebesar 92,0%, dengan rata-rata deteksi mencapai 95,2%. Selain akurat, model ini juga sangat efisien karena hanya membutuhkan waktu 26 milidetik untuk inferensi satu gambar, menjadikannya sangat cocok digunakan dalam sistem deteksi penyakit secara real-time.

8. "Detecting Rice Diseases Using Improved Lightweight YOLOv8n" Zhen Liu, Weixin Zhang, Xinyu Zhang, Lianwen Jin, dan Xianglong Liu (2025)

Penelitian mengembangkan model YOLOv8-DiDL untuk mendeteksi penyakit pada daun padi dengan mengintegrasikan DCNv2, DySample, dan LSK-Attention guna meningkatkan akurasi dan efisiensi. Uji coba terhadap gambar daun padi yang mengalami penyakit seperti blast, bacterial leaf blight, dan brown spot menunjukkan bahwa model ini mencapai mAP@0.5 sebesar 90,8%, dengan ukuran model hanya 7,5 MB dan kecepatan inferensi lebih dari 100 FPS, menjadikannya sangat ideal untuk aplikasi real-time di lapangan.



C. Kerangka Pikir



Gambar 3 Kerangka Berpikir

BAB III

METODE PENELITIAN

A. Tempat Dan Waktu Penelitian

Di Desa Jangan-Jangan,Kec.Pujananting,Kab.Barru Sulawesi Selatan dan waktu pelaksanaan di bulan juni sampai agustus 2025

B. Bahan

1. Kebutuhan Hardware:

a) Leptop Asus tuf gaming A15 dengan spesifikasi:

- Sistem operasinya menggunakan Windows 11 Home Single Language 64-bit
- BIOS yang terpasang adalah FA506IHRB.307
- Prosesor yang digunakan adalah AMD Ryzen 5 4600H with Radeon Graphics
- RAM sebesar 8 GB

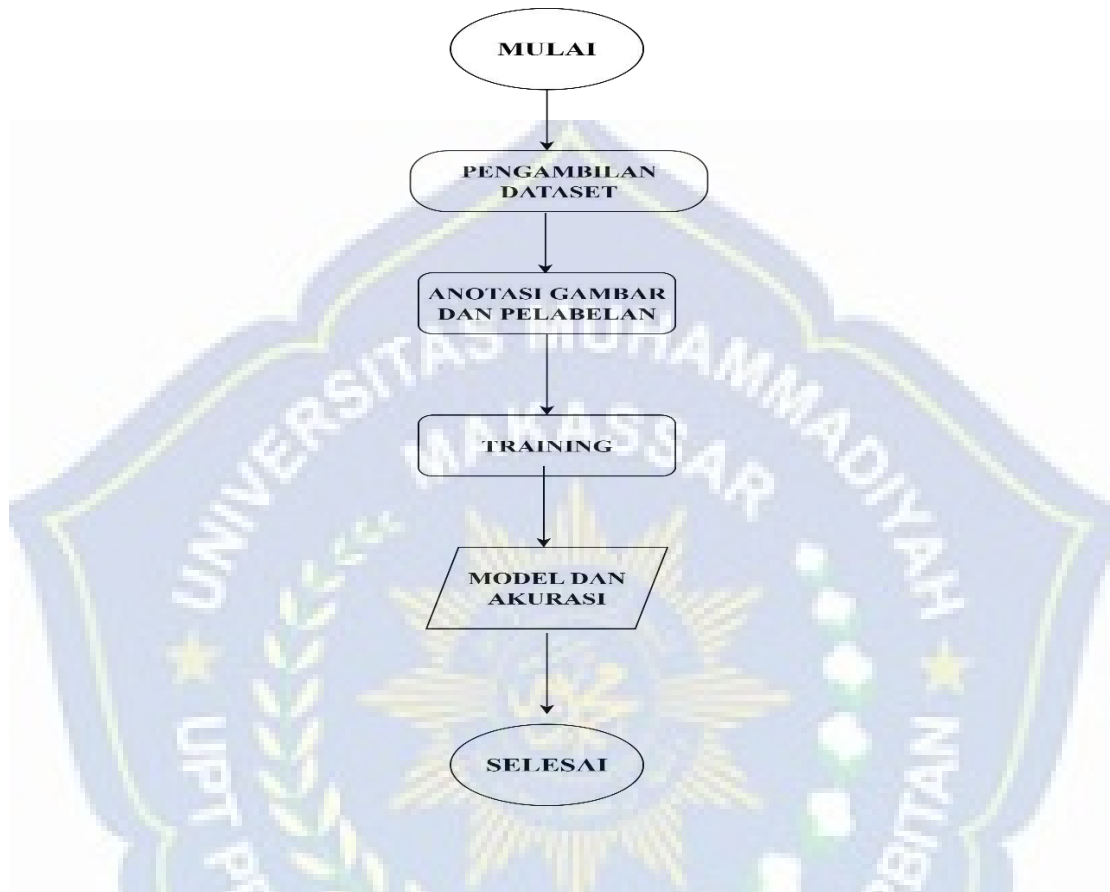
2. Kebutuhan Software:

- a) Vs code
- b) pytorch
- c) Reboflow (untuk antonasi gambar)

C. Perancangan Sistem

Adapun flowchart sistem pendeteksi penyakit daun padi menggunakan algoritma yolo yaitu:

1. Flowchart sistem pendeteksi penyakit daun padi



Gambar 4 Flowchart Sistem Pendeteksi Penyakit Daun Padi

Penjelasan tentang gambar:

a) **Mulai**

Mulainya pengumpulan data menjadi langkah pertama pada rangkaian aktivitas pembangunan sistem deteksi penyakit di daun padi. Pengumpulan data ini sebagai fondasi penting sebab kualitas serta keberagaman data yang diperoleh akan sangat memengaruhi yang akan terjadi akhir dari sistem yang dikembangkan. sehabis data terkumpul, proses akan dilanjutkan ke termin-termin berikutnya, mirip anotasi gambar, pembinaan model, serta evaluasi performa sistem. menggunakan

demikian, dimulainya pengumpulan data ini pula menandai dimulainya pelaksanaan holistik proyek secara menyeluruh dan sistematis.

b) Pengambilan dataset

Pada tahap ini, dilakukan proses pengumpulan data berupa gambar daun padi yang akan digunakan sebagai dataset buat pembinaan contoh deteksi penyakit. Pengambilan gambar dilakukan secara pribadi pada lapangan memakai kamera handphone, yang dipilih sebab praktis serta praktis diakses sang siapa saja, termasuk petani. Gambar-gambar yang dikumpulkan wajib mewakili banyak sekali kondisi tanaman, tidak hanya daun yang terinfeksi penyakit, namun juga daun pada kondisi sehat menjadi pembanding. Selain itu, dataset harus meliputi majemuk jenis penyakit yang umum menyerang tanaman padi, mirip blast, bercak coklat, serta hawar daun, agar model yang dilatih nantinya bisa mengenali berbagai variasi tanda-tanda. buat memastikan kualitas data, gambar sebaiknya diambil berasal aneka macam sudut, pencahayaan, serta latar belakang supaya contoh dapat belajar secara lebih menyeluruh serta mampu mengikuti keadaan terhadap syarat nyata di lapangan.

c) Anotasi Gambar dan Pelabelan

Anotasi adalah proses penting dalam pengolahan data visual yang melibatkan penandaan objek dalam gambar dengan bounding box dan pemberian label sesuai karakteristik objek. Dalam konteks deteksi penyakit daun padi, anotasi dilakukan untuk menandai area daun yang menunjukkan gejala penyakit seperti bercak, perubahan warna, atau kerusakan jaringan. Setiap area yang ditandai diberi label nama penyakit seperti 'blast', 'hawar daun', atau 'bercak coklat'. Proses ini menggunakan tools seperti Roboflow atau LabelImg untuk mempermudah pelabelan secara manual namun efisien. Kualitas anotasi yang teliti dan konsisten sangat berpengaruh terhadap akurasi model deteksi YOLOv8 yang akan dilatih.

a) Pembagian dataset

Dataset dibagi menjadi 3 bagian yaitu:

- **Training Set (buat melatih)**

Training set artinya bagian terbesar dari dataset dan digunakan untuk melatih contoh supaya dapat mengenali pola-pola yang terdapat pada data. di tahap ini, contoh akan belajar berasal gambar serta label yang tersedia, melakukan penyesuaian terhadap bobot serta parameter internalnya agar bisa memprediksi output yang benar. Semakin poly dan berkualitas data training, semakin baik kemampuan contoh dalam mengenali pola.

- **Validation set (buat memvalidasi kinerja selama pelatihan)**

Validation set digunakan selama proses pelatihan buat mengevaluasi kinerja contoh secara terpola. Data ini tak digunakan buat melatih contoh, namun buat mengukur seberapa baik contoh bisa melakukan prediksi terhadap data yang belum pernah dicermati sebelumnya. Hal ini membantu pada mencegah overfitting, yaitu ketika model terlalu menghafal data training serta gagal menggeneralisasi ke data baru.

- **Testing set (buat menguji akurasi contoh sesudah dilatih)**

Testing set ialah kumpulan data yang sah-h Sah diluar proses pelatihan juga validasi. Dataset ini digunakan hanya sekali, yaitu selesainya model terselesaikan dilatih, buat mengukur performa akhir contoh. Testing set memberikan gambaran seberapa baik model bisa bekerja pada kondisi nyata, karena data ini tidak pernah dilihat atau dipengaruhi selama proses training.

b) Training (Pelatihan Model)

Contoh YOLOv8 dilatih memakai data pembinaan yang terdiri berasal deretan gambar daun padi yang sudah diberi anotasi atau label sinkron menggunakan jenis penyakitnya. pada proses pelatihan ini, model secara bertahap akan menelaah dan mengenali pola-pola visual spesial yang muncul di setiap kategori penyakit, mirip perubahan warna, bentuk bercak, atau kerusakan jaringan daun. dengan memanfaatkan teknik deep learning, contoh bisa mengekstraksi fitur-fitur krusial berasal gambar buat

membedakan satu jenis penyakit menggunakan yang lainnya. Semakin banyak serta bervariasi data yang digunakan, maka semakin baik juga kemampuan contoh dalam melakukan deteksi dan klasifikasi secara akurat di data baru yang belum pernah dilihat sebelumnya.

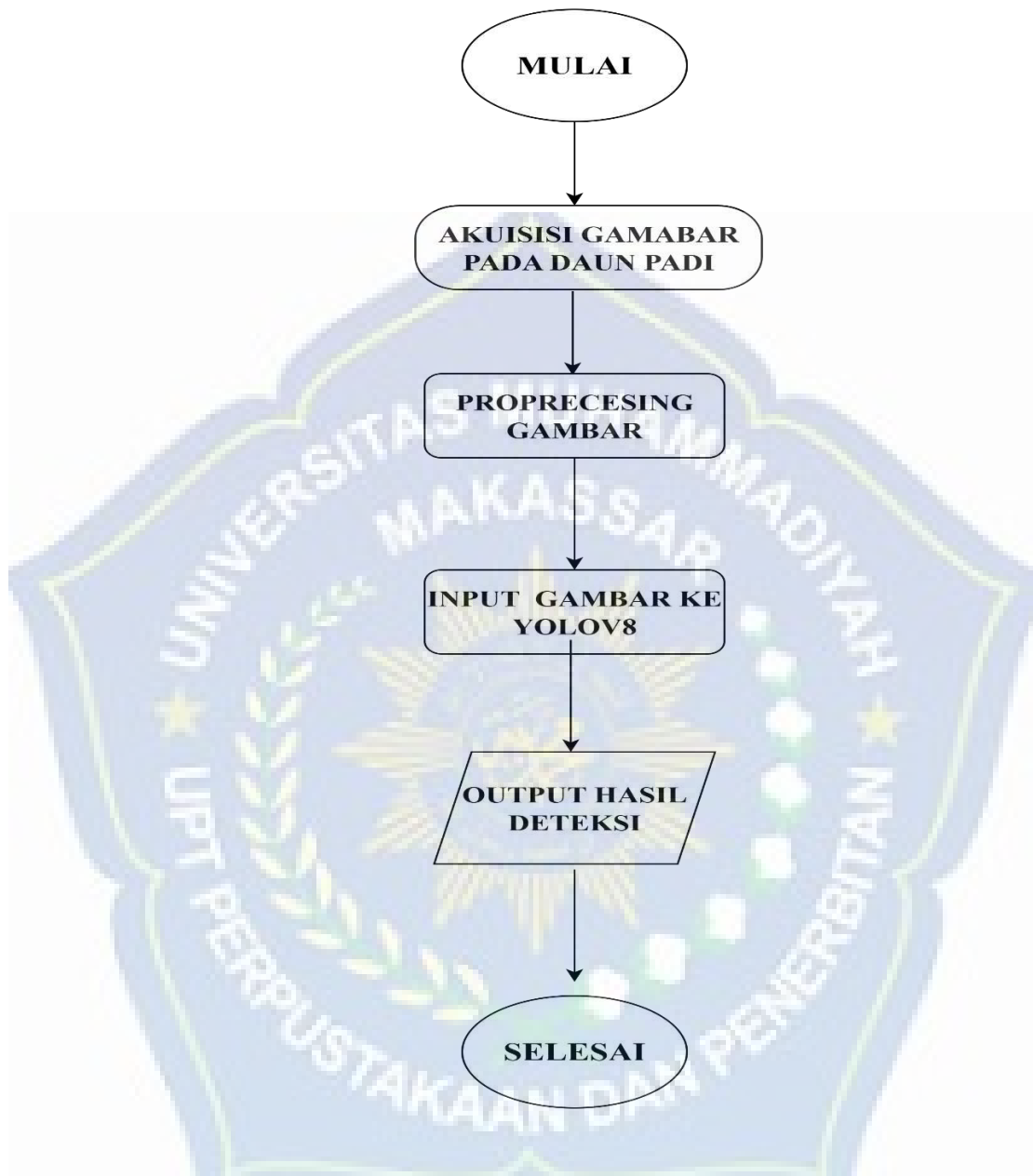
c) Model dan Akurasi

Pada tahap model dan akurasi, proses pelatihan menghasilkan sebuah model yang mampu mendeteksi dan mengklasifikasikan penyakit pada daun padi berdasarkan pola visual yang telah dipelajari dari data yang dianotasi sebelumnya. Setelah model terbentuk, dilakukan evaluasi untuk mengukur tingkat ketepatan atau akurasi dari model tersebut dalam mengenali jenis penyakit. Akurasi ini menunjukkan seberapa sesuai hasil deteksi model dengan kondisi sebenarnya. Semakin tinggi akurasi yang dicapai, maka semakin baik pula kemampuan model dalam mengenali gejala penyakit secara tepat.

d) Selesai

Kalimat ini menandakan berakhirnya semua rangkaian proses pendeteksian yang telah dilakukan. dengan demikian, sistem sekarang sudah siap sepenuhnya buat dipergunakan oleh para petani di lapangan menjadi alat bantu pada mengidentifikasi banyak sekali jenis penyakit pada daun padi secara lebih cepat, seksama, dan efisien.

2. *Flowchart* algoritma YOLO penyakit pada daun padi



Gambar 5 Flowchart Algoritma YOLOv8 Deteksi Penyakit Pada Daun Padi

Penjelasan tentang gambar di atas:

- Akuisisi gambar adalah tahapan awal dalam sistem deteksi penyakit dimana gambar atau citra daun padi dikumpulkan sebagai data input data
- Preprocessing adalah proses persiapan dan penyesuaian hasil akuisisi agar dapat di proses secara optimal oleh model deteksi seperti yolov8.tahap ini

sangat penting untuk memastikan akurasi deteksi tinggi serta konsistensi yang sesuai dengan arsitektur model tujuan nya mengurangi noise atau gangguan visual yang bisa mempengaruhi kinerja model

- c) YOLOv8 adalah model deep learning yang di rancang untuk melakukan deteksi objek dengan akurasi tinggi dan kecepatan tinggi.yolov8 digunakan untuk mendeteksi gejala-gejala penyakit daun padi seperti:blast,hawar daun bakteri dan bercak coklat pada daun padi
- d) Sistem ini akan menghasilkan output deteksi yaitu informasi yang menunjukkan jenis penyakit daun padi yang terdeteksi adapun komponen utama output deteksi:
 - Bounding box (kotak deteksi)
kotak ini menunjukan lokasi tempat bercak atau area yang terinfeksi
 - Kelas label (label penyakit)
Label ini di ambil dari hasil pelatihan model berdasarkan deteksi penyakit daun padi
 - Confidence score (tingkat keyakinan)
Semakin tinggi confidence semakin tinggi akurat prediksi model
- e) Tahapan ini sangat penting karena memberikan informasi langsung kepada pengguna mengenai apa jenis penyakit yang dideteksi serta di mana letak letak gejalanya pada gambar daun padi.

D. Teknik Pengujian Sistem

Pengujian data bertujuan buat mengevaluasi kinerja model deteksi penyakit daun padi yang telah dilatih memakai prosedur pemecahan YOLOv8.menggunakan metode YOLOv8 ialah proses yang krusial buat memastikan keefektifan dan keandalan sistem yang dikembangkan.pengujian akan melibatkan sistem seperti akurasi,persisi serta recall buat mendeteksi penyakit daun padi. Teknik yang digunakan pada pengujian ini meliputi beberapa metode menjadi berikut:

- 1) Split dataset (hold-out validation)

Dataset pada bagi menjadi dua utama:

- Data Latih (training Set): dipergunakan buat melatih contoh YOLOv8 agar mengenali pola penyakit.
- Data Uji (Testing Set): dipergunakan buat menguji kinerja model setelah pelatihan, tanpa mempengaruhi proses pembelajaran.

2) Evaluasi Menggunakan Metode Confusion Matrix

Evaluasi kinerja sistem deteksi penyakit pada daun padi dilakukan menggunakan metode Confusion Matrix, yang merupakan alat analisis untuk menilai performa model klasifikasi, dalam hal ini algoritma YOLO (You Only Look Once).

Confusion Matrix membantu mengukur sejauh mana model dapat mengenali daun padi yang sehat dan yang terkena penyakit, serta mengidentifikasi kesalahan prediksi.

Table 2 Confusion Matrix

	Prediksi positif	Prediksi negatif
Aktual positif	True positive (TP)	False negative (FN)
Aktual negatif	False positive (FP)	True Negative (TN)

Penjelasan dalam konteks penelitian adalah sebagai berikut:

1. True positive (TP)

Daun padi yang benar-benar sakit dan terdeteksi sebagai sakit sistem yolo

2. False negative (FN)

Daun padi yang sakit, tetapi tidak terdeteksi oleh sistem. ini berbahaya karena bisa menyebabkan keterlambatan penanganan

3. False positif (FP)

Daun padi yang sehat tetapi salah terdeteksi sakit ini bisa menimbulkan kekhawatiran atau tindakan yang tidak perlu

4. True negative (TN)

Daun padi yang sehat dan terdeteksi dengan benar sebagai sehat oleh sistem

Menjadi dasar untuk menghitung matrik evaluasi adalah sebagai berikut:

a) Akurasi

Akurasi adalah matrik evaluasi yang mengukur seberapa banyak prediksi yang benar (baik positif maupun negatif) dibandingkan dengan seluruh jumlah prediksi yang dilakukan oleh model.

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

b) Presisi

Presisi (Precision) adalah metrik evaluasi yang mengukur seberapa akurat prediksi positif yang dibuat oleh model.

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (6)$$

c) Recall

Recall (juga disebut Sensitivity atau True Positive Rate) adalah metrik evaluasi yang mengukur kemampuan model dalam menemukan semua data yang benar-benar positif.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (7)$$

d) F1 score

F1 Score adalah metrik evaluasi dalam machine learning yang mengukur keseimbangan antara presisi (precision) dan recall, terutama berguna saat data tidak seimbang (misalnya kasus positif jauh lebih sedikit dari negatif).

$$\text{F1 Score} = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (8)$$

Penggunaan confusion matrix dalam evaluasi ini sangat penting untuk memastikan sistem deteksi penyakit daun padi berbasis YOLO bekerja secara optimal, sehingga dapat membantu petani dalam mengambil keputusan yang cepat, akurat, dan efisien terhadap pencegahan maupun penanganan penyakit tanaman.

1) Mean Average Precision (mAP)

Mean Average Precision (mAP) merupakan salah satu metrik evaluasi primer yang dipergunakan buat mengukur kinerja contoh deteksi objek mirip YOLO (You Only Look Once). dalam konteks penelitian ini, mAP digunakan buat menilai seberapa baik contoh YOLOv8 dalam mendeteksi serta

mengklasifikasikan penyakit di daun padi sesuai lokasi dan label objek di gambar. mAP mengukur seberapa baik model mendeteksi serta mengklasifikasikan objek (penyakit daun padi) berdasarkan posisi dan label yang benar. Nilai mAP berkisar antara 0 hingga 1 atau 0%–100%. Semakin tinggi mAP, semakin baik kinerja contoh.

2) Visualisasi Hasil Deteksi

Visualisasi ini dilakukan menggunakan menampilkan bounding box (kotak pembatas) yang mengelilingi area daun padi yang terdeteksi mengandung penyakit. Tujuan dari visualisasi ini artinya buat memastikan bahwa sistem deteksi tidak hanya seksama secara angka, namun juga bisa mengidentifikasi lokasi penyakit dengan sempurna dan kentara pada gambar asli. Hal ini sangat membantu dalam validasi akibat, baik sang peneliti juga pengguna akhir seperti petani, sebab mereka bisa langsung melihat bagaimana sistem bekerja dalam mengenali tanda-tanda penyakit pada daun padi. menggunakan demikian, visualisasi hasil deteksi sebagai pelengkap penting pada menunjukkan keandalan serta penerapan mudah dari sistem yang dikembangkan

E. Teknik Analisis Data

Dalam penelitian ini, teknik analisis data dilakukan buat mengevaluasi performa deteksi penyakit di daun padi proses pelatihan dan pengujian menggunakan prosedur pemecahan YOLOv8. Analisis dilakukan secara kualitatif (melalui visualisasi hasil deteksi). Dengan tujuan untuk memahami dan menggambarkan bagaimana implementasi algoritma yolo dapat digunakan untuk mendeteksi penyakit pada daun padi secara otomatis. Pendekatan ini dipilih untuk mengeksplorasi secara mendalam proses, sistem, serta manfaat penerapan teknologi deteksi citra dalam konteks pertanian, khususnya pada petani padi di Desa Jangan-Jangan. Lokasi ini dipilih karena merupakan daerah yang mayoritas masyarakatnya bermata pencaharian sebagai petani padi dan sering mengalami kesulitan dalam mengenali penyakit pada tanaman secara cepat dan akurat. Pengumpulan dataset dilakukan dengan cara observasi, wawancara, dan pengambilan dokumentasi.

a) Teknik analisis data

Data penelitian ini analisis secara kualitatif melalui beberapa tahapan yaitu:

- Reduksi Data

Pada tahap ini, peneliti menyaring dan merangkum data yang diperoleh dari hasil observasi, wawancara, serta dokumentasi, sehingga hanya data yang relevan dan sesuai dengan fokus penelitian yang digunakan untuk dianalisis lebih lanjut.

- Penyajian Data

Data yang telah direduksi kemudian disusun secara sistematis dalam bentuk narasi, tabel, maupun visual (seperti gambar hasil deteksi gejala penyakit pada daun padi), dengan tujuan mempermudah proses interpretasi dan pemahaman terhadap hasil penelitian.

- Penarikan Kesimpulan

Tahap akhir ini dilakukan dengan menginterpretasikan seluruh data yang telah disajikan, baik hasil pengujian sistem deteksi berbasis YOLO, tanggapan petani terhadap sistem, maupun efektivitas penggunaan teknologi ini dalam mendeteksi penyakit daun padi di lapangan.

b) Validasi data

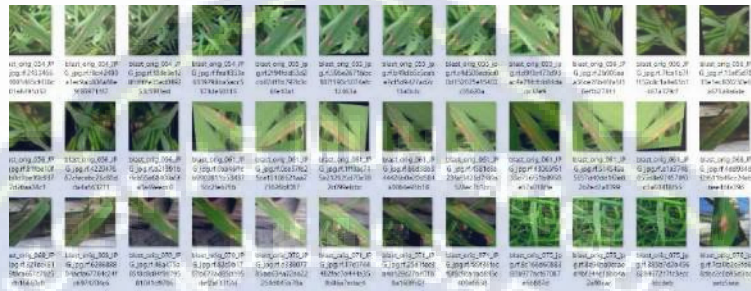
Untuk memastikan keakuratan data, penelitian ini menggunakan teknik triangulasi sumber, yaitu membandingkan dan mengkonfirmasi data yang diperoleh dari berbagai metode, seperti observasi, wawancara, dan dokumentasi. Pendekatan ini bertujuan untuk menghasilkan kesimpulan yang lebih kuat, objektif, dan dapat dipertanggungjawabkan.

BAB IV

HASIL DAN PEMBAHASAN

A. Pengumpulan dataset

Proses pengambilan dataset di lakukan secara langsung di sawah di desa jang-an-jangan kec.pujannting kab.barru dengan dataset di bagian menjadi 3 kelas penyakit.keseluruhan dataset terdiri dari 1431 data gambar yang berhasil di kumpulkan.adapun gambar-gambar di bawah ini merupakan contoh dataset yang saya ambil langsung di sawah



Gambar 6 Penyakit Blast



Gambar 7 Penyakit Hawar Bacteri



Gambar 8 Bercak Coklat

Table 3 Jumlah Data

Jenis penyakit	Jumlah data
Penyakit blast	442
Penyakit hawar bakteri	500
Bercak coklat	489
Total	1431

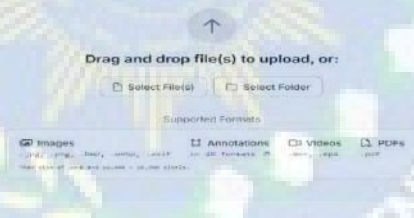
Setelah semua data terkumpul, data tersebut akan di bagi menjadi 3 bagian 80% untuk data train, 10% untuk data validasi, 10% data test. pembagian ini dilakukan agar pengujian dapat menghasilkan nilai akurasi yang tinggi.

B. Anotasi Data Di Roboflow

Pada tahap ini, gambar-gambar di beri label untuk memungkinkan mengenali objek yang akan di deteksi. Proses pelabelan menggunakan roboflow, dengan tujuan menambahkan label pada gambar-gambar untuk mendeteksi penyakit pada padi, sesuai dengan pembagian dan kategori yang telah di tentukan

a) Uploud dataset

Data gambar yang telah di kumpulkan menggunakan kamera smarphone di upload di Robotflow sebelum proses di lakukan



Gambar 9 Upload Dataset

b) Pembuatan kelas dataset

Tujuan utama dari pembuatan kelas ini adalah untuk mempermudah proses pengelompokan dataset gambar yang sebelumnya telah dianotasi menggunakan bounding box. Dalam pengolahan data visual, khususnya pada proyek deteksi objek atau klasifikasi berbasis visi komputer, pengelompokan data menjadi bagian penting agar data yang tersedia terorganisasi dengan baik dan siap digunakan dalam proses pelatihan model. Melalui sistem ini, setiap gambar yang sudah diberi anotasi akan secara otomatis diklasifikasikan ke dalam tiga kategori yang telah ditentukan sebelumnya.

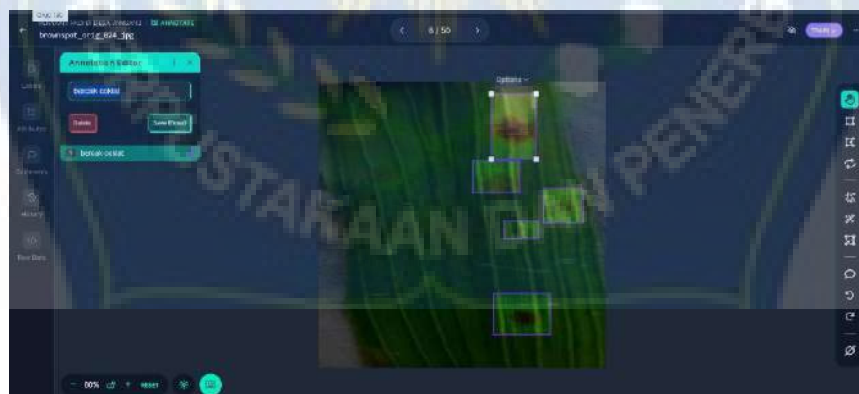
Pengelompokan ini dilakukan berdasarkan label hasil anotasi, sehingga setiap data akan tersimpan di direktori atau folder yang sesuai dengan kategori tersebut. Dengan demikian, proses pengelolaan data menjadi lebih efisien, rapi, dan terstruktur, serta memudahkan pengguna dalam melakukan validasi, analisis, maupun pengembangan lebih lanjut. Selain itu, sistem ini juga diharapkan dapat mengurangi risiko terjadinya kesalahan penempatan file dan mendukung standarisasi dalam pengolahan dataset, terutama dalam skenario yang melibatkan skala data yang besar. Dengan adanya pengelompokan otomatis, pengguna tidak perlu lagi melakukan klasifikasi gambar secara manual yang memakan waktu dan rentan terhadap human error.



Gambar 10 Jenis Penyakit

C. Labeling data di roboflow

Pelabelan gambar dilakukan melalui platform Roboflow dengan cara menambahkan bounding box atau kotak pembatas di sekitar objek yang terdapat dalam gambar. Setelah bounding box dibuat, sistem secara otomatis menampilkan pilihan kelas berdasarkan kategori yang telah ditentukan sebelumnya, sehingga memudahkan proses anotasi.



Gambar 11 Proses Pelabelan

D. Pembagian atau split data training dan testing

Pembagian dataset dilakukan secara sistematis ke dalam tiga subset utama, yaitu training set, validation set, dan test set. Sebanyak 1002 gambar (atau 70% dari total dataset) digunakan sebagai training set, yang berfungsi untuk melatih model

dalam mengenali pola dan karakteristik objek dalam gambar. Kemudian, sebanyak 286 gambar (20%) dialokasikan untuk validation set, yang digunakan untuk mengevaluasi performa model selama proses pelatihan berlangsung, sekaligus mencegah terjadinya overfitting. Terakhir, test set terdiri dari 143 gambar (10%) yang digunakan untuk menguji akurasi dan kemampuan generalisasi model terhadap data yang belum pernah dilihat sebelumnya. Pembagian ini dirancang untuk memastikan bahwa model dapat belajar dengan optimal serta memiliki kemampuan prediksi yang baik pada data nyata.

E. Proses Training



Gambar 12 Split Dataset

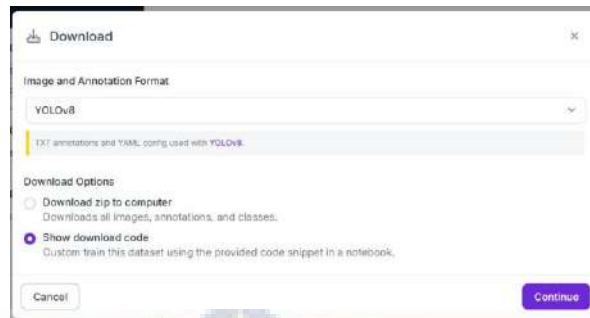
Setelah proses pengolahan dan anotasi data selesai dilakukan sesuai dengan kebutuhan, dataset tersebut kemudian diekspor untuk digunakan dalam proses pelatihan model di platform Google Colab. Proses ekspor ini menghasilkan sebuah tautan API yang berfungsi sebagai akses langsung ke dataset yang telah diproses di platform Roboflow. API ini nantinya digunakan di lingkungan Google Colab untuk mengunduh dataset secara otomatis dan terintegrasi, sehingga memudahkan pengguna dalam menyiapkan data pelatihan tanpa perlu mengunduh dan mengunggah ulang secara manual. Dengan memanfaatkan API ini, proses pelatihan model YOLOv8 di Google Colab dapat berjalan lebih praktis, efisien, dan terhubung langsung dengan dataset yang telah dianotasi, sehingga memastikan data yang digunakan sudah sesuai dengan standar yang diinginkan.

1. Export dataset roboflow

Setelah proses anotasi dataset selesai dilakukan, langkah selanjutnya adalah mengekspor dataset menggunakan format YOLOv8 Export. Proses ekspor ini bertujuan untuk mengubah hasil anotasi menjadi format yang sesuai dan kompatibel dengan arsitektur YOLOv8, sehingga dataset siap digunakan dalam proses pelatihan model.

Dengan menggunakan fitur Export YOLOv8 yang tersedia di platform Roboflow, seluruh gambar beserta file anotasi akan disusun dalam struktur folder dan format file yang telah disesuaikan dengan kebutuhan sistem deteksi YOLOv8. Hal ini

mempermudah integrasi dataset ke dalam proses training, baik secara lokal maupun melalui platform seperti Google Colab.



Gambar 13 Proses Pembuatan Api

Setelah proses ekspor dataset selesai dilakukan, Roboflow akan menyediakan sebuah API endpoint yang berisi tautan akses ke dataset yang telah dianotasi dan diekspor. API ini berfungsi sebagai penghubung antara dataset di Roboflow dengan lingkungan pengembangan model, khususnya saat melakukan pelatihan di Google Colab.

Dengan memanfaatkan API tersebut, pengguna dapat langsung memanggil dan mengunduh dataset ke dalam sesi Google Colab tanpa perlu mengunduh secara manual. Hal ini tidak hanya mempercepat proses persiapan pelatihan, tetapi juga memastikan bahwa dataset yang digunakan adalah versi terbaru dan sesuai dengan hasil anotasi yang telah dilakukan di Roboflow.

```
!pip install ultralytics  
!pip install roboflow
```

Gambar 14 Hasil Export Dataset Menjadi Bentuk Api

2. Training di googlecolab

Proses pelatihan model dilakukan dengan menggunakan API yang sebelumnya diperoleh dari Roboflow. API ini digunakan sebagai jalur akses untuk mengambil dataset yang telah dianotasi dan diekspor. Dalam proses pelatihan di Google Colab, API tersebut dipanggil secara langsung agar dataset dapat diunduh dan digunakan sebagai data pelatihan model.

Dengan cara ini, pengguna tidak perlu lagi melakukan pengunduhan manual, karena seluruh proses pengambilan data hingga pelatihan model dapat berjalan secara otomatis dan terintegrasi di lingkungan Google Colab. Pendekatan ini

memungkinkan proses training berjalan lebih efisien dan memastikan dataset yang digunakan sesuai dengan hasil anotasi terbaru dari Roboflow.

```
[ ] rf = Roboflow(api_key="f1bdk3k207a33515jff")

[ ] project = rf.workspace("suandi-arifmujiaya-q1co6").project("penyakit-padi-di-desa-jangan2")
dataset = project.version(1).download("yolov8") # Format yolov8

loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip to penyakit-padi-di-desa-jangan2-1 to yolov8:: 100% [██████████] 36205/36205 [00:01:400:00, 32172.911t/s]
Extracting Dataset Version Zip to penyakit-padi-di-desa-jangan2-1 in yolov8:: 100% [██████████] 2871/2871 [00:00:00:00, 7670.841t/s]
```

Gambar 15 Proses Pemanggilan Api

Penjelasan:

- API Key Berfungsi sebagai kunci autentikasi agar sistem dapat mengakses dataset milik Anda di Roboflow.
- Workspace Memilih Workspace milik Anda di Roboflow.
- Project Memilih Project yang berisi dataset yang sudah dianotasi.
- Workspace tempat semua project disimpan.
- Project satu kelompok dataset yang dikelola/anotasi di dalam workspace.
- Version(1)Memilih versi ke-1 dari dataset tersebut.
- Download("yolov8")Mendownload dataset dengan format YOLOv8, yang cocok digunakan untuk training YOLOv8.
- Dataset yang diunduh akan otomatis diekstrak di direktori kerja Google Colab.

```
86 epochs completed in 0.767 hours.
optimizer stripped from pendeteksi-padi/yolov8n-train/weights/last.pt, 6.3MB
optimizer stripped from pendeteksi-padi/yolov8n-train/weights/best.pt, 6.3MB

Validating pendeteksi-padi/yolov8n-train/weights/best.pt...
ultralytics 8.3.160 python 3.11.13 torch 2.6.0+cu124 CUDA0 (Tesla T4, 15099MiB)
Model Summary (fused): 72 layers, 3,006,233 parameters, 0 gradients, 8.1 GLOPs

Class Images Instances Box(P R mAP50 mAP50-95) 100% ██████████ 9/9 [00:05:00:00, 1.61it/s]
  all      280      611    0.538  0.528  0.498  0.185
  bercak coklat 104      165    0.582  0.595  0.6   0.227
  haur bakteri  193      190    0.578  0.617  0.582  0.234
  penyakitblast  72       236    0.455  0.352  0.313  0.052

Speed: 0.3ms preprocess, 2.7ms inference, 0.8ms loss, 4.9ms postprocess per image
Results saved to pendeteksi-padi/yolov8n-train
ultralytics.utils.metrics.DetMetrics object with attributes:
```

Gambar 16 Hasil Training

Gambar di atas adalah hasil evaluasi model YOLOv8 setelah proses training selesai. Berikut penjelasan tiap bagian pentingnya:

1. Training Berhasil (86 epochs)

- Model dilatih selama 86 epoch dalam waktu 0.767 jam.
- Selama proses pelatihan ini, model mengalami pembaruan bobot secara bertahap di setiap epoch agar mampu mengenali pola data dengan lebih baik. Jumlah epoch yang digunakan bertujuan untuk memastikan model memperoleh pembelajaran yang optimal tanpa mengalami overfitting.

Waktu pelatihan yang relatif singkat menunjukkan bahwa model dapat dilatih secara efisien dengan konfigurasi dan dataset yang digunakan.

- File model disimpan di pendeteksi-padi/YOLOv8n-train/weights/best.pt dan last.pt (ukuran 6.3 MB)

Hasil pelatihan model disimpan dalam file best.pt dan last.pt yang berada di direktori pendeteksi-padi/YOLOv8n-train/weights/ dengan ukuran masing-masing sekitar 6,3 MB. File best.pt merupakan model dengan performa terbaik selama proses pelatihan berdasarkan evaluasi validasi, sedangkan last.pt adalah model hasil dari epoch terakhir. Kedua file ini dapat digunakan untuk melakukan pengujian atau penerapan model secara langsung di aplikasi pendeteksi penyakit daun padi.

2. Validasi Model

- YOLOv8 versi 8.3.1 dengan Python 3.11.13 menggunakan CUDA GPU (Tesla T4).

YOLOv8 merupakan versi terbaru dari keluarga algoritma YOLO (You Only Look Once) yang dirancang khusus untuk deteksi objek dengan kecepatan dan akurasi tinggi. Pada versi 8.3.1 ini, terdapat berbagai pembaruan fitur dan peningkatan performa, baik dari sisi arsitektur model maupun efisiensi pemrosesan. Penggunaan Python 3.11.13 memastikan kompatibilitas dengan pustaka-pustaka yang dibutuhkan selama pelatihan. Selain itu, dukungan CUDA GPU dari perangkat keras Tesla T4 memungkinkan proses komputasi berjalan lebih cepat dibandingkan menggunakan CPU biasa. Dengan kemampuan pemrosesan paralel dari Tesla T4, waktu pelatihan model dapat dipersingkat secara signifikan, serta memungkinkan pengolahan data dalam jumlah besar secara lebih efisien.

- Model memiliki 72 layer dan sekitar 3 juta parameter.

Jumlah layer dan parameter ini menunjukkan kompleksitas arsitektur model yang cukup efisien untuk mendeteksi objek pada gambar. Dengan 72 layer, model mampu membangun representasi fitur yang mendalam dari data pelatihan, sedangkan jumlah parameter sekitar 3 juta memastikan model tetap ringan dan cepat dalam proses inferensi. Arsitektur ini dirancang untuk mencapai keseimbangan antara akurasi deteksi dan kecepatan pemrosesan,

sehingga cocok digunakan untuk aplikasi real-time, terutama di perangkat dengan keterbatasan sumber daya. Jumlah parameter yang relatif kecil juga mengurangi risiko overfitting, namun tetap cukup untuk menghasilkan performa deteksi yang optimal.

- Box

Box (Bounding Box) merupakan hasil keluaran model berupa kotak pembatas yang mengelilingi objek target pada gambar. Pada proses deteksi, model menghasilkan prediksi dalam bentuk bounding box yang berisi informasi lokasi dan ukuran objek yang terdeteksi. Setiap box dilengkapi dengan koordinat posisi (x, y), lebar, tinggi, serta skor kepercayaan (confidence score) yang menunjukkan seberapa yakin model terhadap hasil deteksi tersebut. Akurasi box sangat bergantung pada kemampuan model dalam mempelajari pola dan ciri khas dari objek yang dideteksi. Semakin presisi posisi dan ukuran bounding box terhadap objek sesungguhnya, semakin baik pula performa deteksi model.

Dalam pengujian ini, bounding box menjadi indikator utama yang digunakan untuk menilai efektivitas model dalam mengidentifikasi penyakit pada daun padi, baik dalam pengujian gambar diam maupun dalam pengujian real-time melalui video atau kamera.

- $\text{Box(R)} = \text{Recall deteksi bounding box}$

Box (R) atau Recall Bounding Box merupakan metrik evaluasi yang menunjukkan sejauh mana model mampu mendeteksi semua objek yang seharusnya terdeteksi dalam suatu gambar. Recall ini mengukur persentase objek target yang berhasil dikenali dan diberikan bounding box oleh model dibandingkan dengan jumlah objek sebenarnya di dalam dataset. Nilai recall yang tinggi menunjukkan bahwa model mampu menangkap hampir semua objek yang ada, sedangkan nilai yang rendah menandakan masih banyak objek yang terlewat (missed detection).

Dalam konteks pendeteksian penyakit daun padi, Box Recall sangat penting karena menunjukkan kemampuan model dalam menemukan seluruh gejala penyakit yang ada di gambar tanpa melewatkan satu pun. Hal ini menjadi indikator bahwa model tidak hanya akurat, tetapi juga sensitif terhadap kemunculan objek target di berbagai kondisi gambar.

- $\text{Box(mAP50)} = \text{Mean Average Precision pada IoU 0.5}$

Box (mAP50) atau Mean Average Precision pada threshold IoU 0.5 merupakan salah satu indikator utama dalam mengukur performa model deteksi objek. mAP50 dihitung berdasarkan rata-rata presisi model saat Intersection over Union (IoU) antara bounding box prediksi dan ground truth mencapai minimal 0.5 (50%). IoU sendiri mengukur seberapa besar tumpang tindih antara kotak prediksi dengan kotak sebenarnya. Jika $\text{IoU} \geq 0.5$, maka prediksi dianggap benar (true positive), sehingga mAP50 menggambarkan rata-rata keberhasilan model dalam mendeteksi objek dengan tingkat ketepatan posisi yang memadai. Semakin tinggi nilai mAP50, semakin baik performa model karena berarti model tidak hanya mendeteksi objek, tetapi juga memposisikannya dengan akurat.

Dalam kasus pendeteksian penyakit daun padi, mAP50 menjadi acuan penting untuk menilai apakah model mampu memberikan deteksi yang tepat secara posisi dan bentuk terhadap gejala penyakit yang muncul di gambar.

- $\text{Box(mAP50-95)} = \text{mAP rata-rata dari IoU 0.5 sampai 0.95}$

Box (mAP50-95) merupakan metrik Mean Average Precision yang dihitung berdasarkan rata-rata dari berbagai threshold Intersection over Union (IoU) mulai dari 0.5 hingga 0.95 dengan interval 0.05. Berbeda dengan mAP50 yang hanya menggunakan satu ambang batas ($\text{IoU} = 0.5$), mAP50-95 memberikan gambaran performa model secara lebih menyeluruh di berbagai tingkat ketelitian overlap antara bounding box prediksi dan ground truth. Semakin tinggi nilai mAP50-95, semakin baik kemampuan model dalam mendeteksi dan memposisikan objek secara presisi, baik dalam kasus overlap yang longgar maupun yang sangat ketat. mAP50-95 sering dianggap sebagai indikator evaluasi yang lebih ketat dan representatif karena memperhitungkan berbagai skenario kedekatan bounding box. Dalam konteks pendeteksian penyakit daun padi, nilai mAP50-95 yang tinggi menunjukkan bahwa model tidak hanya mampu mendeteksi, tetapi juga dapat memberikan prediksi lokasi penyakit dengan tingkat ketelitian yang konsisten di berbagai kondisi gambar.

1. Kecepatan Inference

a) Preprocess: 0.3 ms

Preprocess sebesar 0.3 milidetik menunjukkan waktu yang dibutuhkan model untuk melakukan pra-pemrosesan data sebelum proses inferensi dilakukan. Tahap pra-pemrosesan meliputi beberapa langkah penting, seperti pengubahan ukuran gambar, normalisasi piksel, dan penyesuaian format input agar sesuai dengan kebutuhan model. Meskipun proses ini terkesan sederhana, pra-pemrosesan berperan penting dalam memastikan data yang masuk dapat diproses secara optimal oleh model. Waktu pra-pemrosesan yang sangat singkat, yaitu hanya 0.3 ms, menunjukkan efisiensi sistem dalam menyiapkan data, sehingga tidak menambah beban signifikan terhadap keseluruhan waktu inferensi. Hal ini sangat menguntungkan, terutama untuk aplikasi real-time, karena mampu menjaga kecepatan deteksi secara keseluruhan.

b) Inference: 2.7 ms

Inference sebesar 2.7 milidetik menunjukkan waktu yang dibutuhkan model untuk melakukan proses prediksi terhadap gambar input setelah tahap pra-pemrosesan selesai. Pada tahap ini, model menganalisis citra dan mengidentifikasi objek yang relevan (dalam hal ini, gejala penyakit pada daun padi) menggunakan bobot dan parameter yang telah diperoleh selama proses pelatihan. Waktu inferensi yang sangat cepat, yakni hanya 2.7 ms per gambar, menandakan bahwa model bekerja dengan efisien dan responsif, sehingga sangat cocok diterapkan dalam sistem deteksi real-time. Kemampuan ini memungkinkan proses identifikasi penyakit dilakukan hampir seketika, tanpa jeda yang signifikan, yang sangat penting dalam aplikasi lapangan seperti monitoring tanaman secara langsung menggunakan kamera.

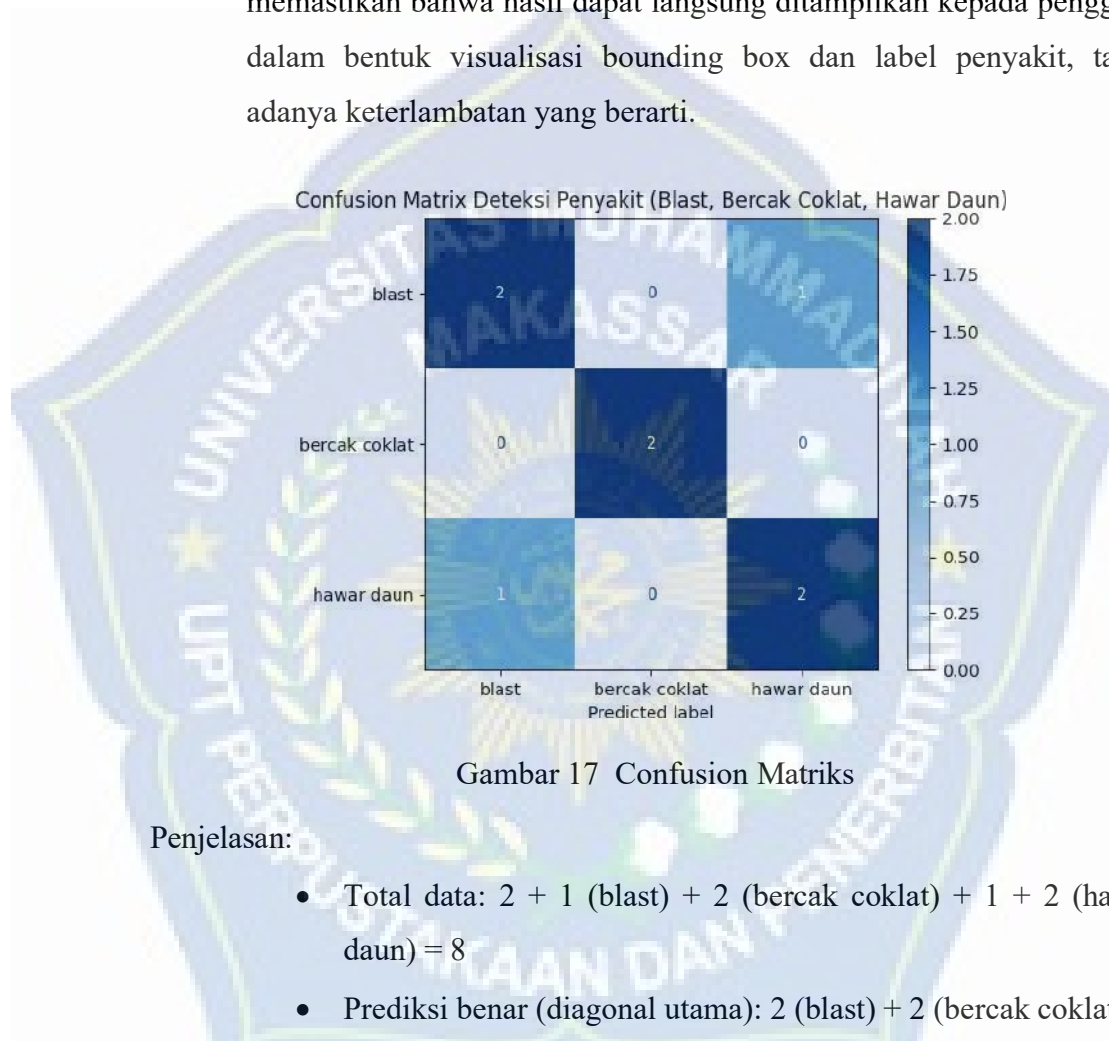
c) Postprocess: 4.9 ms

Postprocess sebesar 4.9 milidetik merupakan waktu yang dibutuhkan untuk menyelesaikan tahap akhir setelah inferensi, yaitu mengolah hasil prediksi model sebelum ditampilkan atau disimpan.

Tahap post-pemrosesan ini mencakup berbagai proses penting seperti

penerapan threshold confidence, Non-Maximum Suppression (NMS) untuk menghilangkan prediksi duplikat, serta pengelompokan hasil deteksi ke dalam label yang sesuai.

Waktu post process yang relatif singkat menunjukkan bahwa sistem mampu mengelola hasil deteksi dengan efisien tanpa menghambat keseluruhan performa deteksi. Dalam konteks aplikasi pendeteksi penyakit daun padi secara real-time, kecepatan pada tahap ini memastikan bahwa hasil dapat langsung ditampilkan kepada pengguna dalam bentuk visualisasi bounding box dan label penyakit, tanpa adanya keterlambatan yang berarti.



Gambar 17 Confusion Matriks

Penjelasan:

- Total data: 2 + 1 (blast) + 2 (bercak coklat) + 1 + 2 (hawar daun) = 8
- Prediksi benar (diagonal utama): 2 (blast) + 2 (bercak coklat) + 2 (hawar daun) = 6
- Prediksi salah: 8 - 6 = 2

1. Akurasi

$$\text{Akurasi} = \frac{\text{Jumlah Prediksi Benar}}{\text{Total Sampel}} = \frac{6}{8} = 0.75 \text{ atau } 75 \%$$

2. Penyakit bercak coklat

$$\text{Precision bercak} = \frac{2}{2+0} = \frac{2}{2} = 1,00$$

$$\text{Recall bercak} = \frac{2}{2+0} = \frac{2}{2} = 1,00$$

$$\text{F1 bercak} = 2 \times \frac{1,00 \times 1,00}{1,00 + 1,00} = 2 \times \frac{1,00}{2,00} = 1,00$$

3. Penyakit hawar daun

$$\text{Precision hawar} = \frac{2}{2+1} = \frac{2}{3} = 0,67$$

$$\text{Recall hawar} = \frac{2}{2+1} = \frac{2}{3} = 0,67$$

$$\text{F1 bercak} = 2 \times \frac{0,67 \times 0,67}{0,67 + 0,67} = 2 \times \frac{0,4489}{1,34} = 0,67$$

4. Penyakit blast

$$\text{Precision blast} = \frac{\text{Jumlah predeksi benar}}{\text{Jumlah predeksi ke blast}} = \frac{2}{3+1} = \frac{2}{4} = 0,5$$

$$\text{Recall blast} = \frac{\text{Jumlah predeksi benar}}{\text{Jumlah predeksi ke blast}} = \frac{2}{2+1} = \frac{2}{3} = 0,67$$

$$\text{F1 blast} = 2 \times \frac{0,5 \times 0,67}{0,5 + 0,67} = 2 \times \frac{0,335}{1,17} = 0,57$$

Perhitungan manual Yolo:

1) Confidence Score

$$P(\text{Object}) = 0.85$$

$$\text{IoU} = 0.7716535433$$

$$\text{IoU} = 0.7716535433$$

$$\text{IoU} = 0.7716535433$$

$$\text{Confidence Score} = 0.85 \times 0.7716535433 = 0.6559055118$$

$$\text{Jadi Confidence Score} = 0.656$$

2) Confidence

$$P(\text{Object}) = 0.85$$

$$\text{IoU} = 0.7716535433$$

$$\text{IoU} = 0.7716535433$$

$$\text{Confidence} = 0.85 \times 0.7716535433 = 0.6559$$

Model 85% yakin ada objek, dan prediksi kotaknya tumpang tindih 77% dengan ground truth Confidence final sekitar 65.6%.

3) Class conferendesen

$$P(\text{Object}) = 0.85$$

$$\text{IoU} = 0.7716535433$$

Distribusi kelas prediksi: [0.1, 0.7, 0.2]

artinya:

$$P(\text{Class1} \mid \text{Object}) = 0.1$$

$$P(\text{Class2} \mid \text{Object}) = 0.7$$

$$P(\text{Class3} \mid \text{Object}) = 0.2$$

a) Untuk Class₁:

$$0.85 \times 0.77165 \times 0.1 = 0.0656$$

b) Untuk Class₂:

$$0.85 \times 0.77165 \times 0.7 = 0.4591$$

c) Untuk Class₃:

$$0.85 \times 0.77165 \times 0.2 = 0.1312$$

Jadi kelas dengan confidence tertinggi adalah Class₂ (0.4591) itulah yang dipilih YOLO sebagai hasil deteksi.

4) Final Score

$$P(\text{Object}) = 0.85$$

$$\text{IoU} = 0.77165$$

$$\text{IoU} = 0.77165$$

$$\text{Confidence} = 0.85 \times 0.77165 = 0.6559 = 0.85 \times 0.77165 = 0.6559$$

Distribusi kelas (dari model):

$$P(\text{Class1} \mid \text{Object}) = 0.1$$

$$P(\text{Class2} \mid \text{Object}) = 0.7$$

$$P(\text{Class3} \mid \text{Object}) = 0.2$$

1. Class_1

$$0.1 \times 0.6559 = 0.06560.1$$

2. Class_2

$$0.7 \times 0.6559 = 0.45910.7$$

3. Class_3

$$0.2 \times 0.6559 = 0.13120.2$$

Kelas dengan Final Score tertinggi adalah Class_2 (0.4591), sehingga YOLO memilih kotak ini sebagai deteksi dengan label Class_2 .

5) LOS

$$\text{Prediksi: } [x, y, w, h] = [0.5, 0.5, 0.3, 0.3]$$

$$\text{Ground truth: } [0.5, 0.5, 0.4, 0.4]$$

$$\lambda_{\text{coord}} = 5$$

$$\begin{aligned} & (x - x^*)^2 + (y - y^*)^2 + (w - w^*)^2 + (h - h^*)^2 \\ &= (0.5 - 0.5)^2 + (0.5 - 0.5)^2 + (0.3 - 0.4)^2 + (0.3 - 0.4)^2 = 0 + 0 + 0.01 + 0.01 = 0.02 \end{aligned}$$

$$\text{Lcoord} = 5 \times 0.02 = 0.1$$

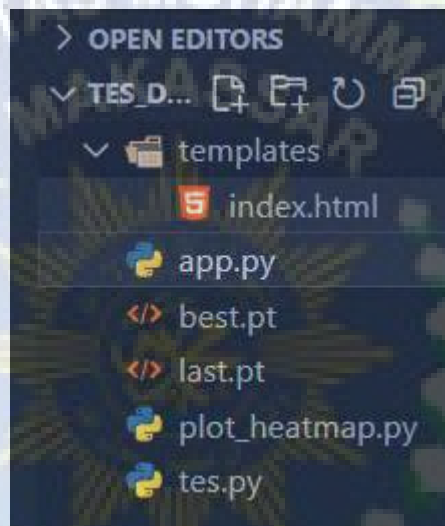
Setelah proses training selesai, akan dihasilkan file model bernama **best.pt** yang menyimpan bobot terbaik dari hasil pelatihan. File ini digunakan untuk menguji kinerja model terhadap video yang belum pernah digunakan selama proses training maupun validasi.



best.pt

Gambar 18 Folder Hasil Training

F. Pengujian System



Gambar 19 Folder Pyton Dalam Pengujian Sistem

Berikut adalah penjelasan dari struktur folder dan file yang ada di proyek Anda berdasarkan gambar:

- 1) Templates Folder ini berisi file HTML yang digunakan oleh framework web (seperti Flask) untuk tampilan antarmuka pengguna.
- 2) Index.html Ini adalah file halaman web utama yang akan ditampilkan di browser. Biasanya digunakan sebagai halaman depan aplikasi web.
- 3) App.py File Python utama yang kemungkinan besar berfungsi sebagai aplikasi backend (misalnya menggunakan Flask). Di sini biasanya didefinisikan route, fungsi pendeteksian, dan logika pengolahan data.
- 4) Best.pt File model hasil training YOLOv8 yang memiliki performa terbaik (best model checkpoint). Digunakan saat proses inferensi/deteksi objek.

- 5) Last.pt File model hasil dari epoch terakhir training. Kadang digunakan untuk perbandingan performa dengan model best.pt.
- 6) lot_heatmap.py Biasanya berisi script Python untuk membuat visualisasi heatmap, misalnya dari confusion matrix atau analisis data lainnya.
- 7) Tes.py Biasanya digunakan sebagai file pengujian (testing script). Bisa berisi fungsi percobaan, validasi model, atau testing sederhana terhadap sistem.

Sourcode Pendeteksi:



Gambar 20 Sourcode Pendeteksi




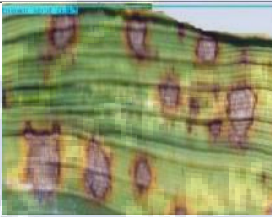








Penjelasan source code di atas menunjukkan proses pengujian sistem yang memanfaatkan model deteksi yang telah selesai dilatih sebelumnya di Google Colab. Pada tahap training, model dilatih menggunakan dataset yang telah dipersiapkan dan disesuaikan dengan kebutuhan sistem. Setelah model berhasil dilatih dan menghasilkan file model terbaik (best.pt), tahap selanjutnya adalah melakukan pengujian.

Pengujian ini dilakukan dengan menggunakan data atau objek baru yang sebelumnya tidak pernah dilihat atau dikenali oleh model saat proses trainin Tujuannya adalah untuk menguji kemampuan generalisasi model dalam mengenali objek di luar data latihnya.

Dalam pengujian ini, sistem dijalankan secara realtime dengan memanfaatkan webcam internal laptop sebagai media input. Kamera akan menangkap objek secara langsung, kemudian gambar hasil tangkapan tersebut diproses oleh model untuk dilakukan deteksi. Dengan pengujian realtime ini, dapat dilihat sejauh mana tingkat akurasi dan kecepatan respon sistem saat digunakan dalam kondisi sebenarnya.

G. Testing

Table 4 Hasil Deteksi Penyakit Bercak Coklat

Gambar	Hasil deteksi	akurasi	Penyakit sebenarn ya	Salah/benar
		0.91%	Bercak coklat	Benar
		0.93%	Bercak coklat	Benar
		0.85%	Bercak coklat	Benar
		0.81%	Bercak coklat	Benar
		0.92%	Bercak coklat	Benar
		0.80%	Bercak coklat	Benar

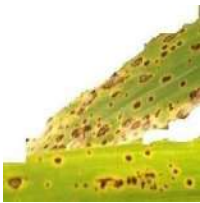










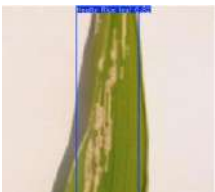
		0.82%	Bercak coklat	Benar
		0.81%	Bercak coklat	Benar
		0.81%	Bercak coklat	Benar

Table 5 Penyakit Hawar Daun

Gambar	Hasil deteksi	akurasi	Penyakit sebenarnya	Benar/salah
		0,77%	Hawar daun	Benar
		0,70%	Hawar daun	Benar
		0.81%	Hawar daun	Benar





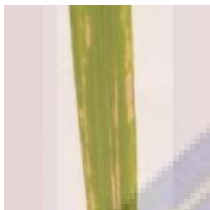



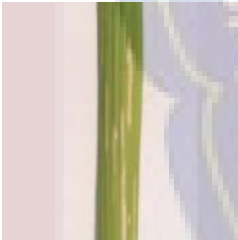







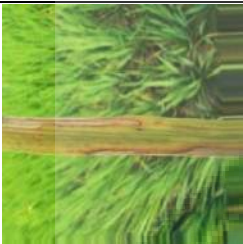
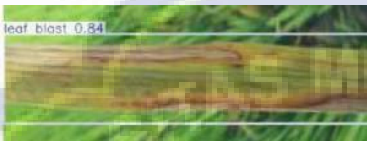
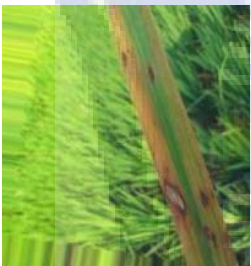

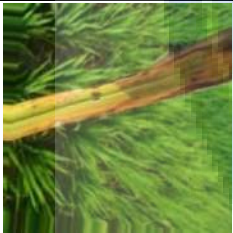



		0.78%	Hawar daun	Benar
		0.88%	Hawar daun	Benar
		0.89%	Hawar daun	Benar
		0.89%	Hawar daun	Benar
		0.77%	Hawar daun	Benar
		0.76%	Hawar daun	Benar
		0.88%	Hawar daun	Benar

Table 6 Hasil Deteksi Penyakit Padi Blast

Gambar	Penyakit yang terdeteksi	akurasi	Penyakit	
			sebenarnya	Benar/salah
		0.76%	blast	benar
		0.84%	blast	benar
		0.76%	blast	benar
		0.71%	blast	benar
		0.82%	blast	benar





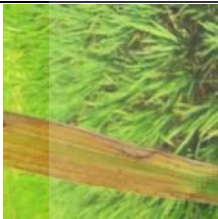



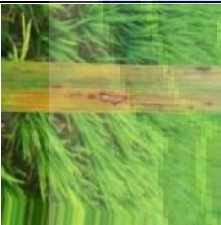
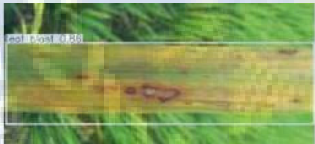




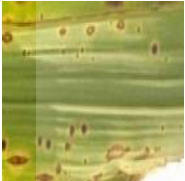
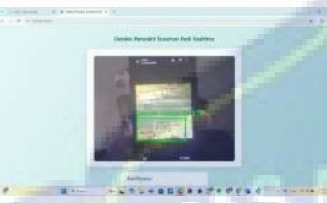

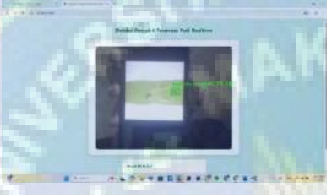

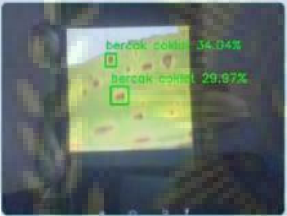


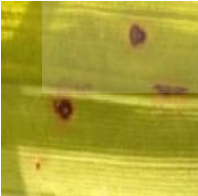



		0.77%	blast	benar
		0.79%	blast	benar
		0.81%	blast	benar
		0.90%	blast	benar
		0.86%	blast	benar
		0.75%	blast	benar

Table 7 Real Time Deteksi Bercak Coklat

Gambar	Hasil deteksi	akurasi	Penyakit sebenarnya a	Salah/benatr
		26.16%	Bercak coklat	Benar
		32.97%	Bercak coklat	Benar
		25.38%	Bercak coklat	Benar
		34.904%	Bercak coklat	Benar
		37.09%	Bercak coklat	Benar
		39.94%	Bercak coklat	Benar
		28.60%	Bercak coklat	Benar







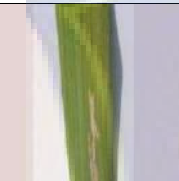

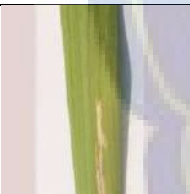







		30.36%	Bercak coklat	Benar
		41.45%	Bercak coklat	Benar

Table 8 Real Time Penyakit Hawar Daun

Gambar	Hasil deteksi	akurasi	Penyakit sebenarnya	Benar/salah
		37.15%	Hawar daun	Benar
		32.29%	Hawar daun	Benar
		38.37%	Hawar daun	Benar
		31.14%	Hawar daun	Benar
		37.40%	Hawar daun	Benar
		30.19%	Hawar daun	Benar






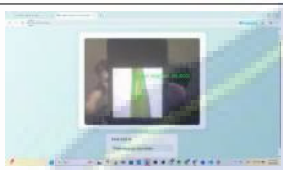

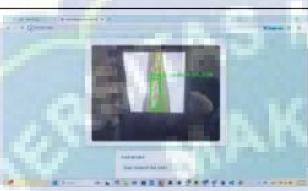
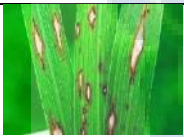
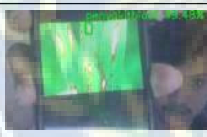
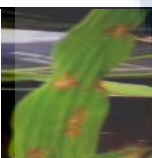
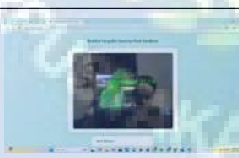
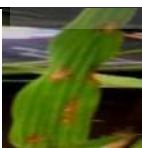

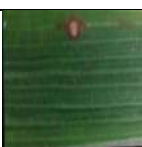







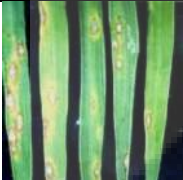

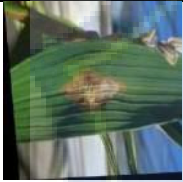





		33.21%	Hawar daun	Benar
		40.13%	Hawar daun	Benar
		28.60%	Hawar daun	Benar
		25.22%	Hawar daun	Benar

Table 9 Real Time Penyakit Blast

Gambar	Penyakit yang terdeteksi	akurasi	Penyakit sebenarnya	Benar/salah
		49.48%	blast	benar
		25.31%	blast	benar
		29%	blast	benar
		25.53%	blast	benar

		25.18%	blast	benar
		25.11%	blast	benar
		27.73%	blast	benar
		25.32%	blast	benar
		25.34%	blast	benar
		25.90%	blast	benar
		26.64%	blast	benar

H. Hasil test

1) Hasil Deteksi Tidak Real Time

Pada penyakit bercak coklat yang diuji dengan 9 gambar, seluruhnya terdeteksi dengan benar dengan rata-rata akurasi sekitar 86%. Untuk penyakit hawar daun dengan 10 gambar uji, seluruh hasil deteksi juga benar dengan rata-rata akurasi sekitar 81%. Sementara itu, pada penyakit blast yang diuji dengan 11 gambar, sistem kembali memberikan hasil yang benar seluruhnya dengan rata-rata akurasi sekitar 80%.

2) Hasil Deteksi Real Time

Berdasarkan hasil pengujian deteksi penyakit padi, sistem mampu mengenali ketiga jenis penyakit, yaitu Bercak Coklat, Hawar Daun, dan Blast dengan benar pada seluruh gambar uji. Pada deteksi penyakit Bercak Coklat sebanyak 9 gambar, seluruhnya terdeteksi sesuai dengan label sebenarnya, dengan akurasi berkisar antara 25,38% hingga 41,45%. Begitu pula pada deteksi penyakit Hawar Daun sebanyak 10 gambar, sistem berhasil mendeteksi dengan benar seluruhnya, dengan tingkat akurasi antara 25,22% hingga 40,13%. Sedangkan pada penyakit Blast, dari 11 gambar uji, seluruh hasil deteksi juga sesuai dengan label aslinya, dengan akurasi berkisar antara 25,11% hingga 49,48%.



BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pengujian sistem pendeteksi penyakit daun padi menggunakan algoritma YOLOv8 yang telah dilakukan di Desa Jangan-Jangan, Kecamatan Pujananting, Kabupaten Barru, maka dapat disimpulkan sebagai berikut:

1. Implementasi algoritma YOLOv8 dalam mendeteksi penyakit pada daun padi dapat dilakukan melalui tahapan utama, yaitu pengumpulan dan anotasi data, pelatihan model dengan parameter yang sesuai, serta pengujian sistem menggunakan data uji. Dengan langkah-langkah tersebut, sistem mampu mengenali berbagai jenis penyakit daun padi secara otomatis.
2. Tingkat akurasi sistem dalam mendeteksi penyakit daun padi menunjukkan bahwa YOLOv8 memiliki potensi yang baik sebagai metode deteksi, meskipun tingkat ketelitian hasil deteksi sangat dipengaruhi oleh kualitas data latih, jumlah dataset, serta parameter yang digunakan pada saat pelatihan model.

B. Saran

Dari hasil penelitian ini, disarankan agar pengembangan sistem pendeteksi penyakit daun padi menggunakan YOLOv8 terus ditingkatkan, khususnya dalam memperbanyak dan memperluas variasi dataset untuk meningkatkan akurasi dan generalisasi model. Selain itu, sistem sebaiknya dikembangkan dalam bentuk aplikasi mobile atau web agar lebih mudah diakses oleh petani. Pengujian di lapangan secara luas juga penting dilakukan guna memastikan efektivitas sistem dalam kondisi nyata. Kolaborasi dengan petani dan penyuluh pertanian sangat dianjurkan untuk memperluas manfaat dari sistem ini dalam mendukung pertanian yang lebih cerdas dan responsif terhadap penyakit tanaman.

DAFTAR PUSTAKA

- Agustiani, S., Tajul Arifin, Y., Junaidi, A., Khotimatul Wildah, S., & Mustopa, A. (2022). Klasifikasi Penyakit Daun Padi menggunakan Random Forest dan Color Histogram. *Jurnal Komputasi*, 10(1). <https://doi.org/10.23960/komputasi.v10i1.2961>
- Aisyah, W., Koman, F., Janur, A., Nisaul, F., Putri, I. D., & Pratiwi, G. (2021). Perbandingan Metode Otomatisasi Supervised Machine Learning terhadap Perubahan Tutupan Lahan. *Prosiding Forum Ilmiah Tahunan (FIT)- Ikatan Surveyor Indonesia (ISI)*, 1(0), 301–307. <https://proceedings.undip.ac.id/index.php/isiundip2021/article/view/617>
- Alwie, rahayu deny danar dan alvi furwanti, Prasetyo, A. B., Andespa, R., Lhokseumawe, P. N., & Pengantar, K. (2020). Tugas Akhir Tugas Akhir. *Jurnal Ekonomi Volume 18, Nomor 1 Maret 201*, 2(1), 41–49.
- Baay, M. N., Irfansyah, A. N., & Attamimi, M. (2021). Sistem Otomatis Pendeteksi Wajah Bermasker Menggunakan Deep Learning. *Jurnal Teknik ITS*, 10(1). <https://doi.org/10.12962/j23373539.v10i1.59790>
- Gazali, W., Soeparno, H., & Ohliati, J. (2014). Dalam Pengolahan Citra Digital. *Jurnal Mat Stat*, 12(2), 103–113.
- Hary Candana, E. W., Gede, I., Gunadi, A., & Divayana, D. G. H. (2021). Perbandingan Fuzzy Tsukamoto, Mamdini Dan Sugeno Dalam Penentuan Hari Baik Pernikahan Berdasarkan Wariga Menggunakan Confusion Matrix. *Jurnal Ilmu Komputer Indonesia (JIK)*, 6(2), 14–22.
- Hastari, N. P., Rohana, T., Fitri, A., Masruriyah, N., & Wahiddin, D. (2024). *CLASSIFICATION OF RICE ELIGIBILITY BASED ON INTACT AND NON-INTACT RICE SHAPES USING YOLO V8-BASED CNN ALGORITHM DAN TIDAK UTUH MENGGUNAKAN ALGORITMA CNN BERBASIS YOLO V8*. 5(4), 1309–1318.
- Manurung, D. G., Pinasthika, M. R., Vasya, M. A. O., Putri, R. A. D. S., Tampubolon, A. P., Prayata, R. F., Nisa, S. K., & Yudistira, N. (2024). Deteksi Dan Klasifikasi Hama Potato Beetle Pada Tanaman Kentang Menggunakan YOLOV8. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 11(4), 723–734. <https://doi.org/10.25126/jtiik.1148092>
- Munantri, N. Z., Sofyan, H., & Florestiyanto, M. Y. (2020). Aplikasi Pengolahan Citra Digital Untuk Identifikasi Umur Pohon. *Telematika*, 16(2), 97. <https://doi.org/10.31315/telematika.v16i2.3183>

- Pokhrel, S. (2024). No Title. *EΛENH. Aγaη*, 15(1), 37–48.
- Prayudi, Y. (2024). *Analisis Dan Klasifikasi Penyakit Pada Daun Padi Dengan Menggunakan Metode Yolov8*. 11(6), 5676–5683.
- Purnamawati, A., Nugroho, W., Putri, D., & Hidayat, W. F. (2020). Deteksi Penyakit Daun pada Tanaman Padi Menggunakan Algoritma Decision Tree, Random Forest, Naïve Bayes, SVM dan KNN. *InfoTekJar: Jurnal Nasional Informatika Dan Teknologi Jaringan*, 5(1), 212–215. <https://doi.org/10.30743/infotekjar.v5i1.2934>
- Putra Pranjaya, A., Rizki, F., Kurniawan, R., & Daulay, N. K. (2024). KLIK: Kajian Ilmiah Informatika dan Komputer Klasifikasi Penyakit Pada Daun Tanaman Padi Berbasis YoloV5 (You Only Look Once). *Media Online*, 4(6), 3127–3136. <https://doi.org/10.30865/klik.v4i6.1916>
- Qodryantha, A., Rudawan, R. A., & Sularsa, A. (2023). *Pembangunan Sistem Pengawasan Cerdas Dengan Visualisasi 3D Development Of Smart Surveillance System With 3D Visualization*. 9(1), 1–16. <https://openlibrarypublications.telkomuniversity.ac.id/index.php/appliedscience/article/view/19508?btwaf=91705735>
- Santosa, A. A., Fu'adah, R. Y. N., & Rizal, S. (2023). Deteksi Penyakit pada Tanaman Padi Menggunakan Pengolahan Citra Digital dengan Metode Convolutional Neural Network. *Journal of Electrical and System Control Engineering*, 6(2), 98–108. <https://doi.org/10.31289/jesce.v6i2.7930>
- Widyono, M. F., & Pratama, F. F. (2022). *Rancang Bangun Sistem Live Stream Video Dan Pengenalan Penyakit Padi*.
- Yasen, N. M., Rifka, S., Vitria, R., & Yulindon, Y. (2023). Pemanfaatan Yolo Untuk Deteksi Hama Dan Penyakit Pada Daun Cabai Menggunakan Metode Deep Learning. *Elektron : Jurnal Ilmiah*, 15, 63–71. <https://doi.org/10.30630/eji.0.0.397>

L

A

M

P

I

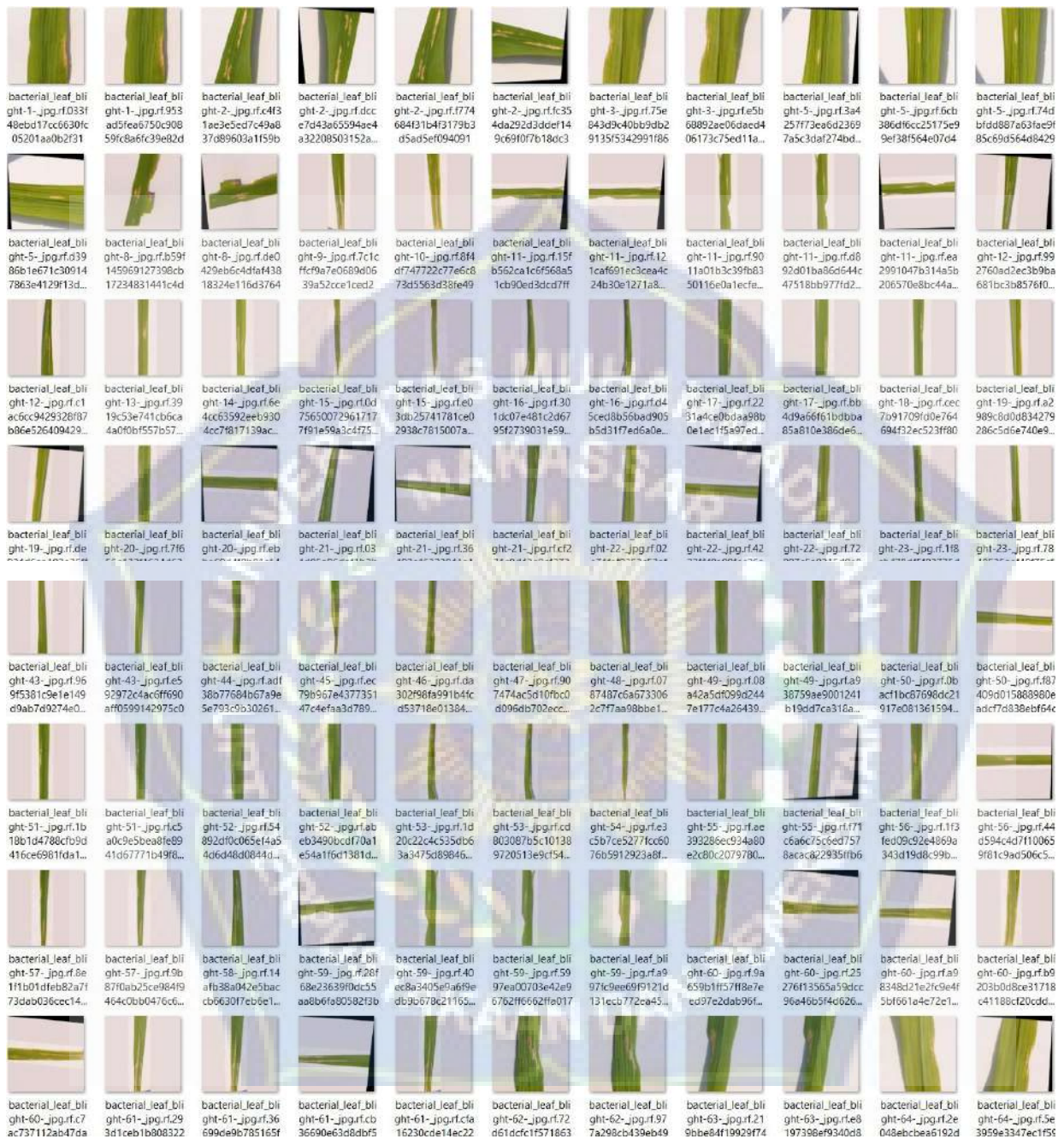
R

A



N

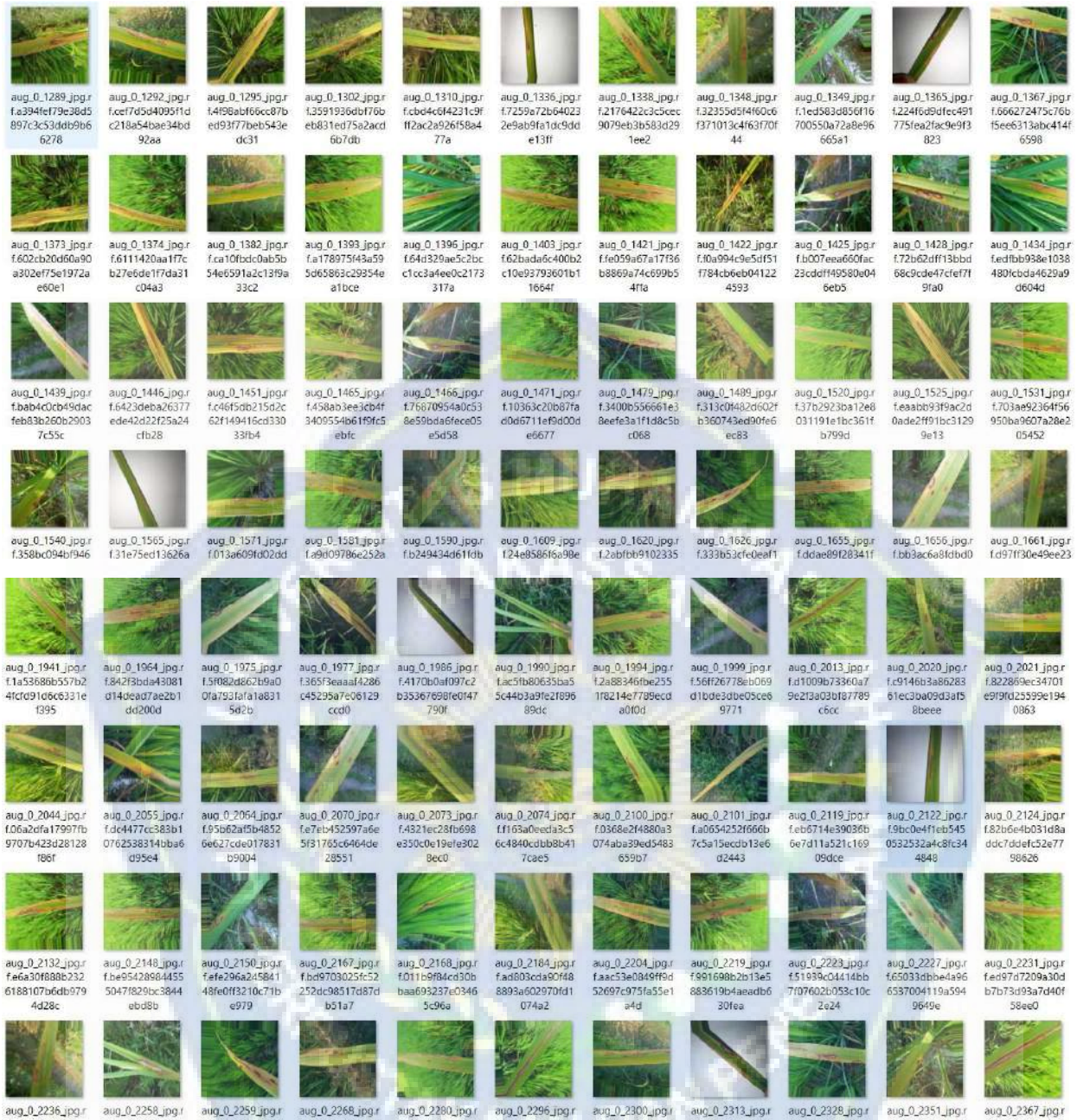
Lampiran 1 Dataset













Lampiran 2 Source Code

```
!pip install ultralytics
!pip install roboflow

from roboflow import Roboflow
from ultralytics import YOLO
import os

rf = Roboflow(api_key="vElUWkCK2NTaJJS1SjFE")

project = rf.workspace("suandi-aritmawijaya-q1cp6").project("penyakit-
daun-padi-mt7eg-mqlwm")
dataset = project.version(1).download("yolov8")
import numpy as np

model = YOLO('yolov8s.pt')

import numpy as np

def yolo_loss(pred, target, lambda_coord=5, lambda_noobj=0.5):
    """
    pred: [x, y, w, h, conf, class_probs...]
    target: sama seperti pred
    """
    # Lokasi
    loss_xy = (pred[0] - target[0])**2 + (pred[1] - target[1])**2
    loss_wh = (np.sqrt(pred[2]) - np.sqrt(target[2]))**2 +
    (np.sqrt(pred[3]) - np.sqrt(target[3]))**2
    loss_loc = lambda_coord * (loss_xy + loss_wh)

    # Confidence
    loss_conf = (pred[4] - target[4])**2

    # Klasifikasi
    loss_class = np.sum((pred[5:] - target[5:])**2)

    total_loss = loss_loc + loss_conf + loss_class
    return total_loss

# Contoh data: [x, y, w, h, conf, kelas1, kelas2, kelas3]
prediksi = np.array([0.5, 0.5, 0.3, 0.3, 0.8, 0.1, 0.7, 0.2])
ground_truth = np.array([0.5, 0.5, 0.4, 0.4, 1.0, 0.0, 1.0, 0.0])
```

```
print("Loss total:", yolo_loss(prediksi, ground_truth))
```

```
# Fungsi menghitung IoU (Intersection over Union)
def calculate_iou(box1, box2):
    """
    box format: [x1, y1, x2, y2] (xmin, ymin, xmax, ymax)
    """
    x1_inter = max(box1[0], box2[0])
    y1_inter = max(box1[1], box2[1])
    x2_inter = min(box1[2], box2[2])
    y2_inter = min(box1[3], box2[3])

    if x2_inter < x1_inter or y2_inter < y1_inter:
        return 0.0 # Tidak ada overlap

    intersection_area = (x2_inter - x1_inter) * (y2_inter - y1_inter)

    box1_area = (box1[2] - box1[0]) * (box1[3] - box1[1])
    box2_area = (box2[2] - box2[0]) * (box2[3] - box2[1])

    union_area = box1_area + box2_area - intersection_area
    iou = intersection_area / union_area
    return iou

# Contoh input
P_object = 0.85 # confidence dari model bahwa ada objek
ground_truth_box = [50, 50, 200, 200] # (xmin, ymin, xmax, ymax)
predicted_box = [60, 60, 210, 210] # prediksi dari YOLO misalnya

# Hitung IoU
iou = calculate_iou(ground_truth_box, predicted_box)

# Hitung Confidence Score
confidence_score = P_object * iou

# Cetak hasil
print("IoU:", iou)
print("Confidence Score:", confidence_score)
```

```
# Fungsi menghitung IoU (Intersection over Union)
def calculate_iou(box1, box2):
```

```

"""
box format: [x1, y1, x2, y2] (xmin, ymin, xmax, ymax)
"""

x1_inter = max(box1[0], box2[0])
y1_inter = max(box1[1], box2[1])
x2_inter = min(box1[2], box2[2])
y2_inter = min(box1[3], box2[3])

# Jika tidak ada overlap
if x2_inter < x1_inter or y2_inter < y1_inter:
    return 0.0

# Luas irisan
intersection_area = (x2_inter - x1_inter) * (y2_inter - y1_inter)

# Luas masing-masing box
box1_area = (box1[2] - box1[0]) * (box1[3] - box1[1])
box2_area = (box2[2] - box2[0]) * (box2[3] - box2[1])

# Luas gabungan
union_area = box1_area + box2_area - intersection_area

# IoU
iou = intersection_area / union_area
return iou

# Input manual
P_object = 0.9 # Probabilitas ada objek menurut model
ground_truth_box = [100, 100, 200, 200] # Format: [xmin, ymin, xmax, ymax]
predicted_box = [110, 110, 190, 190] # Box dari hasil prediksi model

# Hitung IoU
iou = calculate_iou(ground_truth_box, predicted_box)

# Hitung Confidence Score
confidence = P_object * iou

# Tampilkan hasil
print("IoU =", round(iou, 4))
print("Confidence =", round(confidence, 4))

# Contoh output model untuk satu prediksi
P_object = 0.8 # probabilitas ada objek
P_class_given_object = {
    "bercak_coklat": 0.1,

```



```

    "hawar_daun": 0.6,
    "blast": 0.3
}

# Hitung confidence setiap kelas:  $\text{Confidence}_i = P(\text{Object}) \times P(\text{Class}_i | \text{Object})$ 
confidence_scores = {}
for class_name, p_class in P_class_given_object.items():
    confidence = P_object * p_class
    confidence_scores[class_name] = confidence

# Tampilkan hasil
print("P(Object) =", P_object)
print("\nP(Classi | Object):")
for class_name, p_class in P_class_given_object.items():
    print(f" {class_name}: {p_class}")

print("\nConfidence (P(Object) × P(Classi | Object)):")
for class_name, confidence in confidence_scores.items():
    print(f" {class_name}: {round(confidence, 4)}")

# Probabilitas ada objek (dari prediksi model)
P_object = 0.85

# Probabilitas setiap kelas, diberikan objek (dari model)
P_class_given_object = {
    "bercak_coklat": 0.1,
    "hawar_daun": 0.7,
    "blast": 0.2
}

# Hitung Confidence Score (misalnya pakai IoU atau lainnya)
# Contoh di sini:  $\text{confidence} = P(\text{Object}) \times \text{IoU}$ 
IoU = 0.6
confidence = P_object * IoU

# Hitung Final Score =  $P(\text{Class}_i | \text{Object}) \times \text{Confidence}$ 
final_scores = {}
for class_name, p_class_given_object in P_class_given_object.items():
    final_score = p_class_given_object * confidence
    final_scores[class_name] = final_score

# Tampilkan hasil
print(f"P(Object): {P_object}")
print(f"IoU: {IoU}")

```

```
print(f"Confidence = P(Object) × IoU = {confidence:.4f}\n")
```

```
print("Final Score untuk setiap kelas:")  
for class_name, score in final_scores.items():  
    print(f" {class_name}: {score:.4f}")
```

```
from ultralytics import YOLO  
import os  
  
# Muat model yang sudah dilatih  
model = YOLO("/content/drive/MyDrive/Penyakit daun  
Padi.v1i.yolov8/best.pt") # Pastikan nama dan path-nya sesuai  
  
# Path ke folder gambar uji  
image_folder = "/content/drive/MyDrive/Penyakit daun  
Padi.v1i.yolov8/valid/images" # Ubah sesuai folder kamu  
  
# Buat folder output  
output_folder = "/content/hasil_deteksi"  
os.makedirs(output_folder, exist_ok=True)  
  
# Deteksi semua gambar dalam folder  
for filename in os.listdir(image_folder):  
    if filename.endswith(".jpg") or filename.endswith(".png"):  
        path_gambar = os.path.join(image_folder, filename)  
        hasil = model(path_gambar, save=True, project=output_folder,  
name="hasil")  
  
import numpy as np  
import seaborn as sns  
import matplotlib.pyplot as plt  
from sklearn.metrics import confusion_matrix, classification_report  
  
# Confusion Matrix Manual (Baris = aktual, Kolom = prediksi)  
# Kelas: [blast, bercak coklat, hawar daun]  
conf_matrix = np.array([  
    [2, 0, 1], # Actual: blast  
    [0, 2, 0], # Actual: bercak coklat  
    [1, 0, 2]  # Actual: hawar daun  
)  
  
# Generate data berdasarkan matrix  
actual = []
```

```

predicted = []

for actual_class, row in enumerate(conf_matrix):
    for predicted_class, count in enumerate(row):
        actual.extend([actual_class] * count)
        predicted.extend([predicted_class] * count)

# Label kelas
labels = ['blast', 'bercak coklat', 'hawar daun']

cm = confusion_matrix(actual, predicted)

plt.figure(figsize= (8, 6))
sns.heatmap(cm, annot=True, cmap='Blues', fmt='g',
            xticklabels=labels, yticklabels=labels)
plt.xlabel("Predicted label")
plt.ylabel("True label")
plt.title("Confusion Matrix Deteksi Penyakit Daun Padi")
plt.show()

report = classification_report(actual, predicted, target_names=labels)
print("Classification Report:")
print(report)

from google.colab import files
uploaded = files.upload()

import os
from ultralytics import YOLO

# Ambil nama file upload
uploaded_filename = list(uploaded.keys())[0]
uploaded_path = os.path.join('/content', uploaded_filename)

# Load model hasil training
model = YOLO('/content/drive/MyDrive/Penyakit daun Padi.v1i.yolov8/best.pt') # Ganti path best.pt sesuai model kamu

# Jalankan prediksi
results = model.predict(source=uploaded_path, save=True, conf=0.25)

# Tampilkan hasil prediksi
results[0].show()

```

```

from flask import Flask, render_template, Response
from ultralytics import YOLO
import cv2

app = Flask(__name__)
model = YOLO('best.pt')
cap = cv2.VideoCapture(0)

CONFIDENCE_THRESHOLD = 20

def generate_frames():
    while True:
        ret, frame = cap.read()
        if not ret:
            break

        frame = cv2.resize(frame, (640, 480))
        results = model(frame, stream=True)

        detected = False
        for info in results:
            for box in info.boxes:
                confidence = float(box.conf[0]) * 100
                cls_idx = int(box.cls[0])
                label = model.names[cls_idx] if cls_idx <
len(model.names) else f"Class {cls_idx}"
                if confidence > CONFIDENCE_THRESHOLD:
                    x1, y1, x2, y2 = map(int, box.xyxy[0])
                    cv2.rectangle(frame, (x1, y1), (x2, y2), (0,
255, 0), 3)
                    cv2.putText(frame, f'{label}
{confidence:.2f}%', (x1, y1 - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0,
255, 0), 2)
                    detected = True

        if not detected:

```

```

        cv2.putText(frame, 'Tidak ada penyakit terdeteksi',
(20, 40),
                    cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255),
2)

    ret, buffer = cv2.imencode('.jpg', frame)
    frame_bytes = buffer.tobytes()

    yield (b'--frame\r\n'
           b'Content-Type: image/jpeg\r\n\r\n' + frame_bytes
+ b'\r\n')

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/video_feed')
def video_feed():
    return Response(generate_frames(), mimetype='multipart/x-
mixed-replace; boundary=frame')

if __name__ == '__main__':
    app.run(debug=True)

```


Lampiran 3 Dokumentasi Pengambilan Dataset



 **MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH**
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN
Alamat kantor: Jl.Sultan Alauddin NO.259 Makassar 90221 Tlp.(0411) 866972,881593, Fax.(0411) 865588

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

SURAT KETERANGAN BEBAS PLAGIAT

UPT Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,
Menerangkan bahwa mahasiswa yang tersebut namanya di bawah ini:

Nama : Suandi Aritmawijaya
Nim : 105841102921
Program Studi : Teknik Informatika
Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	8%	10 %
2	Bab 2	5%	25 %
3	Bab 3	4%	10 %
4	Bab 4	0%	10 %
5	Bab 5	3%	5 %

Dinyatakan telah lulus cek plagiat yang diadakan oleh UPT- Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan
seperlunya.

Makassar, 16 Agustus 2025
Mengetahui,
Kepala UPT- Perpustakaan dan Penerbitan,


Nursandi, S. Hani, M.I.P.
NBM. 964 591

Jl. Sultan Alauddin no 259 makassar 90222
Telepon (0411)866972,881 593,fax (0411)865 588
Website: www.library.unismuh.ac.id
E-mail : perpustakaan@unismuh.ac.id

Bab I Suandi Aritmawijaya

105841102921

by Tahap Tutup

Submission date: 16-Aug-2025 05:05AM (UTC+0700)
Submission ID: 2730181635
File name: BAB_1_suandi_aritmawijaya-1.docx (22.54K)
Word count: 908
Character count: 5917

Bab I Suandi Aritmawijaya 105841102921

ORIGINALITY REPORT

8%

SIMILARITY INDEX

6%

INTERNET SOURCES

5%

PUBLICATIONS

5%

STUDENT PAPERS

PRIMARY SOURCES

1

Sarifah Agustiani, Yoseph Tajul Arifin, Agus Junaidi, Siti Khotimatul Wildah, Ali Mustopa. "Klasifikasi Penyakit Daun Padi menggunakan Random Forest dan Color Histogram", Jurnal Komputasi, 2022

Publication

5%

2

eprints.umm.ac.id

Internet Source

2%

3

e-repository.perpus.iainsalatiga.ac.id

Internet Source

2%

Exclude quotes

Off

Exclude matches

< 2%

Exclude bibliography

On

Bab II Suandi Aritmawijaya

105841102921

by Tahap Tutup

Submission date: 16-Aug-2025 05:07AM (UTC+0700)

Submission ID: 2730181906

File name: BAB_II_suandi_aritmawijaya-1.docx (164.05K)

Word count: 3377

Character count: 21282

Bab II Suandi Aritmawijaya 105841102921

ORIGINALITY REPORT

5%

SIMILARITY INDEX

6%

INTERNET SOURCES

3%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1

ojs.umrah.ac.id

Internet Source

3%

2

www.vinniola.com

Internet Source

2%

Exclude quotes

Off

Exclude matches

< 2%

Exclude bibliography

Off

Bab III Suandi Aritmawijaya

105841102921

by Tahap Tutup

Submission date: 15-Aug-2025 09:47AM (UTC+0700)

Submission ID: 2729802531

File name: BAB_III_suandi_aritmawijaya.docx (1,013.99K)

Word count: 1983

Character count: 12710

Bab III Suandi Aritmawijaya 105841102921

ORIGINALITY REPORT

4%

SIMILARITY INDEX

0% Turnitin

INTERNET SOURCES

2%

PUBLICATIONS

2%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Universitas Muhammadiyah
Makassar

Student Paper

2%

2

Ibran Simbolon, Putra Aditya, Estetika Br
Purba. "Prediksi Performa Akademik Siswa
Berdasarkan Kehadiran dan Aktivitas E-
Learning Menggunakan Algoritma Decision
Tree", RIGGS: Journal of Artificial Intelligence
and Digital Business, 2025

Publication

2%

Exclude quotes

Off

Exclude matches

< 2%

Exclude bibliography

Off

Bab IV Suandi Aritmawijaya

105841102921

by Tahap Tutup

Submission date: 15-Aug-2025 09:49AM (UTC+0700)
Submission ID: 2729803512
File name: BAB_IV_suandi_aritmawijaya.docx (18.67M)
Word count: 2921
Character count: 18636

Bab IV Suandi Aritmawijaya 105841102921

ORIGINALITY REPORT

0%

SIMILARITY INDEX

0%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS


PRIMARY SOURCES

Exclude quotes Off

Exclude bibliography Off

Exclude matches 0%





Bab V Suandi Aritmawijaya

105841102921

by Tahap Tutup

Submission date: 16-Aug-2025 05:07AM (UTC+0700)

Submission ID: 2730182080

File name: BAB_V_suandi_aritmawijaya-1.docx (16.44K)

Word count: 267

Character count: 1771

Bab V Suandi Aritmawijaya 105841102921

ORIGINALITY REPORT

3%

SIMILARITY INDEX

3%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1

repository.unhas.ac.id

Internet Source

3%

Exclude quotes Off

Exclude bibliography Off

Exclude matches 42%