

**PENGEMBANGAN SISTEM DETEKSI PENYAKIT  
DAUN PISANG UNTUK MENINGKATKAN  
KUALITAS EKSPOR MENGGUNAKAN  
ALGORITMA YOLOv5**

(Studi Kasus : Daun Pisang yang terkena Hama Erionota Thrax L.)

**TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Program  
Studi Statistika



Disusun Oleh:

Ika Lailia Nur Rohmatun Nazila

20611106

**PROGRAM STUDI STATISTIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA  
2024**

**HALAMAN PERSETUJUAN PEMBIMBING  
TUGAS AKHIR**

Judul : Pengembangan Sistem Deteksi Penyakit Daun Pisang Untuk Meningkatkan Kualitas Ekspor Menggunakan Algoritma YOLOv5  
(Studi Kasus : Daun Pisang yang terkena Hama Erionota Thrax L.)

Nama Mahasiswa : Ika Lailia Nur Rohmatun Nazila

NIM : 20611106

**TUGAS AKHIR INI TELAH DIPERIKSA DAN DISETUJUI UNTUK  
DIUJIKAN**

Yogyakarta, 25 Januari 2024

Mengetahui

Menyetujui

Ketua Program Studi Statistika

Pembimbing

  
Dr. Atina Ahdika, S.Si., M.Si.

  
Dr. RB Fajriya Hakim, M.Si.

**HALAMAN PENGESAHAN**  
**TUGAS AKHIR**

**PENGEMBANGAN SISTEM DETEKSI PENYAKIT DAUN PISANG  
UNTUK MENINGKATKAN KUALITAS EKSPOR MENGGUNAKAN  
ALGORITMA YOLOv5**

(Studi Kasus : Daun Pisang yang terkena Hama Erionota Thrax L.)

Nama Mahasiswa : Ika Lailia Nur Rohmatun Nazila

NIM : 20611106

**TUGAS AKHIR INI TELAH DIUJIKAN  
PADA TANGGAL : 06 Februari 2024**

**Nama Penguji**

**Tanda Tangan**

1. Dina Tri Utari, S.Si., M.Sc.

2. Arum Handini Primandari, S.Pd.Si., M.Sc

3. Dr. Raden Bagus Fajriya Hakim, S.Si., M.Si.



Mengetahui,

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



(Prof. Riyanto, S.Pd., M.Si., Ph.D.)



## KATA PENGANTAR



*Assalamu 'alaikum Wr. Wb*

Puji dan syukur selalu penulis panjatkan kehadiran Allah SWT atas limpahan rahmat, taufiq, dan karunia-NYA sehingga penulis diberikan kesehatan dan kemudahan dalam menyelesaikan tugas akhir ini. Shalawat serta salam selalu penulis sampaikan kepada nabi akhir zaman, Rasulullah SAW beserta sahabat dan keluarganya yang membawa umat manusia dari zaman jahiliyah menuju zaman islamiyah. Tugas tugas akhir ini disusun sebagai syarat untuk mendapatkan gelar sarjana Program Studi Statistika, Universitas Islam Indonesia. Tugas akhir yang disusun ini berjudul “**Pengembangan Sistem Deteksi Penyakit Daun Pisang Untuk Meningkatkan Kualitas Ekspor Menggunakan Algoritma YOLOv5**” dengan studi kasus Daun Pisang yang terkena Hama Erionota Thrax L.

Dalam melakukan penyusunan tugas akhir ini, penulis memperoleh dukungan, bimbingan dan bantuan dari berbagai pihak sehingga tugas akhir ini dapat diselesaikan dengan baik dan tepat waktu. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terimakasih kepada :

1. Bapak Prof. Riyanto, S.Pd., M.Si., Ph.D. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Islam Indonesia.
2. Bapak Dr. Edy Widodo, S.Si., M.Si. selaku Ketua Jurusan Statistika Universitas Islam Indonesia.
3. Ibu Dr. Atina Ahdika, S.Si., M.Si. selaku Ketua Program Studi Statistika Universitas Islam Indonesia.
4. Bapak Dr. Raden Bagus Fajriya Hakim, S.Si., M.Si. selaku Dosen Pembimbing yang telah memberikan bimbingan dan saran selama proses penyusunan tugas akhir.
5. Bapak dan Ibu Dosen Program Studi Statistika Universitas Islam Indonesia yang telah memberikan wawasan dan ilmu pengetahuan kepada peneliti.
6. Kedua orang tua dan adik yang selalu memberikan doa dan dukungan kepada peneliti.

7. Teman – teman seperjuangan yang telah membantu dalam penyusunan tugas akhir ini.
8. Semua pihak yang telah membantu penulis yang tidak bisa disebutkan satu per satu.

Penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna, oleh sebab itu penulis mengharapkan kritik dan saran yang membangun dalam mencapai kesempurnaan tugas akhir ini. Penulis mengharapkan semoga tugas akhir ini dapat bermanfaat untuk menambah ilmu pengetahuan bagi pembaca.

*Wassalamu 'alaikum Wr.Wb*

Yogyakarta, 25 Januari 2024



Ika Lailia Nur Rohmatun Nazila

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERSETUJUAN PEMBIMBING TUGAS AKHIR.....	ii
HALAMAN PENGESAHAN TUGAS AKHIR .....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI .....	vi
DAFTAR TABEL .....	viii
DAFTAR GAMBAR.....	ix
DAFTAR LAMPIRAN .....	xi
PERNYATAAN .....	xii
INTISARI.....	xiii
ABSTRACT .....	xiv
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah .....	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian .....	3
1.5. Manfaat Penelitian .....	3
BAB II TINJAUAN PUSTAKA .....	5
2.1. Penelitian Terdahulu .....	5
BAB III LANDASAN TEORI.....	9
3.1. Pengertian Tanaman Pohon Pisang.....	9
3.2. Daun yang Terkena Hama Erionota Thrax L.....	9
3.3. Pengertian Citra.....	10
3.3.1 Pengertian Citra Digital.....	10
3.3.2 Jenis – Jenis Citra Digital .....	11
3.3.3 Pengolahan Citra .....	12
3.4. <i>Artificial Intelligence</i> (AI).....	12
3.5. <i>Machine Learning</i> .....	13
3.6. <i>Deep Learning</i> .....	13
3.7. <i>Convolutional Neural Network</i> (CNN) .....	14
3.8. <i>You Only Look Once</i> (YOLO).....	16
3.9. <i>Mean Average Precision</i> (mAP) .....	27
BAB IV METODOLOGI PENELITIAN.....	29
4.1. Populasi dan Sampel Penelitian .....	29
4.2. Tempat dan Waktu Penelitian .....	29
4.3. Definisi Operasional Variabel Penelitian.....	29
4.4. Alat dan cara organisir data.....	29
4.5. Metode Penelitian.....	30
BAB V HASIL DAN PEMBAHASAN .....	32
5.1. Dataset Daun Pisang .....	32
5.2. <i>Preprocessing</i> Dataset .....	33
5.3. <i>Training Data</i> .....	34
5.3.1 Pengujian Nilai <i>Epoch</i> , <i>Batch</i> dan Model YOLOv5 .....	35
5.4. Deteksi Objek dengan YOLOv5 .....	46
5.5. Pengujian Performa Pada Dataset .....	46

BAB VI PENUTUP.....	49
6.1. Kesimpulan .....	49
6.2. Saran.....	49
DAFTAR PUSTAKA.....	50
LAMPIRAN .....	54

## DAFTAR TABEL

<b>Tabel 2.1</b> Tabel Penelitian Sebelumnya .....	5
<b>Tabel 4.1</b> Tabel Variabel Penelitian .....	29
<b>Tabel 5.1</b> Hasil Pengujian Tiga Parameter .....	36
<b>Tabel 5.4</b> Hasil Pengujian Performa pada Dataset .....	46



## DAFTAR GAMBAR

<b>Gambar 3.1</b> Daun yang Terkena Hama Erionota Thrax L. (Aeni, 2022).....	9
<b>Gambar 3.2</b> Pemecahan Gambar 77 Bagian (Geitgey, 2016) .....	14
<b>Gambar 3.3</b> <i>Input</i> kedalam <i>Small Neural Network</i> (Geitgey, 2016) .....	15
<b>Gambar 3.4</b> Proses Pembentukan <i>Array</i> Baru (Geitgey, 2016).....	15
<b>Gambar 3.5</b> Proses <i>Max Pooling</i> (Geitgey, 2016).....	16
<b>Gambar 3.6</b> Sistem Kerja CNN (Geitgey, 2016).....	16
<b>Gambar 3.7</b> Arsitektur Jaringan pada YOLO (Nisa, 2023).....	17
<b>Gambar 3.8</b> Proses Pembagian Grid dan Probabilitas Kelas (Luthfi, 2021) .....	17
<b>Gambar 3.9</b> Proses Perhitungan Nilai pada Sel Grid (Luthfi, 2021).....	18
<b>Gambar 3.10</b> <i>Intersection Over Union</i> (Cheng et al., 2019) .....	19
<b>Gambar 3.11</b> <i>Bounding Box</i> YOLO (Zhang Z. , 2020).....	20
<b>Gambar 3.12</b> Perhitungan Tensor Dua Kelas Sama (Luthfi, 2021) .....	20
<b>Gambar 3.13</b> Perhitungan <i>Bounding Box</i> Kelas Sama (Luthfi, 2021) .....	21
<b>Gambar 3.14</b> Perhitungan <i>Bounding Box</i> Kelas Sama (Luthfi, 2021) .....	21
<b>Gambar 3.15</b> Matriks <i>Bounding Box</i> (Luthfi, 2021) .....	21
<b>Gambar 3.16</b> Perhitungan <i>Bounding Box</i> Kelas Berbeda (Luthfi, 2021).....	21
<b>Gambar 3.17</b> Perhitungan <i>Bounding Box</i> Kelas Berbeda (Luthfi, 2021).....	22
<b>Gambar 3.18</b> Perhitungan <i>Bounding Box</i> Kelas Berbeda (Luthfi, 2021).....	22
<b>Gambar 3.19</b> Matriks <i>Bounding Box</i> (Luthfi, 2021) .....	22
<b>Gambar 3.20</b> Kelas Berdasar Pada Skor (Luthfi, 2021).....	23
<b>Gambar 3.21</b> Pendaftaran Semua <i>Bounding Box</i> (Luthfi, 2021).....	23
<b>Gambar 3.22</b> Perhitungan IoU <i>Bounding Box</i> (Luthfi, 2021) .....	23
<b>Gambar 3.23</b> Perhitungan IoU pada <i>Bounding Box</i> (Luthfi, 2021).....	24
<b>Gambar 3.24</b> Hasil Akhir Pendeteksian YOLO (Luthfi, 2021).....	24
<b>Gambar 3.24</b> Arsitektur Jaringan YOLOv5 (Septyanto, Wibowo, & Setianingsih, 2022) .....	25
<b>Gambar 3.25</b> Tipe dan Performa YOLOv5 (Cochard, 2021).....	25
<b>Gambar 3.26</b> Tipe dan Performa YOLO (Cochard, 2021).....	26
<b>Gambar 3.27</b> Ilustrasi Perhitungan <i>Average Precision</i> (Kukit, 2022) .....	28
<b>Gambar 3.28</b> Ilustrasi Perhitungan <i>Mean Average Precision</i> (Kukit, 2022) .....	28

<b>Gambar 4.1</b> Diagram alir Penelitian .....	30
<b>Gambar 5.1</b> <i>Frame</i> Citra Hasil Konversi Video.....	32
<b>Gambar 5.2</b> Pelabelan Objek pada Gambar .....	33
<b>Gambar 5.3</b> Hasil file “.txt” Pelabelan Objek .....	34
<b>Gambar 5.4</b> Hasil Validasi Model YOLOv5s .....	36
<b>Gambar 5.5</b> Grafik Kinerja Model YOLOv5s.....	37
<b>Gambar 5.6</b> Grafik Nilai <i>F1 Score</i> Terhadap Nilai <i>Confidence</i> .....	38
<b>Gambar 5.7</b> Grafik Nilai <i>Precision</i> Terhadap Nilai <i>Confidence</i> .....	38
<b>Gambar 5.8</b> Grafik Nilai <i>Recall</i> Terhadap Nilai <i>Confidence</i> .....	39
<b>Gambar 5.9</b> Grafik Nilai <i>Precision</i> Terhadap Nilai <i>Recall</i> .....	39
<b>Gambar 5.10</b> Hasil <i>Training</i> Model YOLOv5m .....	40
<b>Gambar 5.11</b> Grafik Kinerja Model YOLOv5m .....	40
<b>Gambar 5.12</b> Grafik Nilai <i>F1 Score</i> Terhadap Nilai <i>Confidence</i> .....	41
<b>Gambar 5.13</b> Grafik Nilai <i>Precision</i> Terhadap Nilai <i>Confidence</i> .....	41
<b>Gambar 5.14</b> Grafik Nilai <i>Recall</i> Terhadap Nilai <i>Confidence</i> .....	42
<b>Gambar 5.15</b> Grafik Nilai <i>Precision</i> Terhadap Nilai <i>Recall</i> .....	42
<b>Gambar 5.16</b> Hasil <i>Training</i> Model YOLOv5x .....	43
<b>Gambar 5.17</b> Grafik Kinerja Model YOLOv5x .....	43
<b>Gambar 5.18</b> Grafik Nilai <i>F1 Score</i> Terhadap Nilai <i>Confidence</i> .....	44
<b>Gambar 5.19</b> Grafik Nilai <i>Precision</i> Terhadap Nilai <i>Confidence</i> .....	44
<b>Gambar 5.20</b> Grafik Nilai <i>Recall</i> Terhadap Nilai <i>Confidence</i> .....	45
<b>Gambar 5.21</b> Grafik Nilai <i>Precision</i> Terhadap Nilai <i>Recall</i> .....	45

## DAFTAR LAMPIRAN

Lampiran 1 .....	54
------------------	----

## PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya karya yang sebelumnya pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali yang diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 25 Januari 2024



Ika Lailia Nur Rohmatun Nazila

## INTISARI

### **PENGEMBANGAN SISTEM DETEKSI PENYAKIT DAUN PISANG UNTUK MENINGKATKAN KUALITAS EKSPOR MENGGUNAKAN ALGORITMA YOLOv5**

(Studi Kasus : Daun Pisang yang terkena Hama Erionota Thrax L.)

Ika Lailia Nur Rohmatun Nazila

Program Studi Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Islam Indonesia

Indonesia merupakan negara agraris yang memiliki kekayaan pada sektor pertanian dan perkebunan. Salah satu aspek yang menjadi fokus utama dalam penelitian ini adalah tanaman pohon pisang. Tanaman pohon pisang adalah salah satu tanaman tropis yang melimpah di Indonesia. Selain produksi buah yang menjadi sumber pangan, daun pisang juga menjadi salah satu komoditas utama di bidang pertanian yang sangat signifikan dalam perkembangan ekonomi. Untuk memenuhi standar mutu dan kriteria yang ditetapkan di negara tujuan ekspor, kualitas produk sangat penting dalam mempertahankan pasar dan tingkat pendapatan. Namun, salah satu tantangan utama dalam budidaya pisang adalah serangan hama, seperti Erionota Thrax L. yang seringkali mengganggu dan dapat mengurangi hasil serta kualitas panen. Penerapan metode konvensional membutuhkan waktu dan sumber daya yang besar. Untuk mengatasi permasalahan tersebut, diperlukan sebuah sistem deteksi yang dapat melakukan klasifikasi dengan cepat dan akurat. Salah satu metode *deep learning* yang dapat digunakan yaitu *You Only Look Once* versi 5 (YOLOv5). Algoritma YOLOv5 memiliki kemampuan pengenalan objek dalam gambar dan video yang dapat diklasifikasikan secara *real-time*. Data yang digunakan dalam penelitian ini adalah data primer sebanyak 407 gambar yang dilatih dengan beberapa *hyperparameter* yang berbeda diantaranya *epoch*, *batch size*, *optimizer* dan model YOLOv5. Hasil analisis menunjukkan bahwa pengujian terbaik pada nilai *epoch 50*, *batch size 8*, *optimizer Stochastic Gradient Descent (SGD)*, model YOLOv5x dengan nilai mAP sebesar 0,867. Dengan demikian, petani dapat mengambil tindakan pencegahan yang tepat secara dini, meningkatkan kualitas produk dan hasil panen, dan memberikan dukungan kuat bagi keberlanjutan dan kesejahteraan petani di sektor industri pisang.

**Kata Kunci** : Daun Pisang, Erionota Thrax L., YOLOv5.

## ABSTRACT

### **Development of a Banana Leaf Disease Detection System to Improve Export Quality Using the YOLOv5 Algorithm**

(Case Study : Banana Leaves Affected by Erionota Thrax L. Pest)

Ika Lailia Nur Rohmatun Nazila

Department of Statistics, Faculty of Mathematics and Natural Sciences  
Universitas Islam Indonesia

*Indonesia is an agrarian country rich in agricultural and plantation resources. One of the main focuses of this research is the Banana Tree Plant. The Banana Tree is one of the abundant tropical plants in Indonesia. Besides its fruit production as a food source, banana leaves also serve as a significant commodity in the agricultural sector, contributing to economic development. To meet the quality standards and criteria set by export destination countries, product quality is crucial for maintaining market access and income levels. However, one of the primary challenges in banana cultivation is pest attacks, such as Erionota Thrax L., which often disturb and can reduce both yield and harvest quality. The implementation of conventional methods requires significant time and resources. To overcome these challenges, a detection system capable of fast and accurate classification is necessary. One deep learning method that can be utilized is You Only Look Once version 5 (YOLOv5). The YOLOv5 algorithm has the ability to recognize objects in images and videos that can be classified in real-time. The data used in this research consists of 407 primary images trained with various hyperparameters, including epoch, batch size, optimizer, and the YOLOv5 model. The analysis results indicate that the best testing results were achieved with an epoch value of 50, batch size of 8, Stochastic Gradient Descent (SGD) optimizer, and YOLOv5x model, with a mean Average Precision (mAP) value of 0.867. Thus, farmers can take timely preventive measures, enhance product quality and harvest yields, providing robust support for the sustainability and well-being of farmers in the banana industry.*

**Keywords:** *Banana Leaves, Erionota Thrax L., YOLOv5.*

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang Masalah**

Pisang (*Musa spp.*) merupakan salah satu tanaman tropis yang berperan penting dalam perekonomian global dan pangan di berbagai negara. Tanaman pohon pisang, memiliki banyak manfaat diantaranya adalah sebagai sumber buah, sayur, obat Antiacne, kosmetik, bahan kerajinan dan pembungkus makanan. Saat ini, produksi buah pisang dan daun pisang dapat menunjang mata pencaharian jutaan petani dan menjadi salah satu komoditas ekspor utama (Tropika, 2014).

Demi memenuhi standar mutu dan kualitas yang ditetapkan oleh negara-negara tujuan ekspor, daun pisang yang akan diekspor juga harus memenuhi kriteria tertentu, termasuk jenis pohon dan ukuran daun. Jenis pohon yang diminati oleh pasar ekspor diantaranya adalah Pisang Raja, Batu, Kapok, Cavendish, Awak, Mas, Tanduk, Abaca, Petung dan Salahat. Sedangkan, ukuran daun harus memiliki minimal panjang 100 cm dan lebar 35 cm. Ekspor daun pisang memiliki potensi ekonomi yang signifikan, terutama menuju negara seperti Jepang dan Amerika Serikat, Harga satu lembar daun pisang di Jepang berkisar antara 278.000 hingga 321.000 rupiah. Sedangkan, di negara Amerika Serikat dijual dengan harga 195.000 hingga 270.000 rupiah (Kintari, 2023).

Produksi pisang berkelanjutan akan dihadapkan dengan beberapa tantangan salah satunya adalah hama pada daun pisang yang dapat menurunkan hasil dan kualitas panen. Salah satu penyakit yang sering ditemukan pada daun pisang adalah Hama Erionota Thrax L. Hama Erionota Thrax L. (*Odoiporus longicolis*) merupakan hama utama pada tanaman pisang. Hama ini menyerang daun pisang yang berakibat tanaman menjadi gundul dan hanya terlihat tulang daunnya. Larva dengan warna hijau muda yang dilapisi tepung berwarna putih dengan panjang sekitar 7 cm akan memotong miring tepi daun dan menggulungnya berbentuk tabung kecil. Dalam gulungan tersebut, larva akan memakan daun hingga habis. Hama ini dapat menyebabkan penurunan kualitas daun pisang, sehingga diperlukan penanganan yang baik agar dapat meminimalisir dampak buruk terhadap produktivitas tanaman dan perekonomian petani (Subari et al., 2022).

Secara konvensional, terdapat berbagai macam metode yang dapat dilakukan untuk pemeriksaan penyakit pada tanaman pohon pisang, diantaranya adalah pengamatan langsung terhadap tanaman oleh petani dan pemeriksaan laboratorium yang dilakukan oleh para peneliti. Metode tersebut membutuhkan biaya yang besar dan jangka waktu yang lama serta hasil pengamatan tidak dapat langsung diidentifikasi serta semakin lama mengidentifikasi maka hama semakin menyebar (Andera et al., 2021). Berdasarkan keterbatasan dari metode sebelumnya, maka diperlukan teknologi untuk dapat mendeteksi daun pisang yang terkena penyakit dengan cepat, otomatis, dan efisien.

Kombinasi *Artificial Intelligence* dan *Computer Vision* telah mengalami perkembangan yang pesat khususnya pada *Machine Learning* yang menjadi cikal bakal *Deep Learning* (DL). Implementasi metode DL salah satunya adalah *Convolutional Neural Network* (CNN) yang menjadi solusi dalam menyelesaikan masalah di berbagai bidang, khususnya agrikultur. Dalam melakukan deteksi objek, algoritma *You Only Look Once* (YOLO) menjadi metode terunggul dari metode lainnya dengan melakukan kalkulasi *Bounding Box* dengan tingkat akurasi yang tinggi dan efisiensi yang baik. YOLOv5 adalah salah satu versi dari algoritma YOLO yang menghadirkan peningkatan signifikan dalam hal akurasi dan kecepatan deteksi objek. Pemilihan YOLOv5 sebagai algoritma deteksi objek dalam penelitian ini didasarkan pada kemampuannya untuk memberikan hasil yang akurat dalam waktu singkat (Wibowo et al., 2023).

Dalam konteks ini, penelitian ini akan fokus pada pengembangan dan evaluasi sistem deteksi daun pisang yang terkena penyakit menggunakan algoritma YOLOv5, dengan harapan bahwa hasil penelitian ini akan membantu dalam meningkatkan budidaya dan ekspor daun pisang serta mengurangi kerugian yang disebabkan oleh penyakit pada tanaman ini. Dengan memahami dampak penyakit daun pisang yang disebabkan oleh Hama Erionota Thrax L. dan solusi teknologi yang dapat diterapkan, penelitian ini bertujuan untuk memberikan kontribusi positif pada industri pertanian dan perdagangan internasional.



## **1.2. Rumusan Masalah**

Berdasarkan penjabaran dari latar belakang di atas, maka rumusan masalah yang akan dikaji dari penelitian ini yaitu:

1. Bagaimana implementasi sistem dalam melakukan deteksi kerusakan akibat Hama Erionota Thrax L pada daun pisang?
2. Bagaimana tingkat akurasi sistem dalam mendeteksi daun pisang yang terkena kerusakan Hama Erionota Thrax L.?

## **1.3. Batasan Masalah**

Agar menghindari perluasan topik bahasan, batasan masalah yang diterapkan dalam penelitian ini adalah sebagai berikut:

1. Data yang digunakan yaitu citra Tanaman Pisang (*Musa spp.*) dengan menggunakan kamera HP Xiaomi Redmi Note 10S.
2. Deteksi penyakit berfokus pada Hama E. Thrax L. (*Odoiporus longicollis*).
3. Wilayah geografis terbatas pada Provinsi Jawa Tengah.
4. Analisis pengolahan gambar menggunakan YOLOv5.
5. Deteksi dilakukan pada daun pisang dengan ukuran minimal panjang 100 cm dan lebar 35 cm, sesuai dengan persyaratan ekspor.

## **1.4. Tujuan Penelitian**

Berdasarkan rumusan masalah, tujuan dari penelitian tugas akhir ini adalah sebagai berikut :

1. Membuat sistem deteksi otomatis untuk mengklasifikasi penyakit pada daun pisang yang terinfeksi Hama Erionota Thrax L. menggunakan algoritma YOLOv5.
2. Mengevaluasi tingkat akurasi sistem deteksi yang dikembangkan dalam mengklasifikasi daun pisang yang terinfeksi Hama Erionota Thrax L. dibandingkan dengan metode konvensional.
3. Menganalisis dampak penggunaan sistem klasifikasi dalam meningkatkan ekspor daun pisang.

## **1.5. Manfaat Penelitian**

Manfaat yang dapat diperoleh dari penelitian ini adalah sebagai berikut :

1. Penelitian ini diharapkan dapat meningkatkan kualitas dan daya saing produk ekspor daun pisang yang bebas dari penyakit
2. Membantu petani dalam mendeteksi penyakit dini pada daun pisang sehingga dapat mengambil tindakan pencegahan yang tepat dan mengurangi kerugian hasil panen
3. Dapat menjadi referensi pada penelitian selanjutnya terkait dengan *object detection* menggunakan algoritma YOLOv5.

## BAB II

### TINJAUAN PUSTAKA

Penelitian terdahulu memiliki peran penting sebagai panduan dan sumber informasi yang dapat memberikan pemahaman mendalam tentang teori-teori yang terkait dengan *object detection* serta metode-metode yang diterapkan dalam penelitian. Penelitian yang digunakan sebagai referensi dalam penelitian ini adalah penelitian-penelitian sebelumnya yang telah mengkaji *object detection* menggunakan YOLO dalam bidang agrikultur. Dalam konteks ini, berikut adalah beberapa studi yang telah menjadi sumber referensi dalam penelitian ini.

#### 2.1. Penelitian Terdahulu

**Tabel 2.1** Tabel Penelitian Sebelumnya

<b>Tahun</b>	<b>Nama</b>	<b>Judul</b>	<b>Hasil Penelitian</b>
2022	Prya Artha Widjaja, Jose Ryu Leonesta	<i>Determining Mango Plant Types Using YOLOv4</i>	Hasil dari penelitian diperoleh deteksi jenis daun mangga menunjukkan tingkat akurasi yang tinggi, dengan Mangga Harum Manis mencapai 99.43%, Mangga Apel sebesar 98.92%, Mangga Golek mencapai 99.34%, dan Mangga Manalagi mencapai 97.45%. Untuk daun yang bukan Tanaman Pohon Mangga, tingkat akurasi deteksinya mencapai 95.45%.
2023	Md. JanibulAlam Soeb, Md. Fahad Jubayer, TahminaAkanjeeTarin, Muhammad RashedAl Mamun, Fahim Mahafuz Ruhad, Aney Parven Nabisab Mujawar Mubarak, Soni Lanka Karri & Islam Md. Meftaul	<i>Tea leaf disease detection and identification based onYOLOv7 (YOLO-T)</i>	Hasil dari penelitian didapatkan bahwa dari lima jenis penyakit pada Tanaman Teh diperoleh akurasi Black rot 97%, Brown blight 97%, Leaf rust 97%, Red spider 98%, Tea mosquito 97%.
2023	Sajitha P, Alwin John, V L Devika, Gayathri S V, Nafla Sakhir	<i>Leaf Disease Detection &amp; Correction using YOLO V7 with GPT3 integrated</i>	Hasil dari penelitian adalah dari gambar tanaman dengan berbagai jenis penyakit yaitu early leaf blight, karat, dan bercak

Tahun	Nama	Judul	Hasil Penelitian
			pada daun seperti cercospora dan tikka diperoleh bahwa hasil akurasi mencapai 96%.
2022	Zhaoyi Chen, Ruhui Wu, Yiyang Lin, Chuyu Li, Siyu Chen, Zheneng Yuan, Shiwei Chen and Xiangjun Zou	<i>Plant Disease Recognition Model Based on Improved YOLOv5</i>	Hasil dari penelitian diperoleh bahwa dari dataset penyakit Tanaman Pohon Karet dengan YOLOv5 tingkat presisi sebesar 86,5% untuk deteksi serbuk sari (powdery mildew) dan 86,8% untuk deteksi antraknosa. mean average precision mencapai 70%, yang merupakan peningkatan sebesar 5,4% dibandingkan dengan YOLOv5 asli
2023	Mahnoor Khalid, Muhammad Shahzad Sarfraz, Uzair Iqbal, Muhammad Umar Aftab, Gniewko Niedbala, and Hafiz Tayyab Rauf	<i>Real-Time Plant Health Detection Using Deep Convolutional Neural Networks</i>	Hasil penelitian dengan menggunakan dataset Daun Paprika dan Kentang, Model YOLOv5 mampu mencapai akurasi 93% untuk melakukan deteksi daun sehat dan tidak sehat.
2021	Rahellea Andera, Priyatiningih, Mutiara Mutiara	Deteksi dan Identifikasi Penyakit Tanaman Pisang (Musa Acuminata) oleh Cendawan Cordana Musae dengan Beberapa Metode	Hasil penelitian didapatkan dengan menggunakan beberapa metode yang dilakukan terdapat cendawan yaitu bercak daun (Cordana musae).
2022	Wulandari Subari, Nureny Goo, Victor G. Siahaya.	Kerusakan Tanaman Pisang Oleh Hama Ulat Penggulung Daun (Erionota thrax L) Di Pulau Ambon	Hasil penelitian diperoleh bahwa intensitas Hama Erionota Thrax L. pada tiga lokasi berbeda yaitu Dusun Tulehu memiliki jumlah gulungan daun sebanyak 300 dengan kerusakan sebesar 63.14% (berat), Dusun Keranjang memiliki 200 gulungan daun dengan intensitas kerusakan sebesar 59.36% (berat) dan Desa Tawiri memiliki 187

Tahun	Nama	Judul	Hasil Penelitian
			gulungan daun dengan intensitas kerusakan sebesar 39.11% (sedang).
2022	Putu Yudhi Artana, Ni Ketut Dewi Ari Jayanti, I Made Arya Budhi Saputra.	Sistem Pakar Diagnosa Penyakit dan Hama Tanaman Pisang Menggunakan Metode Forward Chaining Berbasis Android	Hasil penelitian menunjukkan bahwa sistem pakar mampu mengidentifikasi penyakit dan hama pada tanaman pisang secara efektif.
2018	Suci Nurjanah	Dampak Ekspor Pisang Indonesia ke Jepang dalam Rangka Indonesia-Japan Economic Partnership Agreement (Ijepa)	Hasil penelitian menunjukkan bahwa ekspor pisang ini memberikan dampak ekonomi dan politik yang signifikan pada kedua negara. Indonesia mendapatkan keuntungan dengan meningkatnya pendapatan petani pisang, pertumbuhan produk domestik bruto, peningkatan neraca perdagangan, dan perluasan pasar produk pisang ke Jepang. Jepang juga menguntungkan sebagai tujuan utama ekspor pisang dari Indonesia, yang membantu meningkatkan aksesnya ke Indonesia.
2023	Lusiana Lusiana, Ari Wibowo, dan Tika Kartika Dewi	Implementasi Algoritma <i>Deep Learning You Only Look Once</i> (YOLOv5) Untuk Deteksi Buah Segar dan Busuk	Hasil penelitian diperoleh bahwa dalam pendeteksian jenis buah segar dan busuk menunjukkan tingkat akurasi yang tinggi yaitu 84% dengan nilai precision rata-rata mencapai 0,990 dan recall mencapai puncak nilai 1,00 pada tingkat kepercayaan 0,00

Berdasarkan penelitian sebelumnya yang terlampir pada **Tabel 2.1** metode YOLO telah diterapkan dengan berbagai data yang berbeda. Sehingga, peneliti memutuskan melakukan penelitian dengan mengimplementasikan algoritma

YOLOv5 dengan pendekatan yang sangat spesifik terhadap deteksi gejala serangan Hama Erionota Thrax L. pada daun pisang, terutama dalam konteks budidaya pisang untuk tujuan ekspor. Penelitian ini yang menjadi pembeda dari penelitian sebelumnya, dimana deteksi Hama Erionota Thrax L. belum banyak dieksplorasi.

## **BAB III**

### **LANDASAN TEORI**

#### **3.1. Pengertian Tanaman Pohon Pisang**

Tanaman pohon pisang adalah salah satu famili *Musaceae* yang berasal dari Asia Tenggara. Pohon pisang ini cocok ditanam di daerah tropis dan merupakan tanaman budidaya yang tidak musiman namun dapat berbuah sepanjang tahun (Paath et al., 2018). Pohon pisang adalah salah satu kekayaan Indonesia dengan nama latin *Musa sp*, yang memiliki keanekaragaman spesies diantaranya pisang cavendish, pisang raja, pisang kepok, pisang ambon, pisang abaka, pisang susu, pisang nangka, dan lain sebagainya. Pohon pisang memiliki batang yang tumbuh secara berkelompok dalam rumpun dengan daun lebar, helai berbentuk lonjong yang terkadang memiliki lapisan lilin dan tangkai daun panjang serta membentuk batang semu. Tanaman pohon pisang memiliki bunga majemuk dengan masing – masing kuncup bunga terbungkus oleh seludang berwarna merah kecoklatan. Buah pisang sering dimakan secara langsung, dikukus, atau diolah menjadi camilan seperti pisang selai, keripik, ataupun dikolak (Indra Putra, 2018a).

#### **3.2. Daun yang Terkena Hama Erionota Thrax L.**

Daun yang terkena Hama Erionota Thrax L. akan menyebabkan tanaman menjadi gundul dan hanya terlihat tulang daunnya. Ketika larva keluar dari telurnya, akan memotong helaian daun dari samping dan menggulungnya hingga daun tersebut mengering dan robek sehingga akan menurunkan produktivitas tanaman pohon pisang.



**Gambar 3.1** Daun yang Terkena Hama Erionota Thrax L. (Aeni, 2022)

Erionota Thrax L. merupakan salah satu hama utama yang menyerang tanaman pisang khususnya pada bagian daun pisang. Hama tersebut termasuk anggota kupu – kupu dalam famili *Hesperiidae* yang berasal dari India, Asia Tenggara hingga Papua Nugini. Erionota Thrax L. memiliki sayap berwarna coklat dan bercak kuning pada sepasang sayap bagian depan. Sedangkan ketika menjadi ulat, kepalanya berwarna hitam dengan badan berwarna putih, gilig, kecil di bagian depan dan belakang, bergaris tengah dengan ukuran 8 mm serta panjangnya 6 cm. Erionota Thrax L. adalah hama yang paling umum dijumpai dengan tingkat serangan tertinggi dibandingkan dengan hama lainnya, Tahapan destruktif hama ini pada tahap larva. (Indra Putra, 2018a).

### **3.3. Pengertian Citra**

Citra merupakan representasi dua dimensi untuk bentuk fisik tiga dimensi secara nyata. Citra diartikan sebagai suatu fungsi dua dimensi  $f(x,y)$  dimana  $x$  dan  $y$  adalah koordinat  $(x,y)$  yang dinamakan intensitas. Dalam perwujudannya, citra sangat beragam mulai dari gambar berwarna putih pada foto (yang tidak bergerak) hingga pada gambar berwarna yang bergerak pada televisi. Citra terdiri dari dua macam yaitu citra analog dan citra digital. Citra digital merupakan citra  $f(x,y)$  yang dilakukan digitalisasi pada segi koordinat dan nilai intensitasnya. Pada satu citra digital tersusun dari sejumlah elemen yang dinamakan *picture element* atau pixel. Sedangkan, citra analog merupakan citra yang bersifat *continue* seperti gambar pada televisi, foto sinar X, dan lain sebagainya (Sri Ratna Sulistiyanti et al., 2016).

#### **3.3.1 Pengertian Citra Digital**

Citra digital adalah susunan *array* yang berisi nilai – nilai *real* atau kompleks yang diwakili oleh urutan bit tertentu. Pada komputer, gambar digital diubah menjadi kumpulan elemen berbentuk kotak kecil yang disebut *pixel*. Masing – masing *pixel* tersusun dari angka yang merepresentasikan saluran warna. Nilai dalam piksel disimpan secara terurut oleh komputer dan sering kali diubah atau dikompres untuk keperluan pengolahan. Sebuah citra dapat didefinisikan sebagai sebuah fungsi  $f(x,y)$  dengan ukuran  $M$  baris dan  $N$  kolom. Dimana  $x$  dan  $y$  adalah koordinat spasial dan amplitude  $f$  pada titik koordinat  $(x,y)$  yang disebut intensitas atau tingkat keabuan bayangan pada titik tersebut. Jika nilai  $x$ ,  $y$ , dan amplitude  $f$



secara keseluruhan bernilai diskrit dan berhingga. Oleh karena itu, citra dapat ditulis sebagai dalam matriks sebagai berikut:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-0,1) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (3.1)$$

Berdasarkan formula di atas, sebuah citra  $f(x,y)$  dapat dituliskan ke dalam fungsi matematis sebagai berikut:

$$\begin{aligned} 0 \leq x \leq M - 1 \\ 0 \leq y \leq N - 1 \\ 0 \leq f(x,y) \leq G - 1 \end{aligned}$$

Dimana :

M : Jumlah baris pada citra

N : Jumlah Kolom pada citra

G : Nilai pada skala *grayscale*

Dalam menentukan besarnya nilai M, N, dan G dapat menggunakan persamaan dibawah ini :

$$M = 2^m; N = 2^n; G = 2^k \quad (3.2)$$

Dimana nilai m, n, dan k adalah bilangan positif dengan interval (0,G) yang dinamakan *greyscale*. Besarnya nilai g bergantung pada proses digitalisasinya. Angka 0 pada *greyscale* menyatakan warna hitam dan angka 1 menyatakan warna putih (Sri Ratna Sulistiyanti et al., 2016).

### 3.3.2 Jenis – Jenis Citra Digital

Citra digital terbagi dalam tiga bagian diantaranya citra biner, *grayscale*, dan warna.

1. Citra Biner merupakan citra yang hanya memiliki dua nilai warna yaitu hitam dan putih, Untuk melakukan penyimpanan pada kedua warna tersebut diperlukan memori sebesar 1 bit.
2. Citra *Greyscale* merupakan citra dengan warna bertingkat abu. Citra *greyscale* mewakili nilai dari masing – masing pixel dengan nilai tunggal, sehingga piksel terdiri dari warna merah, hijau, dan biru dengan intensitas sama. Intensitas pada citra *greyscale* berkisar antara 0 hingga 255. Nilai 0 merepresentasikan warna hitam dan nilai 255 merepresentasikan warna putih.

3. Citra Warna adalah citra yang terdiri dari warna yang dicampur. Pada masing – masing warna *pixel* merupakan kombinasi dari tiga warna RGB (*red, green, blue*). Setiap warna dasar membutuhkan memori sebesar 8 bit. Masing – masing warna mempunyai gradasi sebesar 255 warna dengan artian bahwa setiap *pixel* terdiri dari warna yang dikombinasi lebih dari 16 juta warna (Sri Ratna Sulistiyanti et al., 2016).

### 3.3.3 Pengolahan Citra

Pengolahan citra adalah sebuah disiplin ilmu yang mempelajari tentang perbaikan kualitas gambar yang meliputi transformasi warna, peningkatan kontras dan restorasi citra). Pengolahan citra berarti sebuah proses memasukkan citra dan menghasilkan *output* berupa citra yang dikehendaki. Pengolahan citra dapat dilakukan menggunakan komputer untuk memperoleh hasil yang lebih baik. Pengolahan citra memiliki tujuan untuk melakukan perbaikan citra sehingga dapat diinterpretasi oleh manusia atau sebuah mesin komputer (Jumadi & Sartika, 2021)

### 3.4. Artificial Intelligence (AI)

*Artificial intelligence* atau kecerdasan buatan merupakan sebuah sistem komputer yang mampu melakukan tugas – tugas manusia. Teknologi ini membuat keputusan dengan melakukan analisis data yang ada didalam sistem. Proses yang terdapat dalam *artificial intelligence* diantaranya adalah *learning, reasoning, dan self – correction*. Menurut John Mc Carthy (1956) *artificial intelligence* memiliki fungsi untuk mengetahui dan melakukan pemodelan proses berpikir manusia dan mendesain mesin sehingga dapat mengikuti perilaku manusia.

*Artificial intelligence* (AI) adalah sebuah inovasi terbaru dalam perkembangan ilmu pengetahuan. Kecerdasan buatan bermula sejak munculnya komputer modern pada tahun 1940. Pada tahun 1950, AI merupakan sebuah komputer yang sudah dapat menjalankan tugas yang dikerjakan oleh manusia dan dimanfaatkan untuk bermain game. Kecerdasan buatan tersebut jika secara teknik hanya terbatas pada suatu alat, namun secara teknologi penerapannya lebih luas. Permasalahan mengenai kecerdasan buatan saling berhubungan terhadap kondisi sosial budaya masyarakat sekitar dan kebijakan dari pemerintah dan industri telekomunikasi (Ahmad, 2017).

### **3.5. Machine Learning**

*Machine learning* merupakan sebuah aplikasi komputer dan algoritma matematika yang diadaptasi dengan cara mempelajari data yang akan menghasilkan prediksi dimasa depan. Proses pembelajaran didefinisikan sebagai sebuah usaha untuk memperoleh kecerdasan melalui dua tahapan diantaranya adalah pelatihan (*training*) dan pengujian (*testing*). *Machine learning* selalu berkaitan dengan pertanyaan mengenai cara pembangunan program komputer agar dapat meningkat secara otomatis berdasarkan pengalaman yang dilalui. *Machine learning* terbagi menjadi tiga kategori diantaranya adalah *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning*.

- a) *Supervised Learning* merupakan metode klasifikasi dimana data yang digunakan dalam pelatihan diberikan label / kategori yang menunjukkan kelas atau kelompoknya. Model yang didapatkan dapat digunakan untuk memprediksi label atau kelas dari data baru yang belum pernah dilihat sebelumnya.
- b) *Unsupervised Learning* sering disebut sebagai *cluster* yang tidak membutuhkan labelisasi pada kumpulan data dan hasilnya tidak mengidentifikasikan kelas yang ditentukan.
- c) *Reinforcement Learning* adalah pendekatan pembelajaran yang fokus pada bagaimana suatu entitas memutuskan tindakan apa yang harus diambil dalam situasi tertentu untuk mencapai hasil optimal, serta bagaimana cara mengaitkan situasi tersebut dengan aksi yang tepat. (Roihan et al., 2020).

### **3.6. Deep Learning**

*Deep learning* merupakan sebuah teknik dalam pembelajaran mesin yang mengarahkan sebuah sistem komputer untuk mempelajari situasi terkait dengan pemrograman tertentu dengan tujuan mesin tersebut mampu bekerja secara natural selayaknya manusia. *Deep learning* merupakan perkembangan dari teknologi kecerdasan buatan. Dalam *deep learning*, mesin komputer akan mempelajari berbagai model dan melakukan klasifikasi tugas – tugas yang diberikan berdasarkan data yang dikumpulkan. Data yang *diinput* dapat berupa teks, gambar maupun suara. *Deep learning* mampu mengolah data dengan jumlah yang besar dengan akurasi yang sangat tinggi.

*Deep Learning* merupakan teknik dalam *neural network* yang menggunakan sebuah cara tertentu seperti *Restricted Boltzmann Machine (RBM)* untuk melakukan percepatan pada proses pembelajaran dalam *neural network* yang menggunakan lapisan lebih dari tujuh lapis. Dengan adanya *deep learning*, waktu yang diperlukan untuk melakukan *training* akan semakin efektif karena masalah hilangnya gradien pada propagasi balik akan semakin rendah. *Deep learning* sering dimanfaatkan pada riset dan industri diantaranya seperti *Computer Vision*, *Natural Language Processing*, dan *Speech Recognition* (Mellyssa et al., 2022).

### 3.7. *Convolutional Neural Network (CNN)*

*Convolutional Neural Network (CNN)* merupakan perkembangan dari *Multilayer Perceptron (MLP)* khususnya untuk pemrosesan data berdimensi dua. *CNN* dikategorikan sebagai *deep neural network* karena kedalaman yang besar dan sering digunakan dalam analisis data gambar. Dalam hal klasifikasi gambar, *MLP* kurang cocok digunakan karena tidak mampu mempertahankan informasi spasial dari data gambar, di mana setiap piksel dianggap sebagai fitur independen. Hal ini mengakibatkan hasil yang kurang akurat. *CNN* bekerja mirip dengan *MLP*, tetapi setiap neuron dalam *CNN* direpresentasikan dalam dua dimensi, berbeda dengan *MLP* di mana setiap neuron hanya memiliki satu dimensi. (I Wayan Suartika E. P et al., 2016). *CNN* memiliki beberapa langkah pengerjaan diantaranya sebagai berikut:

1. Melakukan pemecahan gambar menjadi lebih kecil

Dalam proses ini, gambar yang utuh akan dibagi menjadi beberapa bagian gambar kecil sehingga setiap gambar akan dinyatakan berdasarkan tingkat warna RGB menggunakan angka diskrit.

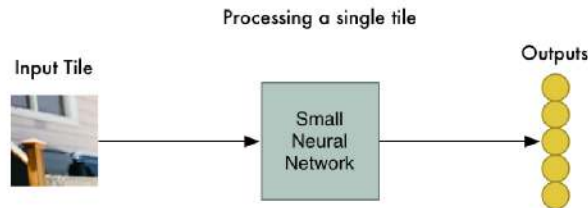


**Gambar 3.2** Pemecahan Gambar 77 Bagian (Geitgey, 2016)

Gambar di atas menunjukkan proses pencacahan gambar, sehingga gambar akan memiliki nilai matematis untuk proses konvolusi

2. *Menginput* setiap gambar kecil kedalam *Small Neural Network*

Pada proses ini, gambar – gambar kecil, di *input* kedalam *Small Neural Network* yang akan menghasilkan representasi fitur. Dengan demikian, CNN akan mudah mengenali objek berdasarkan posisi pada sebuah gambar.

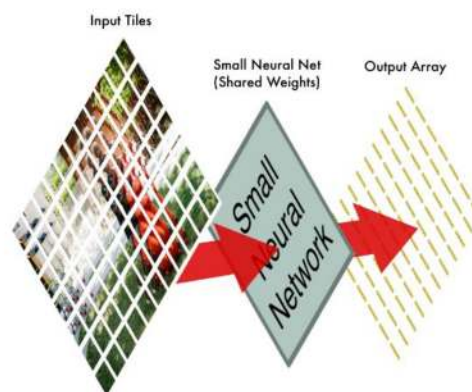


**Gambar 3.3** *Input* kedalam *Small Neural Network* (Geitgey, 2016)

Proses ini dilakukan pada semua gambar kecil menggunakan *filter* yang sama. *Input* gambar telah menjadi bilangan diskrit yang dihasilkan dari proses konvolusi, sehingga *filter* juga berupa bilangan diskrit sebagai pengali gambar. Faktor pengali pada seluruh potongan gambar memiliki nilai sama yang disebut dengan *weight sharing*. Jika suatu gambar terlihat objek yang sesuai dengan label, maka akan diberikan tanda sebagai *object interest*.

3. Melakukan penyimpanan dari *filter* ke dalam *array* baru

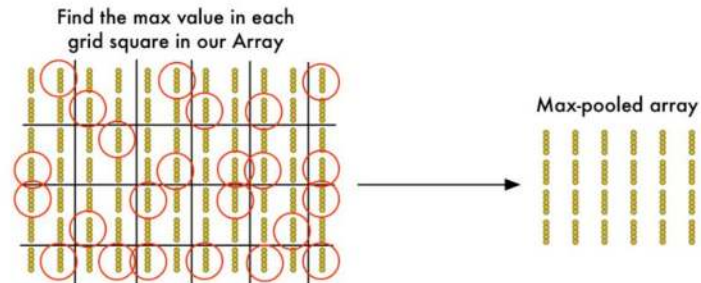
Ketika seluruh potongan gambar telah menghasilkan *output* diskrit dari proses *Small Neural Network*, maka akan diperoleh susunan *array* baru yang menjadi representasi gambar secara utuh.



**Gambar 3.4** Proses Pembentukan *Array* Baru (Geitgey, 2016)

4. *Downsampling*

*Output array* yang dihasilkan pada proses ketiga masih terlalu besar untuk menjadi sebuah pembelajaran model. Sehingga, *array* tersebut harus diperkecil ukurannya dengan menggunakan *downsampling*.

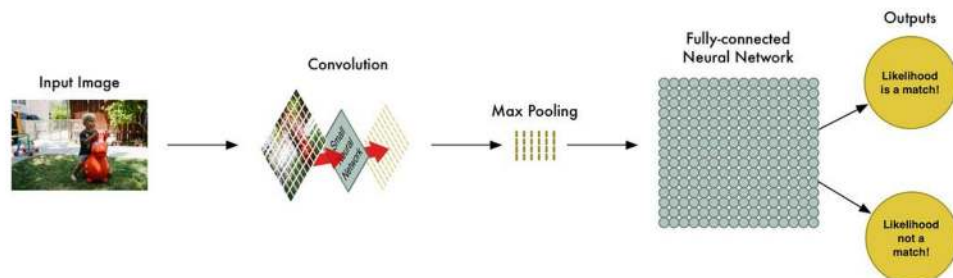


**Gambar 3.5** Proses *Max Pooling* (Geitgey, 2016)

*Downsampling* bekerja berdasar pada pengambilan nilai *pixel* terbesar untuk setiap *pooling kernel* (*max pooling*). Hal ini, akan menghasilkan informasi penting dapat tersaring meskipun melakukan pengurangan jumlah parameter awal.

### 5. Membuat Prediksi

Setelah melakukan keseluruhan proses, gambar yang besar akan menjadi kumpulan *array* yang lebih kecil. Model yang dirancang dari pembelajaran dinamakan sebagai *fully connected Network*. Tahap keseluruhan dalam model CNN untuk melakukan deteksi objek terlihat sebagai berikut.



**Gambar 3.6** Sistem Kerja CNN (Geitgey, 2016)

Dari gambar di atas, *Array* didefinisikan sebagai kelompok angka yang menjadi *input* pada jaringan syaraf untuk memproses kesesuaian pada gambar. Dimana hal tersebut merepresentasikan bahwa apakah gambar yang diinginkan sesuai atau tidak sesuai (Geitgey, 2016).

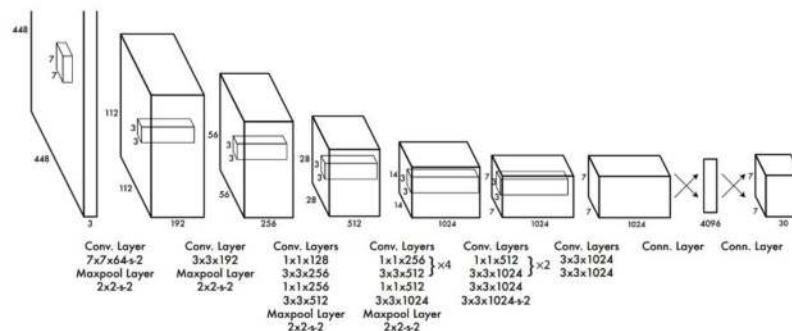
### 3.8. *You Only Look Once* (YOLO)

*You Only Look Once* (YOLO) merupakan sebuah program yang bertujuan untuk melakukan deteksi sebuah objek secara cepat. Sistem deteksi yang dilakukan dengan menggunakan *repurpose classifier* untuk melakukan pendeteksian. Model yang diterapkan dalam sebuah citra di beberapa lokasi dan skala, dimana daerah

dengan citra yang diberikan skor paling tinggi akan dianggap sebagai sebuah deteksi. Sebelum melakukan proses *training* perlu dilakukan anotasi untuk membentuk dataset. Pada setiap data akan memiliki nama kelas, titik koordinat x, koordinat y, panjang kotak pembatas dan lebar kotak pembatas.

YOLO menggunakan pendekatan jaringan syaraf tiruan yang digunakan untuk melakukan deteksi objek gambar. Jaringan membagi citra menjadi beberapa wilayah dan melakukan prediksi pada setiap *bounding box* dan peluang untuk setiap wilayah. *Bounding box* tersebut, dibandingkan dengan setiap peluang yang dilakukan prediksi.

Jaringan yang digunakan YOLO mempunyai 24 lapisan konvolusi (*convolutional layer*) yang dilanjutkan dengan 2 lapis *fully connected layer*. Untuk melakukan pengurangan pada kedalaman *feature maps*, beberapa lapisan konvolusi direduksi 1x1 yang diikuti dengan 3x3 lapisan konvolusional (*convolutional layer*). Arsitektur jaringan yang terdapat pada YOLO adalah sebagai berikut.



**Gambar 3.7** Arsitektur Jaringan pada YOLO (Nisa, 2023)

Berikut merupakan ilustrasi dari data *input* hingga proses deteksi objek pada YOLOv5:

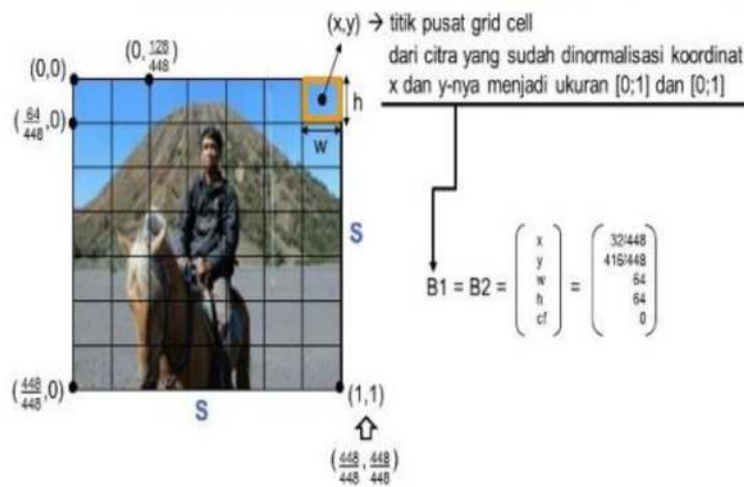
1. Membaca data *input* berupa gambar dengan ukuran bervariasi
2. Mengubah gambar *input* menjadi 488 x 488, kemudian membentuk  $S \times S$  *grid* pada gambar



**Gambar 3.8** Proses Pembagian Grid dan Probabilitas Kelas (Luthfi, 2021)

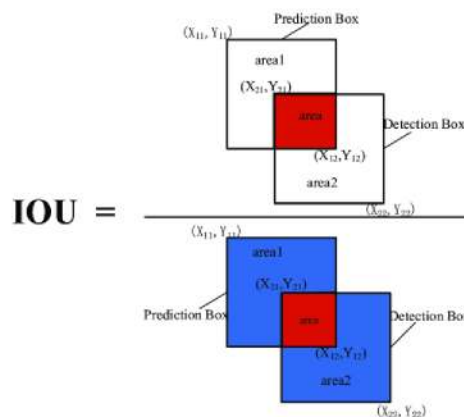
Jika ukuran S adalah 7, untuk masing – masing *grid* akan menjadi 7 x 7 sehingga diperoleh 49 sel *grid*. Pada Gambar 3.7 diperoleh citra yang memiliki 2 kelas ( $nC = 2$ ) yaitu objek “manusia” untuk (c1), “kuda” untuk (c2) dan “background”.

- Masing – masing sel, misalkan *box* ke-n = 2 memiliki *bounding box* memuat lima nilai diantaranya koordinat titik x dengan asumsi didasarkan pada baris ( $x_{baris}$ ), koordinat y dengan asumsi didasarkan pada kolom ( $y_{kolom}$ ), ukuran serta *confidence* ( $x, y, w, h, c$ ) pada  $nB$  *bounding box* di *bounding box* 1 dan *bounding box* 2.



**Gambar 3.9** Proses Perhitungan Nilai pada Sel Grid (Luthfi, 2021)

*Confidence score* merepresentasikan tingkat keakuratan kotak berdasarkan pada perkiraannya. Nilai tersebut didefinisikan sebagai  $Pr(objek) \cdot IoU_{pred}^{truth}$ . Jika tidak terdapat objek yang terdeteksi pada sel grid, maka nilai konfidensinya adalah nol atau nilai konfidensinya akan bernilai sama dengan *intersection over union* antara *ground truth* dan kotak prediksinya.





**Gambar 3.10** *Intersection Over Union* (Cheng et al., 2019)

*Intersection Over Union* adalah sebuah metode yang berguna untuk melakukan pengukuran akurasi deteksi objek pada dataset. *IoU* akan melakukan perbandingan antara *bounding box* pada data aktual dan *bounding box* data prediksi. Pada masing – masing kotak pembatas B tersusun dari 5 komponen diantaranya adalah  $x$ ,  $y$ ,  $w$ ,  $h$ , dan *confidence score*. Nilai dari titik  $(x, y)$  merepresentasikan pusat kotak yang relatif dengan batas *grid*. Nilai titik tersebut, dilakukan normalisasi agar berada diantara nol dan 1. Normalisasi juga akan dilakukan pada lebar ( $w$ ) dan tinggi ( $h$ ) yang relatif pada keseluruhan gambar. *Confidence score* merepresentasikan tingkat keyakinan pada model, dimana kotak pembatas B terdapat sebuah objek dan tingkat keakuratan kotak yang diprediksi tersebut. Prediksi dari YOLO akan memiliki *output vector*  $S, S, (B \cdot 5 + C)$ .

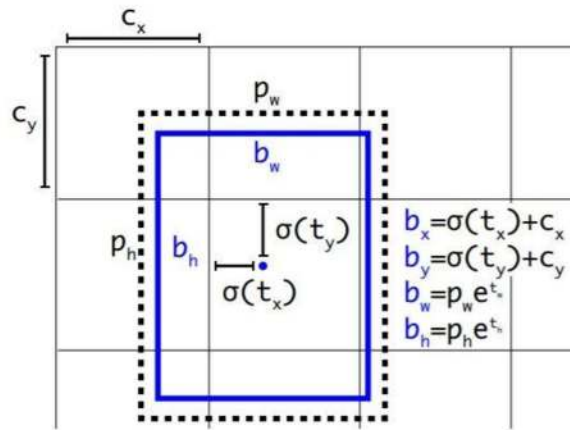
Dalam melakukan prediksi akhir, bergantung pada kelas skor kepercayaan (*class confidence score*) berdasarkan probabilitas kondisi kelas dan kotak skor keyakinan (*box confidence score*). *Class confidence score* digunakan untuk pengukuran nilai konfidensi pada klasifikasi dan lokalisasi sebuah objek sehingga akan diberikan nilai kepercayaan kelas yang spesifik pada setiap kotak yang melakukan pengkodean kemungkinan kelas yang berada di dalam kotak dan tingkat kesesuaian kotak yang diprediksi dengan objeknya. Dalam menghitung *class confidence score* dapat menggunakan persamaan sebagai berikut:

$$\Pr(Class_i|Object) \cdot \Pr(Object) \cdot IoU_{pred}^{truth} = \Pr(Class_i) \cdot IoU_{pred}^{truth} \quad (3.3)$$

Dengan keterangan :

$\Pr(Class_i|Object)$  : Probabilitas kondisi kelas  $i$ .

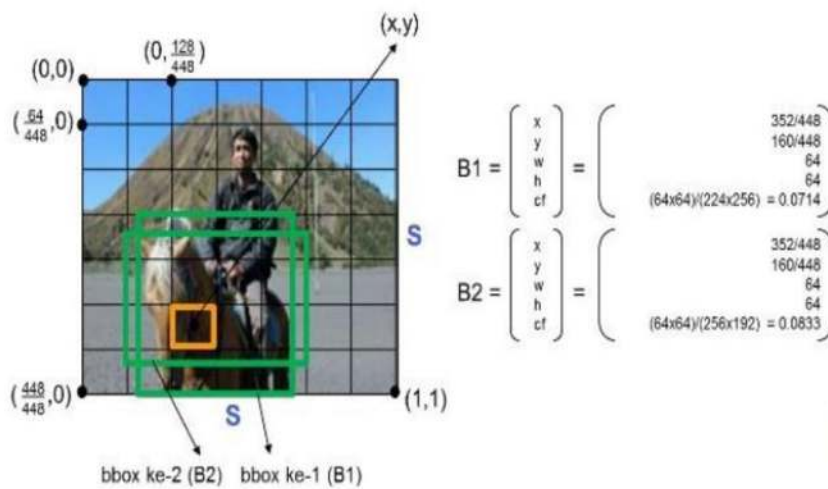
$\Pr(Object)$  : Probabilitas kelas  $i$ .



**Gambar 3.11** Bounding Box YOLO (Zhang Z. , 2020)

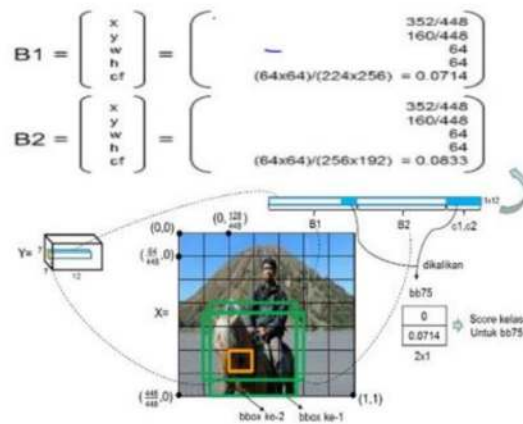
Dengan menggunakan persamaan di atas, akan diperoleh *confidence score* yang spesifik pada kelas. Nilai tersebut menyatakan probabilitas dari sebuah kelas yang berada dalam kotak dan tingkat keakuratan kotak yang dilakukan prediksi.

- Misalkan ada 2 bbox yang berada dalam kelas sama, ukuran tensor yang diperoleh ( $S \times S \times (nC \times 5 + nC)$ ) sehingga menjadi  $(7 \times 7 \times (2 \times 5 + 2)) = (7 \times 7 \times 12)$  tensor.

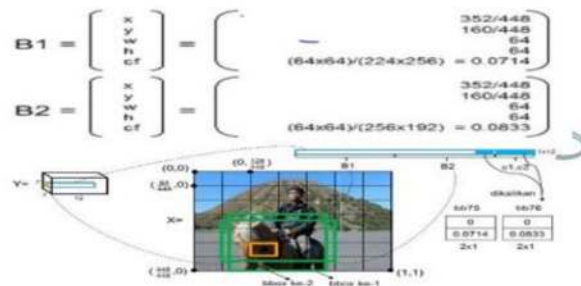


**Gambar 3.12** Perhitungan Tensor Dua Kelas Sama (Luthfi, 2021)

Kemudian, untuk matriks *bounding box* yang berukuran  $(nC \times (S \times S \times nB))$  yaitu  $2 \times (7 \times 7 \times 2) = (2 \times 98)$  yang diawali sel 1 hingga 49 pada 2 *bounding box* yang terdapat di kelas sama misalkan *bounding box* 75 dan *bounding box* 76.



Gambar 3.13 Perhitungan *Bounding Box* Kelas Sama (Luthfi, 2021)



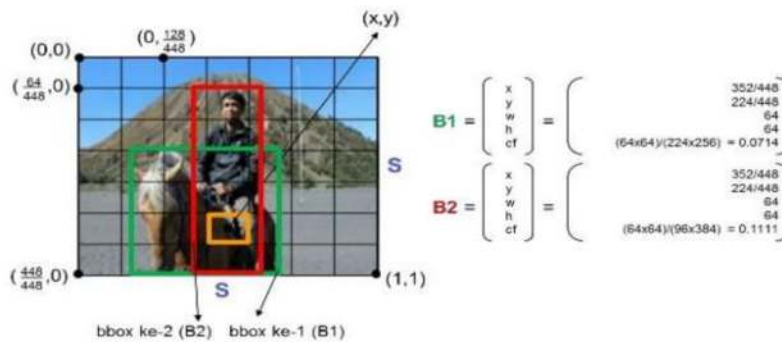
Gambar 3.14 Perhitungan *Bounding Box* Kelas Sama (Luthfi, 2021)

Hasil dari perhitungan matriks *bounding box* adalah :



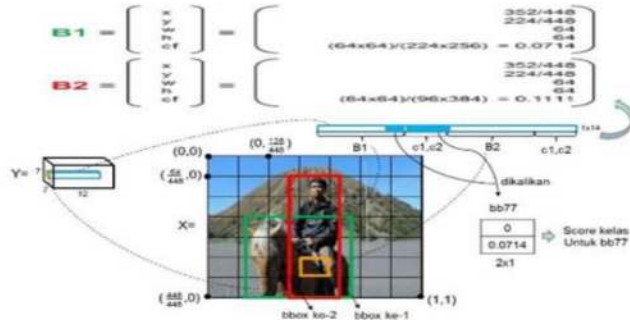
Gambar 3.15 Matriks *Bounding Box* (Luthfi, 2021)

- Jika ada 2 bbox yang berada dalam kelas yang berbeda atau terjadi *overlap*, hasil ukuran tensor yang diperoleh ( $S \times S \times (nB \times 5 + 2 \times nC)$ ) sehingga menjadi  $(7 \times 7 \times (2 \times 5 + 2 \times 2)) = (7 \times 7 \times 14)$  tensor.

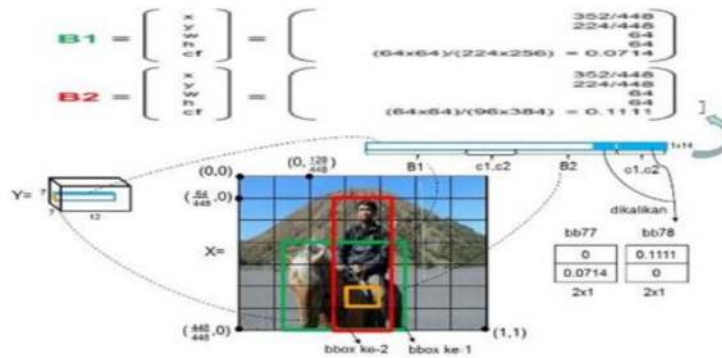


Gambar 3.16 Perhitungan *Bounding Box* Kelas Berbeda (Luthfi, 2021)

*Bounding Box* pertama dan kedua pada objek yang memiliki perbedaan. Untuk melakukan perhitungan matriks *Bounding Box* ukurannya ( $nC \times (S \times S \times nB) = 2 \times (7 \times 7 \times 2) = (2 \times 98)$  yang dimulai pada *grid* 1 hingga 49 pada 2 *bounding box* yang memiliki kelas berbeda misalkan *bounding box* 75 dan *bounding box* 76.



**Gambar 3.17** Perhitungan *Bounding Box* Kelas Berbeda (Luthfi, 2021)



**Gambar 3.18** Perhitungan *Bounding Box* Kelas Berbeda (Luthfi, 2021)

Hasil matriks *Bounding Box* adalah :

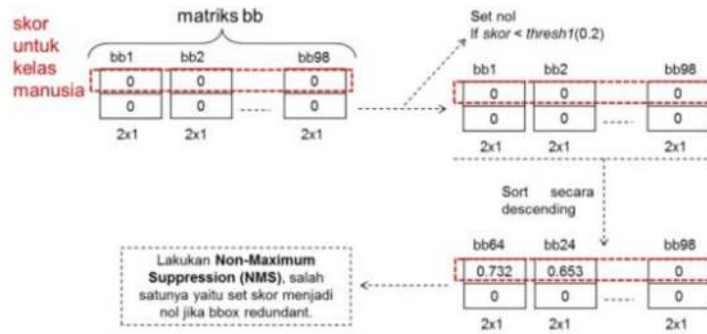
Kelas	bb1	bb2	bb77	bb78	bb97	bb98
manusia	0	0	0	0,1111	0	0
kuda	0	0	0,0714	0	0	0

↑

grid cell ke-1      grid cell ke-39      grid cell ke-49

**Gambar 3.19** Matriks *Bounding Box* (Luthfi, 2021)

- Masing – masing kelas pada matriks *bounding box* diset skor sebesar 0, apabila *score* kurang dari *threshold* 0,02 maka diberi urutan dari besar ke kecil.



**Gambar 3.20** Kelas Berdasar Pada Skor (Luthfi, 2021)

7. Melakukan proses *Non – Maximum Suppression* (NMS) dengan langkah :
  - a) Melakukan pendaftaran semua bbox



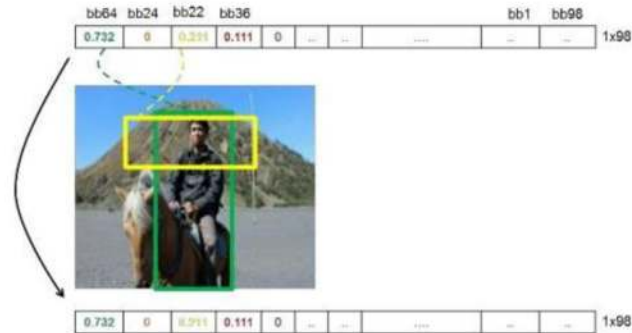
**Gambar 3.21** Pendaftaran Semua *Bounding Box* (Luthfi, 2021)

- b) Melakukan *set* bbox dengan skor tertinggi sebagai “bbox\_max”. misalkan bb65 adalah “bbox\_max” memiliki *score* 0,732
- c) Membandingkan “bbox\_max” dengan *bounding box* lain yaitu “bbox\_cur” dengan nilai lebih rendah daripada “bbox\_max”  $\neq 0$ . Ketika nilai IoU (bbox\_max,bbox\_cur)  $> 0.5$ , *score* pada bbox\_cur menjadi nol. Nilai bbox\_max = bb64, bbox\_cur = bb24, maka  $\text{IoU}(\text{bbox\_max}, \text{bbox\_cur}) > 0.5$  (*true*), skor bb24 = 0.



**Gambar 3.22** Perhitungan IoU *Bounding Box* (Luthfi, 2021)

- d) Kemudian untuk bb22, apabila  $bbox\_max = bb64$ ,  $bbox\_cur = bb22$ , IoU pada  $(bbox\_max, bbox\_cur) > 0.5$  (*false*) dan skor bb22 tidak mengalami perubahan.



**Gambar 3.23** Perhitungan IoU pada *Bounding Box* (Luthfi, 2021)

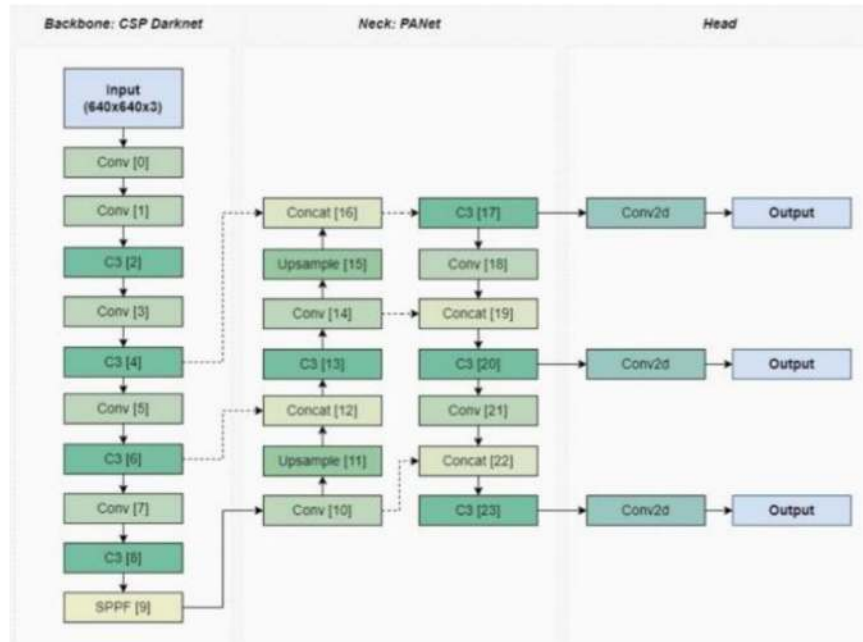
- e) Melanjutkan proses perbandingan hingga  $bbox\_max = bb64$ ,  $bbox\_cur = bb98$ . Lakukan *setting*  $bbox\_max = bb22$ , dan  $bbox\_cur = bb36$  hingga selesai. Lakukan proses yang sama untuk kelas kuda seperti kelas yang telah dilakukan pada objek manusia.

8. Dari hasil NMS maka diperoleh *bounding box*



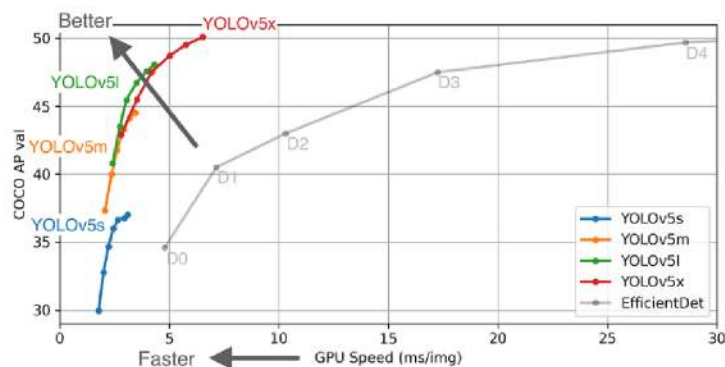
**Gambar 3.24** Hasil Akhir Pendeteksian YOLO (Luthfi, 2021)

YOLOv5 memiliki sebuah perbedaan dengan YOLO versi sebelumnya. Pada YOLOv5 *Darknet* telah diganti menggunakan *PyTorch*. YOLOv5 menggunakan *backbone CPSPDarknet53*. *Backbone* tersebut akan memecahkan perulangan informasi gradien pada *backbone* yang besar dan melakukan integrasi perubahan gradien ke dalam peta fitur dengan melakukan pengurangan kecepatan inferensi, pengurangan ukuran model dengan mengurangi parameter, dan peningkatan akurasi. YOLOv5 menggunakan *persistence apparent network* (PANet) untuk melakukan peningkatan arus informasi. Arsitektur YOLOv5 adalah sebagai berikut.



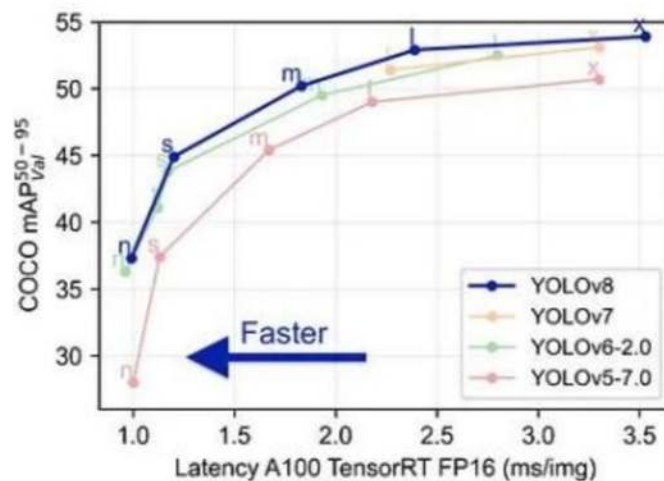
**Gambar 3.25** Arsitektur Jaringan YOLOv5 (Septyanto, Wibowo, & Setianingsih, 2022)

Jaringan pada YOLOv5 terdiri dari empat bagian diantaranya *input*, *backbone*, *neck*, dan *output*. Tahapan *input* berisi pemrosesan awal data seperti augmentasi. Jaringan *backbone* menggunakan *cross stage partial network (CSP)* dan *spatial pyramid pooling (SPP)*. SPP berfungsi mengekstrak peta fitur ukuran gambar berbeda menggunakan beberapa konvolusi dan *pooling*. BottleneckCSP berguna dalam mengurangi waktu dan kecepatan inferensi yang berpengaruh pada tingkat signifikansi model. Jaringan *Neck* terdiri dari struktur *feature pyramid network (FPN)* dan *persistence seaming network (PAN)*. Kedua struktur, memungkinkan ekstrak *feature* lebih efisien dari *backbone*. Tahapan *output* berguna dalam melakukan prediksi objek dengan ukuran berbeda pada peta fitur.



**Gambar 3.26** Tipe dan Performa YOLOv5 (Cochard, 2021)

Pada YOLOv5 terdapat beberapa tipe berdasar pada keunggulan, kecepatan dan performa mAP. YOLOv5 terdiri dari lima arsitektur yaitu YOLOv5s, YOLOv5m, YOLOv5l dan YOLOv5x. Gambar di atas menunjukkan perbedaan dari tipe dan performa YOLOv5. Keempat model memiliki ciri yang beda pada kecepatan deteksi dan keakuratan model. YOLOv5s (*small*) dapat bekerja lebih cepat, namun memiliki kinerja deteksi yang rendah khususnya pada objek yang kecil atau detail. YOLOv5m (*medium*) memiliki ukuran dan kompleksitas sedang serta seimbang antara kecepatan dan akurasi. YOLOv5l (*large*) lebih besar dan kompleks dari YOLOv5m yang memiliki kinerja lebih baik terutama untuk deteksi objek yang kecil dan detail, namun membutuhkan banyak sumber daya komputasi. YOLOv5x (*extra large*) merupakan versi yang paling besar dan kompleks dari YOLOv5 dengan kinerja deteksi yang sangat baik pada objek kecil dan kondisi yang sulit, namun membutuhkan sumber daya komputasi yang besar dan lebih lambat dalam inferensi. Dari grafik di atas terlihat bahwa semakin keatas, tingkat *average precision* akan semakin baik dan semakin ke kiri menunjukkan proses deteksi akan semakin cepat. Terlihat bahwa YOLOv5x memiliki nilai *average precision* terbaik, kemudian diikuti oleh YOLOv5m, YOLOv5l, dan YOLOv5s .



**Gambar 3.27** Tipe dan Performa YOLO (Cochard, 2021)

Diantara versi YOLO, peneliti menggunakan YOLOv5 dikarenakan arsitekturnya lebih sederhana dibandingkan versi setelahnya. Gambar 3.13 di atas menunjukkan grafik perbandingan waktu inferensi yang dibutuhkan dan akurasi mAP\_0.5:95 dalam melakukan deteksi menggunakan algoritma YOLOv5, YOLOv6, YOLOv7, dan YOLOv8. Berdasarkan grafik, YOLOv8 unggul dalam



nilai akurasi dibandingkan YOLO versi lainnya. Akan tetapi, YOLOv5 lebih cepat dalam kecepatan deteksi, karena membutuhkan waktu inferensi yang lebih rendah dibandingkan YOLO versi lainnya dengan rentang waktu 1 ms hingga 3.25 ms serta nilai mAP yang diberikan tidak terlalu jauh berbeda antara YOLOv5 dengan versi setelahnya. Disisi lain YOLOv5 juga mempunyai kelebihan pada *deployment*, sebab model yang diperoleh lebih ringan dengan ukuran yang lebih kecil dari versi sebelumnya.

### 3.9. Mean Average Precision (mAP)

*Mean average precision* (mAP) adalah metrik evaluasi yang digunakan dalam mengukur kualitas sistem pengambilan informasi yang digunakan dalam *object detection*. Dalam YOLOv5, menggunakan mAP50 dan mAP50-95 secara *default*. Nilai dari *mean average precision* diperoleh dari *average precision*. Pengukuran matriks ini, menggunakan *confusion matrix* dalam menghitung *precision* dan *recall* pada setiap label. Matriks mAP50 menggunakan *threshold IoU* sebesar 0,5. Matriks mAP50-95 menggunakan rata – rata mAP dari 10 nilai *threshold IoU* yaitu 0,5; 0,55; 0,60; 0,65; 0,70; 0,75; 0,80; 0,85; 0,90; dan 0,95 (Janapriya, 2023). Perhitungan nilai *mean average precision* dapat dihitung menggunakan persamaan berikut :

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (3.4)$$

Dengan keterangan :

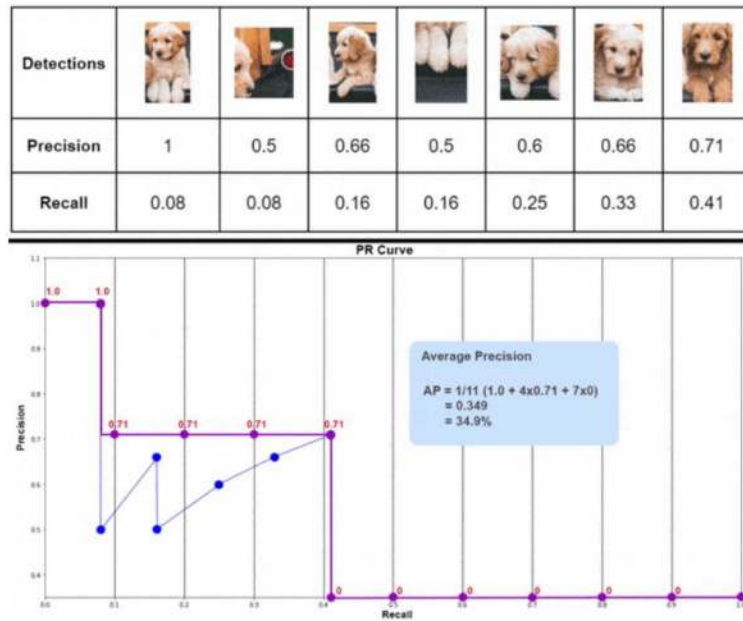
N : Jumlah data *average precision*

AP : *Average precision*

*Average precision* adalah nilai yang dihasilkan untuk setiap objek yang terdeteksi oleh sistem. Dalam hal ini sistem memprioritaskan item yang lebih relevan terdeteksi dimana *bounding box* sesuai dengan *ground truthnya* (Kukit, 2022). Perhitungan nilai *average precision* dapat dihitung menggunakan persamaan berikut :

$$AP = \frac{1}{11} (\text{sum of 11 interpolated precision values}) \quad (3.5)$$

Berikut merupakan ilustrasi perhitungan nilai *average precision* dari sebuah objek anjing:



**Gambar 3.28** Ilustrasi Perhitungan *Average Precision* (Kukit, 2022)

Dari hasil diatas nilai *average precision* dapat dihitung sebagai berikut :

$$\begin{aligned}
 AP_{dog} &= \frac{1}{11} (\text{sum of 11 interpolated precision values}) \\
 &= \frac{1}{11} (1 + 4 * 0,71 + 6 * 0) \\
 &= 0,349
 \end{aligned}$$

Berikut merupakan ilustrasi perhitungan nilai *mAP* dengan objek yang digunakan adalah *dog*, *sheep*, *truck*, dan *teddy* :

CLASS	dog	person	sheep	truck	teddy
AP	0.349	0.545	0.00	1.00	0.50

**Gambar 3.29** Ilustrasi Perhitungan *Mean Average Precision* (Kukit, 2022)

$$\begin{aligned}
 mAP &= \frac{1}{N} \sum_{i=1}^N AP_i \\
 &= \frac{1}{5} (0,349 + 0,545 + 0 + 1 + 0,5) \\
 &= 0,4788 \\
 &= 47, 88\%
 \end{aligned}$$

## BAB IV

### METODOLOGI PENELITIAN

#### 4.1. Populasi dan Sampel Penelitian

Populasi dalam penelitian ini yaitu kebun pisang yang terserang Hama Erionota Thrax L. di Provinsi Jawa Tengah. Sampel yang digunakan dalam penelitian ini adalah kebun pisang yang terserang Hama Erionota Thrax L. di Desa Gemolong, Dusun Buntalan, Kecamatan Klaten Tengah, Kabupaten Klaten, Provinsi Jawa Tengah.

#### 4.2. Tempat dan Waktu Penelitian

Tempat penelitian dilakukan di kebun pisang yang terletak di Desa Gemolong, Dusun Buntalan, Kecamatan Klaten Tengah, Kabupaten Klaten, Provinsi Jawa Tengah. Penelitian ini dilaksanakan di antara bulan November 2023 hingga Desember 2023.

#### 4.3. Definisi Operasional Variabel Penelitian

Variabel yang digunakan pada penelitian ini diantaranya adalah gambar daun pisang. Definisi operasional dari variabel didefinisikan pada tabel berikut.

**Tabel 4.1** Tabel Variabel Penelitian

No.	Variabel	Definisi Operasional Variabel
1	Gambar daun pisang	Gambar daun pisang dari varietas pisang kepok dan cavendish yang mengalami kerusakan akibat Hama Erionota Thrax L. dengan ukuran panjang 100 cm dan lebar 35 cm.

#### 4.4. Alat dan cara organisasi data

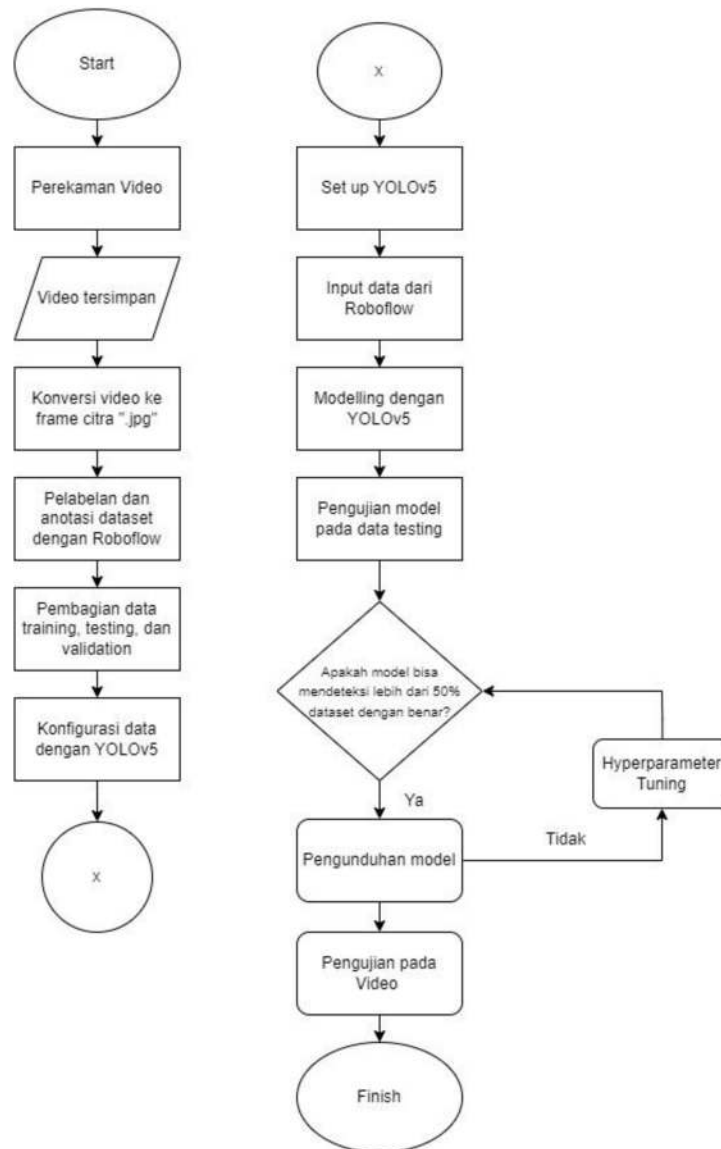
Alat dan *software* yang digunakan dalam penelitian adalah sebagai berikut:

1. Kamera Handphone Xiaomi redmi note 10S untuk mengumpulkan data berupa citra daun pisang yang terkena Hama Erionota Thrax L.
2. *Software* yang digunakan adalah Roboflow yang digunakan untuk melakukan pengelolaan dataset diantaranya adalah mengimpor gambar, memberikan label, melakukan preprocessing data yang meliputi resizing, augmentasi, perataan, pemotongan, perbaikan kualitas citra, hingga ekspor dataset.

3. Google Colaboratory digunakan untuk menjalankan kode *python* dalam melakukan pelatihan model YOLOv5.
4. Visual Studio Code (VS Code) digunakan sebagai *text editor* dan *running script* program.
5. Command Prompt (CMD) digunakan untuk menjalankan skrip program.

#### 4.5. Metode Penelitian

Dalam menyelesaikan tugas akhir ini akan dilakukan dalam beberapa tahapan pekerjaan. Metode yang digunakan untuk melaksanakan tugas akhir ini disajikan dalam diagram berikut:



**Gambar 4.1** Diagram alir Penelitian

Dari diagram di atas, tahapan penelitian yang dilakukan adalah sebagai berikut:

1. Perekaman video, yaitu mengumpulkan dataset daun pisang melalui video pada kamera HP Xiaomi Redmi Note 10S
2. Konversi video ke frame citra, yaitu mengubah format data dari mp4 menjadi ekstensi .jpg melalui *software Roboflow*
3. Pelabelan dataset, yaitu memberikan label pada objek di dataset
4. Pembagian data, yaitu proses membagi data menjadi data *training*, *testing*, dan validasi
5. Konfigurasi data, yaitu melakukan penyesuaian dataset dalam bentuk .txt dan .jpg pada format YOLOv5
6. *Clone github YOLOv5*, yaitu proses menyalin repositori git YOLOv5 ke *google collaboratory*
7. *Input data*, yaitu proses memasukkan data ke *google collaboratory* menggunakan kode *API Roboflow*
8. *Modelling*, yaitu proses pembuatan, penyesuaian dan evaluasi algoritma untuk membuat prediksi terhadap data baru
9. Pengujian model, yaitu menguji dan mengevaluasi kinerja model dapat melakukan prediksi terhadap data *testing*
10. *Hyperparameter Tuning*, yaitu proses mengubah parameter sebelum proses *training*.
11. Pengunduhan model, yaitu proses *mendownload* model terbaik yang telah diperoleh.

## BAB V

### HASIL DAN PEMBAHASAN

Pada penelitian ini, peneliti melakukan deteksi kerusakan daun pisang dengan menggunakan algoritma YOLOv5. Tujuan dari pendeteksian kerusakan ini adalah untuk menentukan apakah daun pisang memenuhi syarat untuk diekspor atau tidak. Tahapan utama dalam membuat model dimulai dengan *training* data. Pada proses ini memiliki tujuan untuk membentuk model yang akan digunakan pada pengujian data uji. Parameter yang digunakan untuk melakukan pengukuran tingkat keberhasilan model adalah *precision*, *recall*, *f1-score* dan nilai *mAP50*. Proses training pada penelitian ini menggunakan YOLOv5 dengan PyTorch pada *google colaboratory*.

#### 5.1. Dataset Daun Pisang

Dataset yang digunakan dalam penelitian ini adalah data primer yang diperoleh dengan cara melakukan pengambilan video pada beberapa kebun pisang di Desa Gemolong, Dusun Buntalan, Kecamatan Klaten Tengah, Kabupaten Klaten, Provinsi Jawa Tengah. Dari hasil perekaman diperoleh dua puluh video. Selanjutnya, dari rekaman tersebut akan dikonversi menjadi beberapa *frame* citra yang memiliki ekstensi “.jpg” dengan menggunakan *software Roboflow*.



**Gambar 5.1** *Frame* Citra Hasil Konversi Video

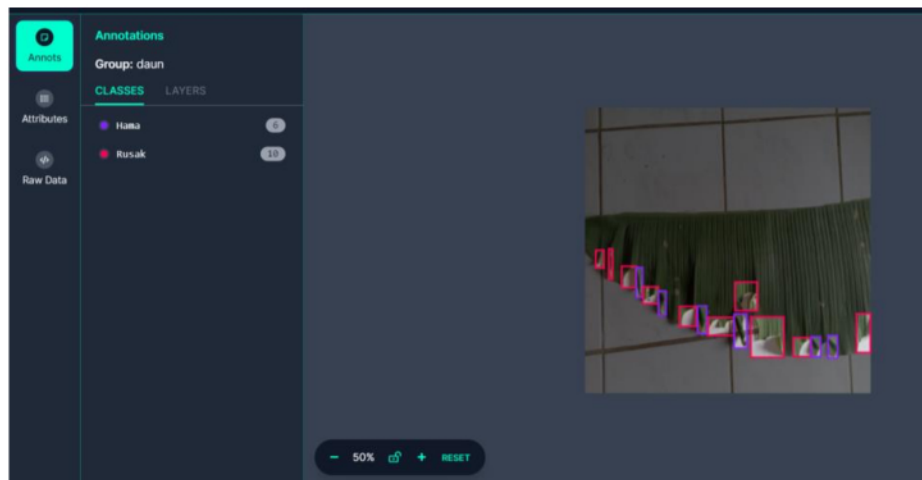
Dari proses konversi dua puluh video rekaman daun pisang diperoleh total 407 *frame* citra. Jumlah tersebut mencukupi untuk melaksanakan deteksi hama dan kerusakan pada daun pisang. Dataset yang digunakan diambil dari sudut pandang

yang beragam untuk memastikan bahwa algoritma YOLOv5 dapat mengenali objek dengan maksimal.

## 5.2. Preprocessing Dataset

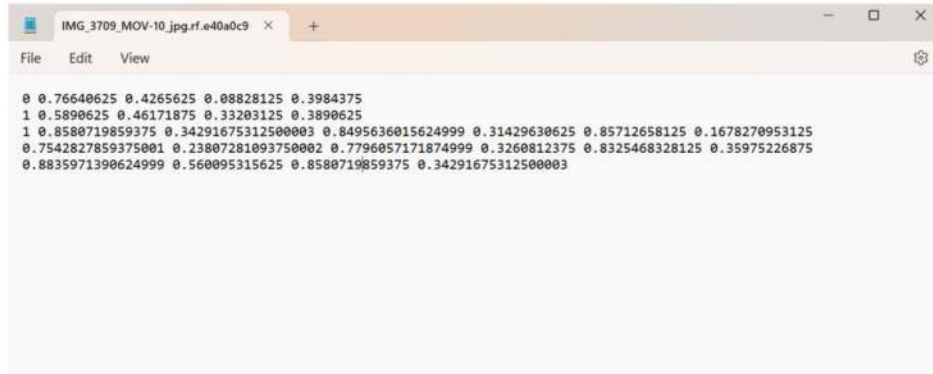
Setelah melakukan konversi data, langkah selanjutnya adalah melakukan *preprocessing* data yang meliputi pelabelan objek, melakukan *resize* gambar pada ukuran, *channel* warna, pembagian dataset *training*, *testing*, *validation*. Tahapan *pre-processing* ini menggunakan aplikasi Roboflow. Peneliti mengubah ukuran gambar menjadi 640 x 640 pixel. Pemilihan ukuran ini dilakukan dengan mempertimbangkan keseimbangan antara informasi pixel yang relevan dan memastikan kinerja algoritma deteksi objek YOLOv5. Pada penelitian ini dataset citra berisi 407 dataset yang dibagi menjadi data *training*, data *testing* dan data validasi. Dataset tersebut terbagi dengan perbandingan 70% dengan jumlah 285 gambar data *training*, 20% data validasi dengan jumlah 82 gambar, dan 10% data *testing* dengan jumlah 40 gambar.

Pada proses pelabelan dilakukan berdasarkan format anotasi YOLO yang mencakup nomor kelas objek *output*, koordinat x dan y objek kelas *output*, serta lebar dan tinggi objek kelas *output*. Pelabelan diberikan pada identifikasi dengan dua kategori yaitu “rusak” dan “hama”. Proses pelabelan objek daun pisang dilakukan secara teliti untuk memastikan bahwa area kerusakan pada daun pisang dan keberadaan hama teridentifikasi dengan baik. Hasil dari pelabelan ini akan disimpan dalam sebuah folder yang didalamnya juga berisikan *file* citra daun pisang.



**Gambar 5.2** Pelabelan Objek pada Gambar

Gambar 5.2 menunjukkan proses pelabelan objek pada gambar, dari pelabelan tersebut akan diperoleh file dengan format “.txt”. File – file tersebut akan menjadi dasar data *training* dengan model YOLOv5.



**Gambar 5.3** Hasil file “.txt” Pelabelan Objek

Pada Gambar 5.3, dapat diinterpretasikan isi dari file “.txt” pelabelan data sebagai berikut.

- a. Nomor kelas objek *output* yang terdiri dari 0 label kelas untuk hama dan 1 untuk rusak.
- b. Koordinat pusat *bounding box* (x dan y) objek kelas *output* pada urutan kedua dan ketiga.
- c. Lebar dan tinggi *bounding box* objek kelas *output* (dari kiri ke kanan) terletak pada urutan empat dan lima.

### 5.3. Training Data

Setelah melakukan *preprocessing* pada dataset, tahapan selanjutnya adalah melakukan *training* data untuk mendeteksi kerusakan daun pisang dengan YOLOv5. Proses *training* data dilakukan pada *platform Google Collaboratory* dengan menggunakan *framework Darknet*. Penelitian ini diawali dengan melakukan *cloning* repositori YOLOv5 dari situs GitHub ultralytic YOLOv5 dan melakukan *installasi* dependensi yang diperlukan. Selanjutnya, peneliti melakukan koneksi ke Roboflow dengan kunci API dengan model YOLOv5. Dalam melakukan deteksi objek, YOLOv5 menggunakan IOU (*intersection over union*) dan *Non – Max Suppression* untuk mengukur rasio *bounding box* pada objek yang akan diprediksikan dengan anotasi benar ketika IOU bernilai  $> 0,5 - 0,9$  sehingga dapat diterima. Ketika objek memiliki nilai *confidence*  $> 0,5$  maka diberikan sebuah *bounding box* pada objek. Namun, jika nilai *confidence*  $< 0,5$  akan diasumsikan



sebagai *background* atau tidak terdeteksi. Pada *training* data, model dilatih menggunakan beberapa parameter diantaranya adalah:

- a. *Img* mendeskripsikan mengenai ukuran gambar *input* yang digunakan dalam *training* model. Pada penelitian ini, ukuran gambar yaitu 640 x 640 pixel. Ukuran gambar *input* tersebut mempengaruhi kecepatan pelatihan dan akurasi dalam deteksi.
- b. *Batch size* yang menjelaskan mengenai jumlah gambar yang diproses dalam satu iterasi pelatihan. Pada penelitian ini menggunakan *batch size* sebesar 8 dan 16. Ukuran *batch* yang semakin besar akan mempercepat dalam *training* model, namun hal tersebut memerlukan lebih banyak GPU.
- c. *Epoch* menjelaskan mengenai jumlah iterasi penuh yang digunakan selama pelatihan. Jumlah *epoch* tersebut menentukan konvergensi model dalam mempelajari pola data. Jumlah *epoch* yang terlalu kecil akan menyebabkan *underfitting* dan jika terlalu besar akan menyebabkan *overfitting*. Pada penelitian ini digunakan nilai *epoch* sebesar 50 dan 100.
- d. Nilai *confidence threshold* yang mendeskripsikan tentang seberapa tinggi nilai kepercayaan deteksi objek agar diperoleh hasil yang valid. Pada penelitian ini, nilai *confidence threshold* sebesar 0,1.
- e. Model YOLOv5 yang digunakan dalam penelitian ini adalah YOLOv5s, YOLOv5m dan YOLOv5x. Hal tersebut digunakan untuk melihat perbandingan dari performa ketiga model.
- f. Data yang mendeskripsikan mengenai struktur dan lokasi dataset yang digunakan dalam *training* model. Pada penelitian ini menggunakan `data.yaml`

### 5.3.1 Pengujian Nilai *Epoch*, *Batch* dan Model YOLOv5

Parameter yang dapat digunakan untuk mendapatkan model yang baik diantaranya adalah *epoch*, *batch* dan model. Pada penelitian ini menggunakan nilai *epoch* sebesar 50 dan 100 dengan *batch size* 8 dan 16 serta tiga model yaitu YOLOv5s, YOLOv5m, dan YOLOv5x. Pengujian ini dilakukan dengan tujuan untuk mengetahui pengaruh tiga parameter tersebut pada hasil mAP. Hasil pengujian disajikan dalam tabel berikut.

**Tabel 5.1** Hasil Pengujian Tiga Parameter

No	<i>Img</i>	<i>Epoch</i>	<i>Batch</i>	Model	mAP
1	640	50	8	YOLOv5s	0,85
2	640	100	8	YOLOv5s	0,847
3	640	50	16	YOLOv5s	0,846
4	640	50	8	YOLOv5m	0,838
5	640	50	8	YOLOv5x	0,867

Berdasarkan Tabel 5.1 dengan membandingkan ketiga parameter, hasil menunjukkan bahwa model dengan 50 epoch memiliki nilai *Mean Average Precision* (mAP) lebih tinggi daripada yang dengan 100 epoch. Hal ini mungkin disebabkan oleh *overfitting* pada model 100 epoch, di mana model terlalu sesuai dengan data *training* dan kurang umum untuk data *testing*. Selain itu, ukuran dataset yang relatif kecil dan faktor-faktor lain seperti *learning rate* dan ketidakstabilan pelatihan dapat memengaruhi performa. Sehingga, model dengan 50 epoch dinilai lebih baik karena mampu menghindari *overfitting* dan menangani tantangan *training* dengan lebih baik dibanding model 100 epoch. Perubahan nilai mAP antara ukuran *batch* 8 dan 16 pada model YOLOv5s cukup kecil, menunjukkan bahwa variasi nilai *batch* memiliki dampak yang relatif kecil terhadap performa deteksi objek. Dari perbandingan model algoritma YOLOv5 diperoleh model YOLOv5x memiliki kinerja yang lebih baik dalam melakukan deteksi objek dibandingkan dengan model YOLOv5s dan YOLOv5m yang menunjukkan bahwa kompleksitas model dapat berpengaruh terhadap kualitas deteksi.

a. Pengujian model YOLOv5s

*Training* data pada YOLOv5s dengan *batch size* 8, *epoch* 50, optimizer SGD menghabiskan waktu sekitar 6,96 menit. Gambar berikut menunjukkan proses *training* dengan model YOLOv5s.

```
50 epochs completed in 0.116 hours.
Optimizer stripped from runs/train/exp3/weights/last.pt, 14.4MB
Optimizer stripped from runs/train/exp3/weights/best.pt, 14.4MB

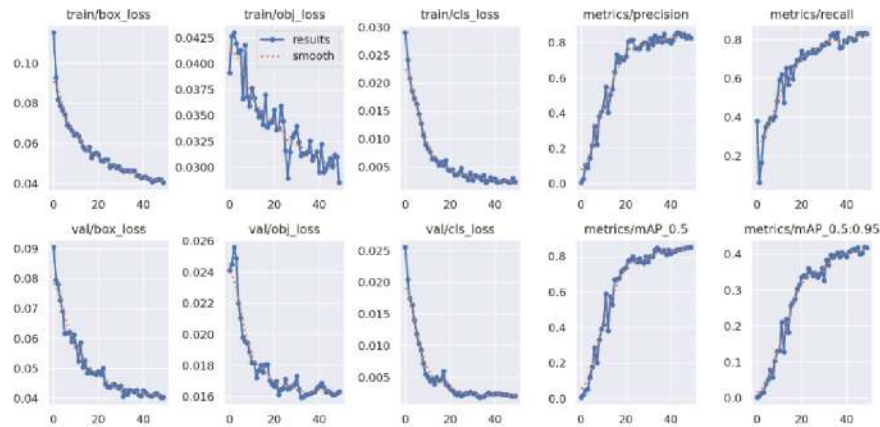
Validating runs/train/exp3/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7015519 parameters, 0 gradients, 15.8 GFLOPs

```

Class	Images	Instances	P	R	mAP50	mAP50-95
all	82	298	0.837	0.835	0.85	0.419
Hama	82	74	0.946	0.943	0.94	0.471
Rusak	82	224	0.728	0.728	0.761	0.367

**Gambar 5.4** Hasil Validasi Model YOLOv5s

Berdasarkan kinerja dari YOLOv5s yang telah dilakukan untuk mendeteksi objek berupa hama diperoleh nilai mAP sebesar 94% dan rusak sebesar 76,1% dengan keseluruhan nilai mAP sebesar 85%. Berikut merupakan grafik kinerja YOLOv5s pada proses *training*.

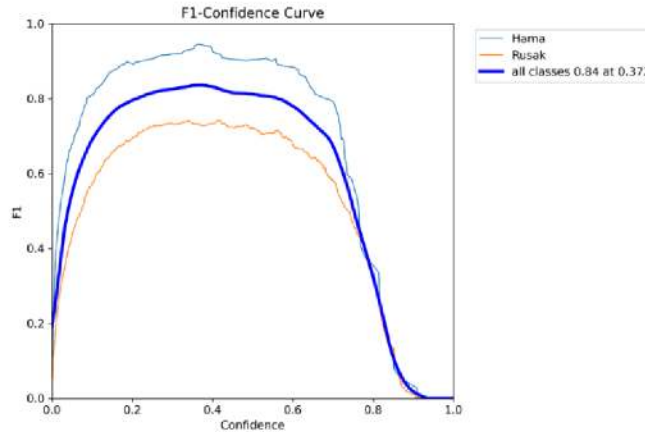


**Gambar 5.5** Grafik Kinerja Model YOLOv5s

Gambar 5.5 merepresentasikan grafik – grafik hasil *training* dan validasi dari data. Pada grafik *box*, *object*, dan *classification* perubahan yang terjadi pada nilai *loss* dapat memberikan informasi mengenai seberapa baik model belajar selama *training* dan seberapa baik generalisasi model pada data validasi. Penurunan pada *train/box\_loss* menunjukkan bahwa model belajar agar lebih akurat dalam melakukan prediksi koordinat *bounding box*. Penurunan pada *train/obj\_loss* menunjukkan bahwa model belajar untuk lebih baik dalam mendeteksi keberadaan objek pada data pelatihan. Penurunan *train/cls\_loss* menunjukkan bahwa model belajar untuk lebih baik dalam melakukan klasifikasi pada data pelatihan. Dari keseluruhan *loss* grafik di atas, terlihat bahwa grafik menunjukkan penurunan *loss* sepanjang waktu sehingga tergolong sebagai indikator positif. Penurunan *loss* menginterpretasikan bahwa model semakin memahami pola dalam data *training* dan melakukan prediksi lebih baik.

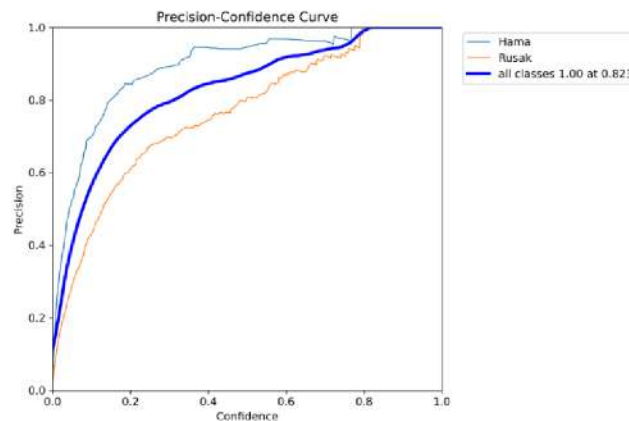
Grafik *metrics/precision* dan *recall* memiliki kecenderungan berkebalikan dengan grafik *box*, *object*, dan *classification*. Peningkatan *precision* mencerminkan peningkatan kemampuan model untuk menghasilkan prediksi positif yang benar. Sedangkan, peningkatan *metrics/recall* mencerminkan peningkatan kemampuan model untuk mendeteksi sebagian besar objek yang sebenarnya. Selain itu, peningkatan mAP mencerminkan peningkatan kemampuan model untuk

mendeteksi dan mengklasifikasikan objek dengan lebih baik secara keseluruhan. Sehingga dapat di simpulkan bahwa semakin tinggi nilai dari presisi dan *recall* maka sistem memiliki kinerja yang lebih baik.



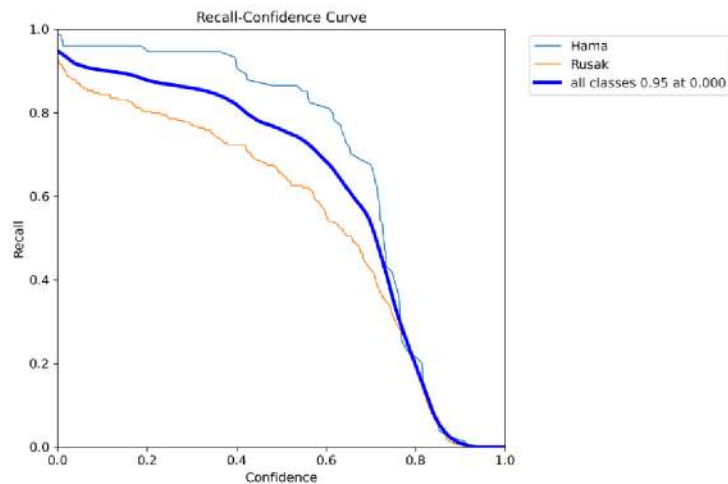
**Gambar 5.6** Grafik Nilai *F1 Score* Terhadap Nilai *Confidence*

Gambar 5.6 merepresentasikan grafik dari *F1 Score*. *F1 score* digunakan untuk mengukur performa model dengan menggabungkan *precision* dan *recall*. Dari grafik terlihat bahwa nilai *F1 Confidence* mencapai titik tertinggi sebesar 0,84 pada keseluruhan kelas pada saat nilai *confidence* sebesar 0,372.



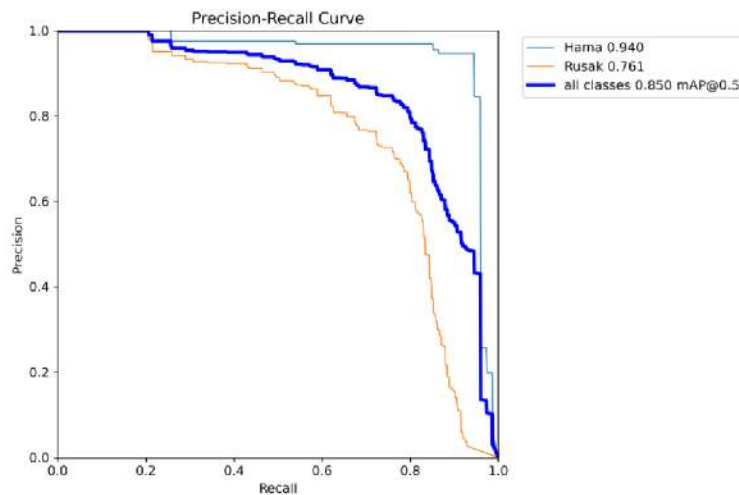
**Gambar 5.7** Grafik Nilai *Precision* Terhadap Nilai *Confidence*

Gambar 5.7 memvisualisasikan antara nilai *confidence* terhadap nilai *precision*. Dari grafik, dapat disimpulkan bahwa kelas hama dan rusak mencapai nilai *precision* tertinggi ketika nilai *confidence* berada pada nilai 0,823. Hal tersebut dapat dikatakan bahwa semakin cepat nilai kelas hama dan rusak mencapai puncak tertinggi, maka sistem akan bekerja lebih baik dalam memprediksi objek.



**Gambar 5.8** Grafik Nilai *Recall* Terhadap Nilai *Confidence*

Gambar 5.8 menunjukkan grafik antara *confidence* dengan nilai *recall*. Grafik tersebut merepresentasikan nilai antara kelas hama dan rusak mencapai nilai *recall* 0,95 saat nilai *confidence* 0. Pada saat *confidence score* nilainya mendekati 1, maka nilai *recall* dari setiap kelas akan mengalami penurunan. Hal ini dikarenakan, nilai *recall* merupakan sebuah matrik yang digunakan dalam mengukur sebuah hal yang positif.



**Gambar 5.9** Grafik Nilai *Precision* Terhadap Nilai *Recall*

Gambar 5.9 menunjukkan kurva antara *precision* dan nilai *recall*. Dari kurva tersebut, dapat dilihat bahwa nilai tertinggi dari kelas hama sebesar 0,940 dan kelas rusak sebesar 0,761 pada saat nilai mAP sebesar 0,5.

b. Pengujian model YOLOv5m

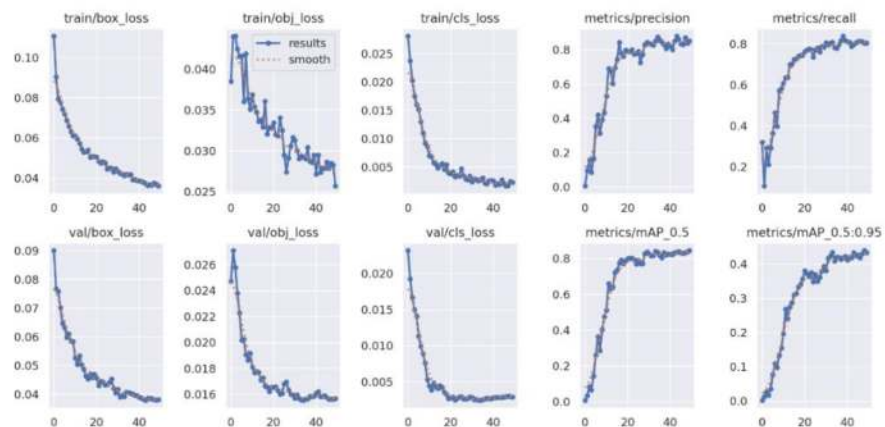
Training data pada YOLOv5 dengan *batch size* 8, *epoch* 50, optimizer SGD dan model YOLOv5m menghabiskan waktu sekitar 10,92 menit. Gambar berikut menunjukkan proses *training* dengan model YOLOv5m.

```
50 epochs completed in 0.182 hours.
Optimizer stripped from runs/train/exp4/weights/last.pt, 42.2MB
Optimizer stripped from runs/train/exp4/weights/best.pt, 42.2MB

Validating runs/train/exp4/weights/best.pt...
Fusing layers...
Model summary: 212 layers, 20856975 parameters, 0 gradients, 47.9 GFLOPs
Class      Images  Instances  P      R      mAP50  mAP50-95: 100% 6/6 [00:03<00:
all        82      298       0.839  0.806  0.838  0.441
Hama      82      74        0.904  0.892  0.915  0.494
Rusak     82      224       0.774  0.719  0.762  0.389
```

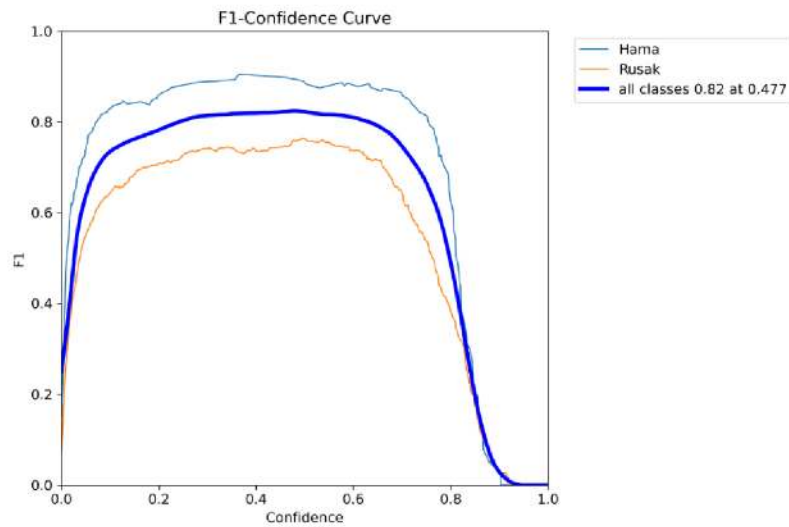
**Gambar 5.10** Hasil *Training* Model YOLOv5m

Berdasarkan kinerja dari YOLOv5m yang telah dilakukan untuk mendeteksi objek berupa hama diperoleh nilai mAP sebesar 91,5% dan rusak sebesar 76,2% dengan keseluruhan nilai mAP sebesar 83,8%. Berikut merupakan grafik kinerja YOLOv5m pada proses *training*.



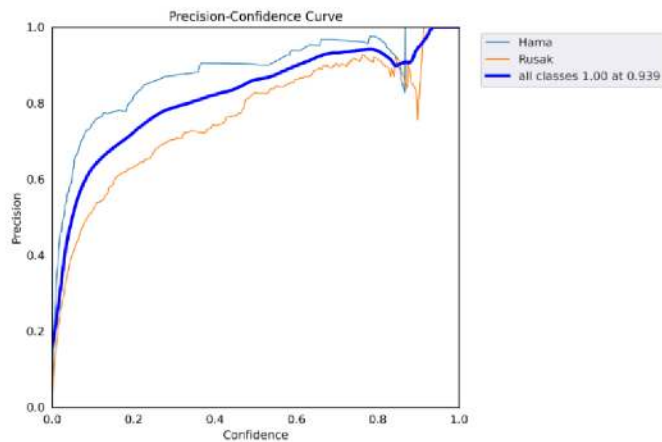
**Gambar 5.11** Grafik Kinerja Model YOLOv5m

Gambar 5.11, merepresentasikan grafik – grafik hasil *training* dan validasi dari data. Grafik *bounding box*, objek, klasifikasi mengalami penurunan secara eksponensial. Hal tersebut dipengaruhi oleh jumlah *epoch* yang dilakukan dalam proses *training*. *Epoch* yang digunakan dalam *training* model YOLOv5m sebanyak 50 *epoch*. Angka tersebut dinilai tidak terlalu memberatkan sistem dan nilai *loss* yang dihasilkan tidak terlalu besar. Pada grafik *precision*, *recall*, mAP 0,5 dan mAP 0,5 : 0,95 berkebalikan dengan tiga grafik sebelumnya. Nilai yang diperoleh dari *precision* dan *recall* untuk mAP 0,5 sebesar 0,838.



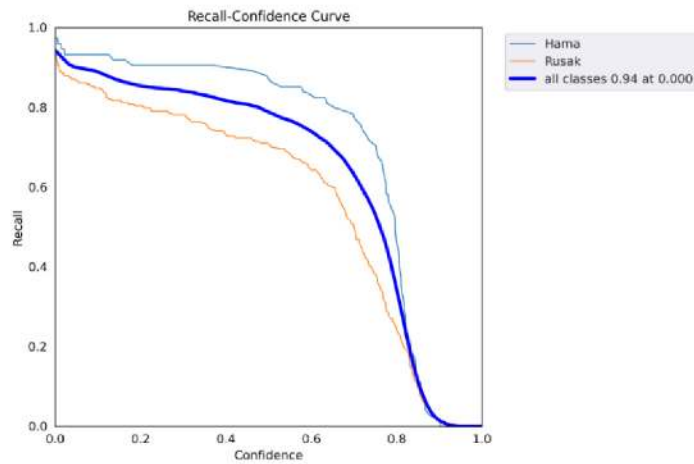
**Gambar 5.12** Grafik Nilai *F1 Score* Terhadap Nilai *Confidence*

Gambar 5.12 merepresentasikan grafik dari *F1 Score*. *F1 score* digunakan untuk mengukur performa model dengan menggabungkan *precision* dan *recall*. Dari grafik terlihat bahwa nilai *f1 confidence* mencapai titik tertinggi sebesar 0,82 pada keseluruhan kelas pada saat nilai *confidence* sebesar 0,477.



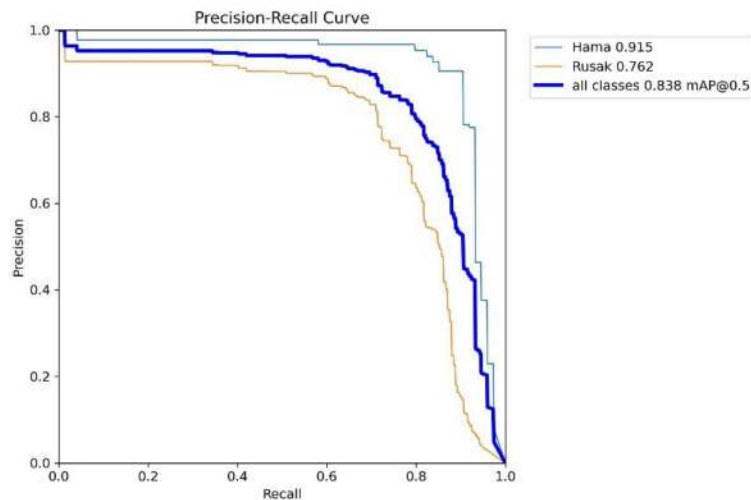
**Gambar 5.13** Grafik Nilai *Precision* Terhadap Nilai *Confidence*

Gambar 5.13 memvisualisasikan antara nilai *confidence* terhadap nilai *precision*. Dari grafik, dapat disimpulkan bahwa kelas hama dan rusak mencapai nilai *precision* tertinggi ketika nilai *confidence* berada pada nilai 0,939. Hal tersebut dapat dikatakan bahwa semakin cepat nilai kelas hama dan rusak mencapai puncak tertinggi, maka sistem akan bekerja lebih baik dalam memprediksi objek.



**Gambar 5.14** Grafik Nilai *Recall* Terhadap Nilai *Confidence*

Gambar 5.14 menunjukkan grafik antara *confidence* dengan nilai *recall*. Grafik tersebut merepresentasikan nilai antara kelas hama dan rusak mencapai nilai *recall* 0,94 saat nilai *confidence* 0. Pada saat *confidence score* nilainya mendekati 1, maka nilai *recall* dari setiap kelas akan mengalami penurunan. Hal ini dikarenakan, nilai *recall* merupakan sebuah matrik yang digunakan dalam mengukur sebuah hal yang positif.



**Gambar 5.15** Grafik Nilai *Precision* Terhadap Nilai *Recall*

Gambar 5.15 menunjukkan kurva antara *precision* dan nilai *recall*. Dari kurva tersebut, dapat dilihat bahwa nilai tertinggi dari kelas hama sebesar 0,915 dan kelas rusak sebesar 0,762 pada saat nilai mAP sebesar 0,5.

c. Pengujian model YOLOv5x



Training data pada YOLOv5x dengan *batch size* 8, *epoch* 50, optimizer SGD menghabiskan waktu sekitar 32,22 menit. Gambar berikut menunjukkan proses *training* dengan model YOLOv5x.

```

50 epochs completed in 0.537 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 173.1MB
Optimizer stripped from runs/train/exp/weights/best.pt, 173.1MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 322 layers, 86180143 parameters, 0 gradients, 203.8 GFLOPs

```

Class	Images	Instances	P	R	mAP50	mAP50-95	100% 6/6 [00:04:00:
all	82	298	0.877	0.866	0.867	0.463	
Hama	82	74	0.917	0.959	0.961	0.528	
Rusak	82	224	0.837	0.772	0.773	0.398	

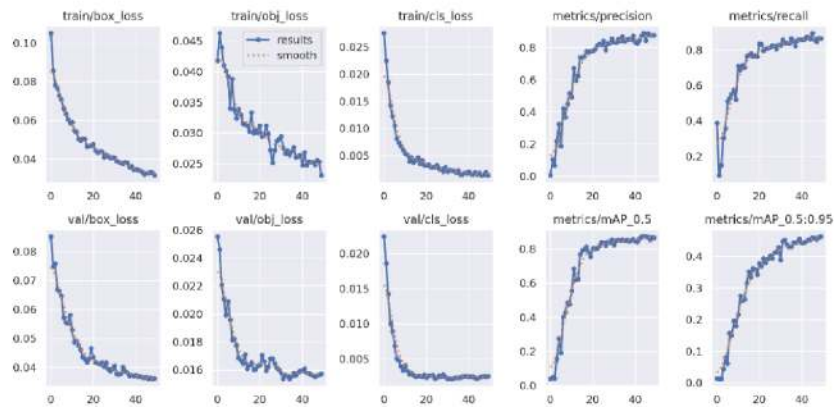
```

Results saved to runs/train/exp

```

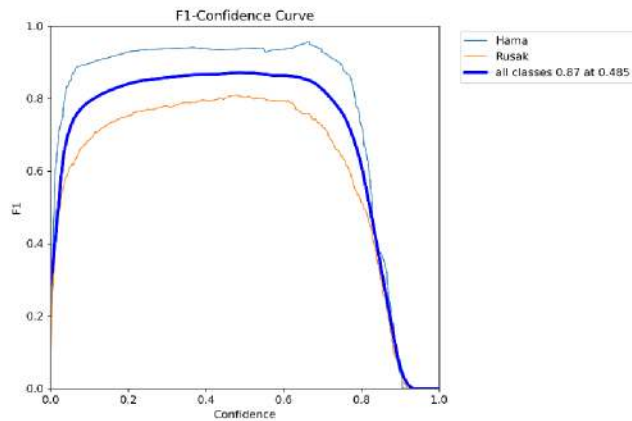
**Gambar 5.16** Hasil *Training* Model YOLOv5x

Berdasarkan kinerja dari YOLOv5x yang telah dilakukan untuk mendeteksi objek berupa hama diperoleh nilai mAP sebesar 96,1% dan rusak sebesar 77,3% dengan keseluruhan nilai mAP sebesar 86,7%. Berikut merupakan grafik kinerja YOLOv5x pada proses *training*.



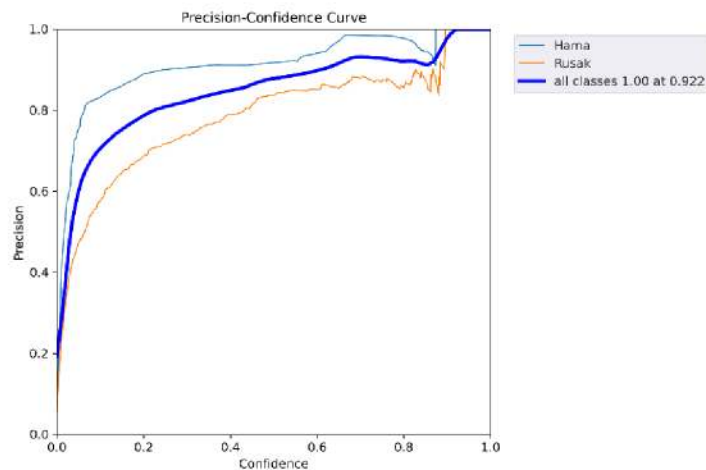
**Gambar 5.17** Grafik Kinerja Model YOLOv5x

Gambar 5.17, merepresentasikan grafik – grafik hasil *training* dan validasi dari data. Grafik *bounding box*, objek, klasifikasi mengalami penurunan secara eksponensial. Hal tersebut dipengaruhi oleh jumlah *epoch* yang dilakukan dalam proses *training*. *Epoch* yang digunakan dalam *training* model YOLOv5x sebanyak 50 *epoch*. Angka tersebut dinilai tidak terlalu memberatkan sistem dan nilai *loss* yang dihasilkan tidak terlalu besar. Pada grafik *precision*, *recall*, mAP 0,5 dan mAP 0,5 : 0,95 berkebalikan dengan tiga grafik sebelumnya. Nilai yang diperoleh dari *precision* dan *recall* untuk mAP 0,5 sebesar 0,867.



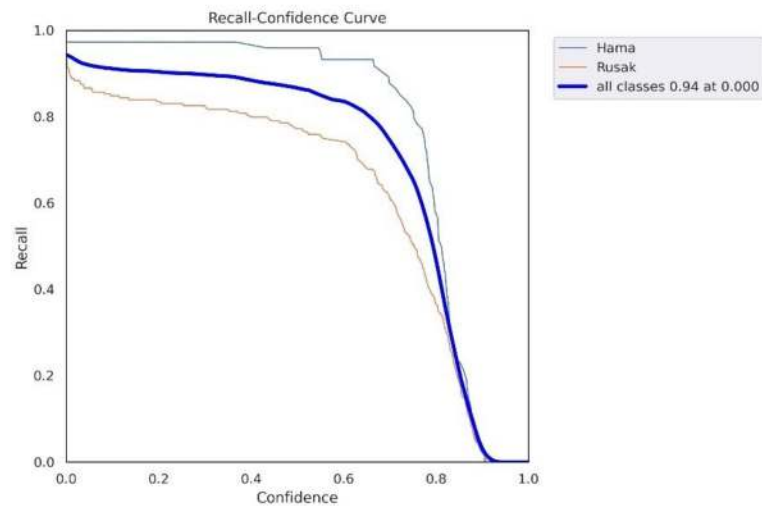
**Gambar 5.18** Grafik Nilai *F1 Score* Terhadap Nilai *Confidence*

Gambar 5.18 merepresentasikan grafik dari *F1 Score*. *F1 score* digunakan untuk mengukur performa model dengan menggabungkan *precision* dan *recall*. Dari grafik terlihat bahwa nilai *F1 Confidence* mencapai titik tertinggi sebesar 0,87 pada keseluruhan kelas pada saat nilai *confidence* sebesar 0,485.



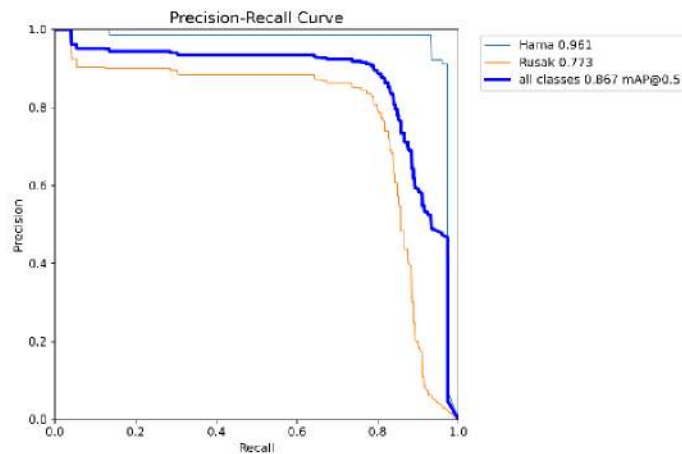
**Gambar 5.19** Grafik Nilai *Precision* Terhadap Nilai *Confidence*

Gambar 5.19 memvisualisasikan antara nilai *confidence* terhadap nilai *precision*. Dari grafik, dapat disimpulkan bahwa kelas hama dan rusak mencapai nilai *precision* tertinggi ketika nilai *confidence* berada pada nilai 0,922. Hal tersebut dapat dikatakan bahwa semakin cepat nilai kelas hama dan rusak mencapai puncak tertinggi, maka sistem akan bekerja lebih baik dalam memprediksi objek.



**Gambar 5.20** Grafik Nilai *Recall* Terhadap Nilai *Confidence*

Gambar 5.20 menunjukkan grafik antara *confidence* dengan nilai *recall*. Grafik tersebut merepresentasikan nilai antara kelas hama dan rusak mencapai nilai *recall* 0,94 saat nilai *confidence* 0. Pada saat *confidence score* nilainya mendekati 1, maka nilai *recall* dari setiap kelas akan mengalami penurunan. Hal ini dikarenakan, nilai *recall* merupakan sebuah matrik yang digunakan dalam mengukur sebuah hal yang positif.



**Gambar 5.21** Grafik Nilai *Precision* Terhadap Nilai *Recall*

Gambar 5.21 memvisualisasikan antara nilai *precision* terhadap nilai *recall*. Dari kurva tersebut, dapat dilihat bahwa nilai tertinggi dari kelas hama sebesar 0,961 dan kelas rusak sebesar 0,773 pada saat nilai mAP sebesar 0,5.

#### 5.4. Deteksi Objek dengan YOLOv5









Dalam proses deteksi menggunakan YOLOv5 yang telah dilatih, langkah pertama yang dilakukan adalah melakukan pengunduhan file *"best.pt"* yang didalamnya memuat bobot final dari *training* data pada jaringan YOLOv5. Selanjutnya, akan dilakukan proses deteksi daun pisang pada citra data *test* menggunakan bobot pada jaringan *training*.







#### 5.5. Pengujian Performa Pada Dataset

Pengujian performa jaringan pada dataset akan dilakukan dengan melakukan perbandingan total daun pisang yang terdeteksi hama dan rusak pada data aktual dan data prediksi dengan menggunakan dataset peneliti sebagai berikut.

**Tabel 5.2** Hasil Pengujian Performa pada Dataset

Aktual	Prediksi	Keterangan
 <p>Hama : 1</p>	 <p>Tidak ada yang terdeteksi</p>	Pada frame gambar ini, terdapat objek Hama yang tidak terdeteksi, hal tersebut disebabkan karena jaringan tidak mampu mengenali fitur dari Hama tersebut.
 <p>Hama : 3 Rusak : 3</p>	 <p>Hama : 3 Rusak : 3</p>	Pada frame gambar ini, semua objek dapat terdeteksi dengan baik.
 <p>Rusak : 2</p>	 <p>Rusak : 2</p>	Pada frame gambar ini, semua objek dapat terdeteksi dengan baik.

Aktual	Prediksi	Keterangan
 <p data-bbox="411 562 544 651">Hama : 1 Rusak : 3</p>	 <p data-bbox="713 562 845 651">Hama : 1 Rusak : 3</p>	<p data-bbox="917 286 1350 427">Pada frame gambar ini, semua objek dapat terdeteksi dengan baik.</p>
 <p data-bbox="411 954 544 1043">Hama : 1 Rusak : 2</p>	 <p data-bbox="713 954 845 1043">Hama : 2 Rusak : 2</p>	<p data-bbox="917 676 1350 1095">Terdapat juga satu objek yang terdeteksi sebagai hama, hal ini mungkin terjadi karena daun memiliki bentuk yang melengkung, sehingga terdeteksi sebagai hama yang mempunyai ciri atau fitur yang mirip dengan hama pada citra ini.</p>
 <p data-bbox="411 1395 544 1435">Rusak : 5</p>	 <p data-bbox="713 1395 845 1435">Rusak : 5</p>	<p data-bbox="917 1120 1350 1261">Pada frame gambar ini, semua objek dapat terdeteksi dengan baik.</p>
 <p data-bbox="411 1731 544 1771">Rusak : 3</p>	 <p data-bbox="713 1731 845 1771">Rusak : 7</p>	<p data-bbox="917 1456 1350 1762">Terdapat juga empat objek yang terdeteksi sebagai rusak, hal ini mungkin terjadi karena daun yang seharusnya tidak terdeteksi mempunyai ciri atau fitur yang mirip dengan rusak pada citra ini.</p>
		<p data-bbox="917 1798 1350 1939">Pada frame gambar ini, semua objek dapat terdeteksi dengan baik.</p>

Aktual	Prediksi	Keterangan
 <p data-bbox="413 510 544 544">Rusak : 4</p>	 <p data-bbox="711 510 842 544">Rusak : 4</p>	
 <p data-bbox="413 842 544 931">Hama : 1 Rusak : 1</p>	 <p data-bbox="711 842 842 931">Hama : 1 Rusak : 1</p>	<p data-bbox="919 568 1356 707">Pada frame gambar ini, semua objek dapat terdeteksi dengan baik.</p>
 <p data-bbox="413 1229 544 1319">Hama : 2 Rusak : 2</p>	 <p data-bbox="711 1229 842 1319">Hama : 2 Rusak : 2</p>	<p data-bbox="919 956 1356 1095">Pada frame gambar ini, semua objek dapat terdeteksi dengan baik.</p>

## **BAB VI**

### **PENUTUP**

#### **6.1. Kesimpulan**

Berdasarkan rumusan masalah dan hasil analisis, didapatkan kesimpulan sebagai berikut.

1. Dengan menggunakan YOLOv5 dengan parameter pengujian *epoch*, *batch*, dan model YOLOv5, hasil pengujian pada dataset sistem deteksi ini berhasil dalam mengklasifikasikan daun yang rusak dan terinfeksi hama dengan benar. Dengan kemampuannya, sistem ini dapat membantu untuk melakukan penyortiran daun.
2. Dari hasil pengujian, diperoleh bahwa algoritma YOLOv5 menghasilkan nilai mAP sebesar 0,867. Hasil tersebut dianggap cukup baik karena mampu mendeteksi daun pisang yang terkena kerusakan Hama Erionota Thrax L. pada pengujian data.

#### **6.2. Saran**

Berdasarkan hasil analisis yang telah dilakukan, penulis memberikan saran sebagai berikut.

1. Menambah total citra serta variasinya. Variasi yang dimaksud antara lain jenis kelas objek, resolusi gambar, dan waktu pengambilan gambar (siang atau malam) sehingga jaringan yang dibangun dapat mengenali lebih banyak karakteristik objek dalam situasi yang berbeda.
2. Menggunakan algoritma YOLO dengan versi terbaru yang saat ini sudah tersedia diantaranya YOLOv7 dan YOLOv8. Dengan menggunakan algoritma deteksi objek yang terbaru diharapkan dapat memberi perbedaan yang signifikan.

## DAFTAR PUSTAKA

- Ahmad, A. (2017). Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning. *Yayasan Cahaya Islam, Jurnal Teknologi Indonesia*, 1-5.
- Andera, R., Mutiara, M., Studi Proteksi Tanaman, P., Pertanian, F., Bengkulu, U., & Karantina Pertanian Kelas Bengkulu Jalan WR Supratman Kandang Limun Bengkulu, S. I. (2021). Deteksi dan Identifikasi Penyakit Tanaman Pisang (*Musa acuminata*) oleh Cendawan *Cordana Musae* dengan Beberapa Metode Detection and Identification of Banana Plant (*Musa Acuminata*) Disease by *Cordana Musae* with Some Methods. *Universitas Negeri Padang*, 01(2021), 700–709. <https://doi.org/10.24036/proseminasbio/vol1/90>
- Artana, P. Y., Ketut, N., Jayanti, D. A., Made, I., & Saputra, A. B. (2022). Seminar Nasional Corisindo. 407 Sistem Pakar Diagnosa Penyakit dan Hama Tanaman Pisang Menggunakan Metode Forward Chaining Berbasis Android.
- Chen, Z., Wu, R., Lin, Y., Li, C., Chen, S., Yuan, Z., Chen, S., & Zou, X. (2022). Plant Disease Recognition Model Based on Improved YOLOv5. *Agronomy*, 12(2), 1–14. <https://doi.org/10.3390/agronomy12020365>
- Cheng, S., Zhao, K., & Zhang, D. (2019). Abnormal water quality monitoring based on visual sensing of three-dimensional motion behavior of fish. *Symmetry*, 11, 1–20. <https://doi.org/10.3390/sym11091179>
- Cochard, D. (2021, Maret 29). *YOLOv5 : The Latest Model for Object Detection*. Diambil kembali dari <https://medium.com/axinc-ai/yolov5-the-latest-model-for-object-detection-b13320ec516b>
- Geitgey, A. (2016, Juni 14). *Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks*. Diambil kembali dari <https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721>
- I Wayan Suartika E. P, Arya Yudhi Wijaya, & Rully Soelaiman. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101. *Jurnal Teknik ITS*, 5(1), A65–A69.



- Indra Putra, I. L. (2018). Ulat Penggulung Daun Pisang *Erionota thrax* L. (Lepidoptera: Hesperidae) dan Parasitoidnya di Kebun Plasma Nutfah Pisang Yogyakarta. *Gontor Agrotech Science Journal*, 4(2), 125–137. <https://doi.org/10.21111/agrotech.v4i2.2645>
- John, A., Devika, V. L., V, G. S., & Sakhir, N. (2023). Leaf Disease Detection & Correction using YOLO V7 with GPT3 integrated. *International Journal of Engineering Research & Technology (IJERT)*, 12(6), 100–105. [www.ijert.org](http://www.ijert.org)
- Jumadi, J., & Sartika, D. (2021). Pengolahan Citra Digital untuk Identifikasi Objek Menggunakan Metode Hierarchical Agglomerative Clustering. *Jurnal Sains dan Teknologi*, 10(2), 148–156.
- Khalid, M., Sarfraz, M. S., Iqbal, U., Aftab, M. U., Niedbala, G., & Rauf, H. T. (2023). Real-Time Plant Health Detection Using Deep Convolutional Neural Networks. *Agriculture (Switzerland)*, 13(2), 1–26. <https://doi.org/10.3390/agriculture13020510>
- Kintari. (2023). Syarat Ekspor Daun Pisang Ke Luar Negeri. Diambil kembali dari <https://alamatjtcargo.com/syarat-ekspor-daun-pisang/>
- Mellyssa, R. W., Dewi, A. F., Misriana, M., Suryati, S., & Rachmawati, R. (2022). Proceeding Seminar Nasional Politeknik Negeri Lhokseumawe Pengaruh Algoritma Deep Learning dalam Meningkatkan Akurasi. *Proceeding Seminar Nasional Politeknik Negeri Lhokseumawe*, 6(1), 12–16.
- Nisa, N. K. (2023, Maret 23). Cara Kerja Object Detection dengan YOLO (You Only Look Once). Diambil kembali dari <https://pacmann.io/blog/cara-kerja-object-detection-dengan-yolo>
- Nugroho, K. S. (2019, November 13). Confusion Matrix untuk Evaluasi Model pada Supervised Learning. Diambil kembali dari <https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine-learning-bc4b1ae9ae3f>
- Nurjanah, S. (2018). Dampak Ekspor Pisang Indonesia ke Jepang dalam Rangka Indonesia - Japan Economic Partnership Agreement (IJEPA). *JOM FISIP*, 5(1), 1–13. <http://www.mofa.go.jp/region/asia->

- Paath, M. C., Pelealu, J., & Meray, E. R. M. (2018). Jenis dan Persentase Parasitoid Telur Hama Penggulung Daun Pisang (*Erionata thrax* L) (Lepidoptera: Hesperiiidae) pada Beberapa Ketinggian Tempat di Kabupaten Minahasa Species and Percentage of Parasitoid of Banana Leaves (*Erionata thrax* L) (Lepidoptera: Hesperiiidae) at Several Altitude Places in Minahasa District. *Cocos*, 10, 1–11.
- Rahma, L., Syaputra, H., Mirza, A. H., & Purnamasari, S. D. (2021). Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once). *Jurnal Nasional Ilmu Komputer*, 2(3), 213–232.
- Roihan, A., Abas Sunarya, P., & Rafika, A. S. (2020). IJCIT (Indonesian Journal on Computer and Information Technology) Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. *IJCIT (Indonesian Journal on Computer and Information Technology)*, 5(1), 75–82.
- Soeb, M. J. A., Jubayer, M. F., Tarin, T. A., al Mamun, M. R., Ruhad, F. M., Parven, A., Mubarak, N. M., Karri, S. L., & Meftaul, I. M. (2023). Tea leaf disease detection and identification based on YOLOv7 (YOLO-T). *Scientific Reports*, 13(1). <https://doi.org/10.1038/s41598-023-33270-4>
- Sri Ratna Sulistiyanti, Fx Arianto Setyawan, & Muhammad Komarudin. (2016). *Pengolahan Citra Dasar dan Contoh Penerapannya*.
- Subari, W., Goo, N., Siahaya, V. G., & Budidaya, J. (2022). *Agrologia* (Vol. 11, Issue 2).
- Tropika, A. (2014). Inventarisasi Parasitoid Hama Penggulung Daun Pisang (*Erionota thrax* L.) di Kabupaten Lampung Selatan (Vol. 2, Issue 3).
- Wibowo, A., Lusiana, L., & Dewi, T. K. (2023). Implementasi Algoritma Deep Learning You Only Look Once (YOLOv5) untuk Deteksi Buah Segar dan Busuk. *Paspalum: Jurnal Ilmiah Pertanian*, 11(1), 123–130. <https://doi.org/10.35138/paspalum.v11i1.489>
- Widjaja, P. A., & Leonesta, J. R. (2022). Determining Mango Plant Types Using YOLOv4. *Formosa Journal of Science and Technology*, 1(8), 1143–1150. <https://doi.org/10.55927/fjst.v1i8.2155>

Zhang, Z. (2020, Juli 8). *Yolo 2 Explained*. Diambil kembali dari <https://towardsdatascience.com/yolo2-walkthrough-with-examples-e40452ca265f>

## LAMPIRAN

### Lampiran 1 *Script Program Python*

```
!git clone https://github.com/ultralytics/yolov5
%cd yolov5
%pip install -qr requirements.txt
%pip install -q roboflow

import torch
import os
from IPython.display import Image, clear_output
print(f"Setup complete. Using torch {torch._version_}
({torch.cuda.get_device_properties(0).name if
torch.cuda.is_available() else 'CPU'})")

!pip install roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="45hBvAN7lYTEmGjE3ODC")
project = rf.workspace("uii-iqcb4").project("tal-xy7hi")
dataset = project.version(2).download("yolov5")
os.environ["DATASET_DIRECTORY"] = "/content/datasets"
!python train.py --img 640 --batch 8 --epochs 100 --data
{dataset.location}/data.yaml --weights yolov5s.pt -cache
os.environ["DATASET_DIRECTORY"] = "/content/datasets"
!python train.py --img 640 --batch 8 --epochs 50 --data
{dataset.location}/data.yaml --weights yolov5s.pt -cache
os.environ["DATASET_DIRECTORY"] = "/content/datasets"
!python train.py --img 640 --batch 16 --epochs 50 --data
{dataset.location}/data.yaml --weights yolov5s.pt -cache
os.environ["DATASET_DIRECTORY"] = "/content/datasets"
!python train.py --img 640 --batch 8 --epochs 50 --data
{dataset.location}/data.yaml --weights yolov5m.pt -cache
os.environ["DATASET_DIRECTORY"] = "/content/datasets"
!python train.py --img 640 --batch 8 --epochs 50 --data
{dataset.location}/data.yaml --weights yolov5x.pt -cache
%load_ext tensorboard
%tensorboard --logdir runs
```

```

!python detect.py --weights runs/train/exp/weights/best.pt --img
416 --conf 0.1 --source {dataset.location}/test/images
!python detect.py --weights runs/train/exp1/weights/best.pt --img
416 --conf 0.1 --source {dataset.location}/test/images
!python detect.py --weights runs/train/exp2/weights/best.pt --img
416 --conf 0.1 --source {dataset.location}/test/images
!python detect.py --weights runs/train/exp3/weights/best.pt --img
416 --conf 0.1 --source {dataset.location}/test/images
!python detect.py --weights runs/train/exp4/weights/best.pt --img
416 --conf 0.1 --source {dataset.location}/test/images

import glob
from IPython.display import Image, display for imageName in
glob.glob('/content/yolov5/runs/detect/exp/*.jpg'): #assuming JPG
display(Image(filename=imageName))
print("\n")

import glob
from IPython.display import Image, display for imageName in
glob.glob('/content/yolov5/runs/detect/exp1/*.jpg'): #assuming JPG
display(Image(filename=imageName))
print("\n")

#display inference on ALL test images
import glob
from IPython.display import Image, display for imageName in
glob.glob('/content/yolov5/runs/detect/exp2/*.jpg'): #assuming JPG
display(Image(filename=imageName))
print("\n")

#display inference on ALL test images
import glob
from IPython.display import Image, display for imageName in
glob.glob('/content/yolov5/runs/detect/exp3/*.jpg'): #assuming JPG
display(Image(filename=imageName))
print("\n")

import glob
from IPython.display import Image, display for imageName in
glob.glob('/content/yolov5/runs/detect/exp4/*.jpg'):
print("\n")

from google.colab import files
files.download('./runs/train/exp4/weights/best.pt'),

##Kode detect.py

```

Usage - sources:

```
$ python detect.py --weights yolov5s.pt --source 0
img.jpg
vid.mp4
screen
path/
list.txt
list.streams
'path/*.jpg'
'https://youtu.be/Zgi9g1ksQHc'
'rtsp://example.com/media.mp4'
```

Usage - formats:

```
$ python detect.py --weights yolov5s.pt
yolov5s.torchscript
yolov5s.onnx
yolov5s_openvino_model
yolov5s.engine
yolov5s.mlmodel
yolov5s_saved_model
yolov5s.pb
yolov5s.tflite
yolov5s_edgetpu.tflite
yolov5s_paddle_model
"""
import argparse
import os
import platform
import sys
from pathlib import Path
import torch
FILE = Path(__file__).resolve()
ROOT = FILE.parents[0]
if str(ROOT) not in sys.path:
    sys.path.append(str(ROOT))
ROOT = Path(os.path.relpath(ROOT, Path.cwd()))
from utils.dataloaders import IMG_FORMATS, VID_FORMATS,
LoadImages, LoadScreenshots, LoadStreams
```

```

from utils.general import (LOGGER, Profile, check_file,
check_img_size, check_imshow, check_requirements, colorstr, cv2,
increment_path, non_max_suppression, print_args, scale_boxes,
strip_optimizer, xyxy2xywh)
from utils.plots import Annotator, colors, save_one_box
from utils.torch_utils import select_device, smart_inference_mode
@smart_inference_mode()
def run(
weights=ROOT / 'yolov5s.pt',
source=ROOT / 'data/images',
data=ROOT / 'data/coco128.yaml',
imgsz=(640, 640),
conf_thres=0.25,
iou_thres=0.45,
max_det=1000,
device='',
view_img=False,
save_txt=False,
save_conf=False,
save_crop=False,
nosave=False,
classes=None,
agnostic_nms=False,
augment=False,
visualize=False,
update=False,
project=ROOT / 'runs/detect',
name='exp',
exist_ok=False,
line_thickness=3,
hide_labels=False,
hide_conf=False,
half=False,
dnn=False,
vid_stride=1,
):
source = str(source)
save_img = not nosave and not source.endswith('.txt')
is_file = Path(source).suffix[1:] in (IMG_FORMATS + VID_FORMATS)

```

```

is_url = source.lower().startswith(('rtsp://', 'rtmp://',
'http://', 'https://'))
webcam = source.isnumeric() or source.endswith('.streams') or
(is_url and not is_file)
screenshot = source.lower().startswith('screen')
if is_url and is_file:
source = check_file(source)
save_dir = increment_path(Path(project) / name, exist_ok=exist_ok)
(save_dir / 'labels' if save_txt else
save_dir).mkdir(parents=True, exist_ok=True)
device = select_device(device)
model = DetectMultiBackend(weights, device=device, dnn=dnn,
data=data, fp16=half)
stride, names, pt = model.stride, model.names, model.pt
imgsz = check_img_size(imgsz, s=stride)
if webcam:
view_img = check_imshow(warn=True)
dataset = LoadStreams(source, img_size=imgsz, stride=stride,
auto=pt, vid_stride=vid_stride)
bs = len(dataset)
elif screenshot:
dataset = LoadScreenshots(source, img_size=imgsz, stride=stride,
auto=pt)
else:
dataset = LoadImages(source, img_size=imgsz, stride=stride,
auto=pt, vid_stride=vid_stride)
vid_path, vid_writer = [None] * bs, [None] * bs
model.warmup(imgsz=(1 if pt or model.triton else bs, 3, *imgsz))
seen, windows, dt = 0, [], (Profile(), Profile(), Profile())
for path, im, im0s, vid_cap, s in dataset:
with dt[0]:
im = torch.from_numpy(im).to(model.device)
im = im.half() if model.fp16 else im.float()
if len(im.shape) == 3:im = im[None]
with dt[1]:
visualize = increment_path(save_dir / Path(path).stem, mkdir=True)
if visualize else False
pred = model(im, augment=augment, visualize=visualize)
with dt[2]:

```



```

pred = non_max_suppression(pred, conf_thres, iou_thres, classes,
agnostic_nms, max_det=max_det)
for i, det in enumerate(pred):
seen += 1
if webcam:
p, im0, frame = path[i], im0s[i].copy(), dataset.count
s += f'{i}: '
else:
p, im0, frame = path, im0s.copy(), getattr(dataset, 'frame', 0)
p = Path(p) # to Path
save_path = str(save_dir / p.name)
txt_path = str(save_dir / 'labels' / p.stem) + ('' if dataset.mode
== 'image' else f'_{frame}')
s += '%gx%g ' % im.shape[2:]
gn = torch.tensor(im0.shape)[[1, 0, 1, 0]]
imc = im0.copy() if save_crop else im0
annotator = Annotator(im0, line_width=line_thickness,
example=str(names))
if len(det):
det[:, :4] = scale_boxes(im.shape[2:], det[:, :4],
im0.shape).round()
for c in det[:, 5].unique():
n = (det[:, 5] == c).sum()
s += f"{n} {names[int(c)]}'s' * (n > 1)}, "
for *xyxy, conf, cls in reversed(det):
if save_txt:
xywh = (xyxy2xywh(torch.tensor(xyxy).view(1, 4)) / gn).view(-
1).tolist()
line = (cls, *xywh, conf) if save_conf else (cls, *xywh) # label
format
with open(f'{txt_path}.txt', 'a') as f:
f.write((' %g ' * len(line)).rstrip() % line + '\n')
if save_img or save_crop or view_img:
c = int(cls)
label = None if hide_labels else (names[c] if hide_conf else
f'{names[c]} {conf:.2f}')
annotator.box_label(xyxy, label, color=colors(c, True))
if save_crop:

```

```

save_one_box(xyxy, imc, file=save_dir / 'crops' / names[c] /
f'{p.stem}.jpg', BGR=True)
im0 = annotator.result()
if view_img:
if platform.system() == 'Linux' and p not in windows:
windows.append(p)
cv2.namedWindow(str(p), cv2.WINDOW_NORMAL | cv2.WINDOW_KEEPRATIO)
cv2.resizeWindow(str(p), im0.shape[1], im0.shape[0])
cv2.imshow(str(p), im0)
cv2.waitKey(1)
if save_img:
if dataset.mode == 'image':
cv2.imwrite(save_path, im0)
else:
if vid_path[i] != save_path:
vid_path[i] = save_path
if isinstance(vid_writer[i], cv2.VideoWriter):
vid_writer[i].release()
if vid_cap:
fps = vid_cap.get(cv2.CAP_PROP_FPS)
w = int(vid_cap.get(cv2.CAP_PROP_FRAME_WIDTH))
h = int(vid_cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
else:
fps, w, h = 30, im0.shape[1], im0.shape[0]
save_path = str(Path(save_path).with_suffix('.mp4')) # force
*.mp4 suffix on results videos
vid_writer[i] = cv2.VideoWriter(save_path,
cv2.VideoWriter_fourcc(*'mp4v'), fps, (w, h))
vid_writer[i].write(im0)
LOGGER.info(f"{s}{' ' if len(det) else '(no detections),
'}{dt[1].dt * 1E3:.1f}ms")
t = tuple(x.t / seen * 1E3 for x in dt) # speeds per image
LOGGER.info(f'Speed: %.1fms pre-process, %.1fms inference, %.1fms
NMS per image at shape {(1, 3, *imgsz)}' % t)
if save_txt or save_img:
s = f"\n{len(list(save_dir.glob('labels/*.txt')))} labels saved to
{save_dir / 'labels'}" if save_txt else ''
LOGGER.info(f"Results saved to {colorstr('bold', save_dir)}{s}")
if update:

```

```

strip_optimizer(weights[0])
def parse_opt():
    parser = argparse.ArgumentParser()
    parser.add_argument('--weights', nargs='+', type=str, default=ROOT
/ 'best.pt', help='model path or triton URL')
    #parser.add_argument('--weights', nargs='+', type=str, default=ROOT
/ 'recognition_best.pt', help='model path or triton URL')
    parser.add_argument('--source', type=str, default=ROOT /
'0', help='file/dir/URL/glob/screen/0(webcam)')
    parser.add_argument('--data', type=str, default=ROOT /
'data/coco128.yaml', help='(optional)dataset.yaml path')
    parser.add_argument('--imgsz', '--img', '--img-size', nargs='+',
type=int, default=[640], help='inference size h,w')
    parser.add_argument('--conf-thres', type=float, default=0.25,
help='confidence threshold')
    parser.add_argument('--iou-thres', type=float, default=0.45,
help='NMS IoU threshold')
    parser.add_argument('--max-det', type=int, default=1000,
help='maximum detections per image')
    parser.add_argument('--device', default='', help='cuda device,
i.e. 0 or 0,1,2,3 or cpu')
    parser.add_argument('--view-img', action='store_true', help='show
results')
    parser.add_argument('--save-txt', action='store_true', help='save
results to *.txt')
    parser.add_argument('--save-conf', action='store_true', help='save
confidences in --save-txt labels')
    parser.add_argument('--save-crop', action='store_true', help='save
cropped prediction boxes')
    parser.add_argument('--nosave', action='store_true', help='do not
save images/videos')
    parser.add_argument('--classes', nargs='+', type=int, help='filter
by class: --classes 0, or --classes 0 2 3')
    parser.add_argument('--agnostic-nms', action='store_true',
help='class-agnostic NMS')
    parser.add_argument('--augment', action='store_true',
help='augmented inference')
    parser.add_argument('--visualize', action='store_true',
help='visualize features')

```

```

parser.add_argument('--update', action='store_true', help='update
all models')
parser.add_argument('--project', default=ROOT / 'runs/detect',
help='save results to project/name')
parser.add_argument('--name', default='exp', help='save results to
project/name')
parser.add_argument('--exist-ok', action='store_true',
help='existing project/name ok, do not increment')
parser.add_argument('--line-thickness', default=3, type=int,
help='bounding box thickness (pixels)')
parser.add_argument('--hide-labels', default=False,
action='store_true', help='hide labels')
parser.add_argument('--hide-conf', default=False,
action='store_true', help='hide confidences')
parser.add_argument('--half', action='store_true', help='use FP16
half-precision inference')
parser.add_argument('--dnn', action='store_true', help='use OpenCV
DNN for ONNX inference')
parser.add_argument('--vid-stride', type=int, default=1,
help='video frame-rate stride')
opt = parser.parse_args()
opt.imgsz *= 2 if len(opt.imgsz) == 1 else 1
return opt

def main(opt):
check_requirements(ROOT / 'requirements.txt',
exclude=('tensorboard', 'thop'))
run(**vars(opt))
if __name__ == '__main__':
opt = parse_opt()
main(opt)

```