

Perbandingan Deep Learning YOLOv5 dan YOLOv8 Untuk Deteksi Penyakit Daun Tanaman Tomat

Siti Choiriyah[#], Aji Supriyanto^{*}

[#] Sistem Informasi, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank, Semarang, Indonesia

^{*} Teknologi Informasi, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank, Semarang, Indonesia

E-mail: [sitichoiriya\[at\]mhs.unisbank.ac.id](mailto:sitichoiriya[at]mhs.unisbank.ac.id), [ajisup\[at\]edu.unisbank.ac.id](mailto:ajisup[at]edu.unisbank.ac.id)

ABSTRACTS

Agriculture is one of the mainstays of the country's economy, especially the horticulture sub-sector such as fruits and vegetables. Tomato plants are one of the leading commodities. However, the failure of tomato cultivation due to the many types of diseases that exist is still an obstacle and interferes with plant growth, reduces yields, and even causes tomato plant death. This study aims to detect tomato leaf diseases by comparing the performance of the two YOLOv5 and YOLOv8 models. The purpose of comparing models is to determine the level of accuracy and to conclude which version of YOLO provides a better level of accuracy in the hope of helping to determine which method is most appropriate and appropriate to needs. The results showed that both YOLOv5m and YOLOv8m models performed very well in detection. Both models showed high precision, recall, and mAP values. YOLOv8m is better able to detect all objects in the image where the precision value is superior to YOLOv5m. YOLOv8m is superior in precision with a value of 0.95%, a difference of 0.02% with YOLOv5m and mAP50:95 which is 0.92%, a difference of 0.02% with YOLOv5m which means that YOLOv8m is better at identifying objects very precisely and objects of various sizes, but YOLOv8m requires a slightly longer training time than YOLOv5m. YOLOv8m is better able to detect all objects in the image where the precision value is 0.02% superior to YOLOv5m.

Manuscript received Feb 01, 2025;
revised Feb 05, 2024, accepted Mar
09, 2025 Date of publication Mar
31, 2025. International Journal,
JITSI: Jurnal Ilmiah Teknologi
Sistem Informasi licensed under a
Creative Commons Attribution-
Share Alike 4.0 International
License



ABSTRAK

Pertanian merupakan salah satu penunjang perekonomian negara, khususnya sub-sektor hortikultura seperti buah dan sayur. Tanaman tomat menjadi salah satu komoditas unggulan. Namun kegagalan budidaya tanaman tomat karena banyaknya jenis penyakit yang ada masih menjadi kendala dan mengganggu pertumbuhan tanaman, mengurangi hasil panen, dan bahkan menyebabkan kematian tanaman tomat. Penelitian ini bertujuan untuk mendeteksi penyakit daun tanaman tomat dengan membandingkan kinerja kedua model YOLOv5 dan YOLOv8. Tujuan membandingkan model adalah untuk menentukan tingkat akurasi dan dapat menyimpulkan versi YOLO yang memberikan tingkat akurasi yang lebih baik dengan harapan dapat membantu dalam menentukan metode mana yang paling tepat dan sesuai dengan kebutuhan. Hasil penelitian menunjukkan kedua model YOLOv5m dan YOLOv8m memberikan kinerja yang sangat baik dalam mendeteksi. Kedua model menunjukkan nilai presisi, recall, dan mAP yang tinggi. YOLOv8m lebih mampu mendeteksi semua objek yang ada dalam gambar dimana nilai presisi lebih unggul dibanding YOLOv5m. YOLOv8m lebih unggul dalam presisi dengan nilai 0.95% selisih 0.02% dengan YOLOv5m dan mAP50:95 yaitu 0.92% selisih 0.02% dengan YOLOv5m yang berarti bahwa YOLOv8m lebih baik dalam mengidentifikasi objek dengan sangat tepat dan objek dengan berbagai ukuran, namun YOLOv8m membutuhkan waktu pelatihan sedikit lebih lama

dibanding YOLOv5m. YOLOv8m lebih mampu mendeteksi semua objek yang ada dalam gambar dimana nilai presisi lebih unggul 0.02% dibanding YOLOv5m.

Keywords / Kata Kunci — *Deteksi; Penyakit Daun Tomat; Pertanian; YOLOv5; YOLOv8*

CORRESPONDING AUTHOR

Siti Choiriyah
Sistem Informasi, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank, Semarang, Indonesia
Email: sitichoairyah[at]mhs.unisbank.ac.id

1. PENDAHULUAN

Pertanian merupakan industri penting dalam menunjang kehidupan masyarakat dan perekonomian nasional[1]. Sub-sektor hortikultura, terutama dalam produksi buah-buahan dan sayuran memberikan kontribusi penting terhadap Produk Domestik Bruto (PDB) negara[2]. Dalam sektor ini tanaman tomat menjadi salah satu komoditas unggulan dalam membudidayakannya[3]. Menurut data Badan Pusat Statistik(BPS) produksi tomat pada tahun 2021 mencapai 1.114.399 ton meningkat sebesar 2,71 persen dari tahun 2020. Konsumsi tomat pada sektor rumah tangga tahun 2021 mencapai 677,97 ribu ton, meningkat sebesar 6,93 persen dari tahun 2020[4]. Dalam mendukung ketahanan pangan dan ekonomi khususnya pada sektor hortikultura pentingnya menjaga kesehatan tanaman harus diperhatikan secara khusus. Tanaman yang terhindar dari hama dan berproduksi secara optimal dapat memenuhi kebutuhan gizi masyarakat dengan kualitas yang tinggi. Tanaman tomat selain mudah dalam budidayanya tetapi juga rentan terkena penyakit, salah satunya penyakit daun[5]. Penyakit pada daun dapat mengganggu pertumbuhan tanaman, mengurangi hasil panen, dan bahkan menyebabkan kematian tanaman tomat[6]. Sehingga permasalahan yang akan timbul berupa kerugian ekonomi yang signifikan serta berkurangnya ketersediaan pasokan di pasar[2].

Dalam proses pertumbuhan tanaman, daun tomat memiliki peran dan fungsi yang sangat penting, sehingga ketika daun tanaman tomat terserang penyakit dapat mempengaruhi pertumbuhan tanaman secara keseluruhan dan akan mempengaruhi hasil produksi panen. Jenis penyakit pada daun dapat dikenali dari gejalanya yang berbeda-beda, meskipun gejala tersebut tidak selalu dapat menjelaskan dengan tepat nama penyakit yang menyerang. Hal ini disebabkan oleh banyaknya jenis penyakit yang memiliki gejala serupa dan hampir terlihat sama[7]. Beberapa penyakit daun tanaman tomat seperti early blight, late blight, leaf mold, septoria, mosaic virus, yellow leaf curl virus, dan spider mites[8]. Identifikasi penyakit pada daun tanaman tomat umumnya dilakukan di laboratorium, yang membutuhkan waktu dan biaya yang cukup besar[7].

Perkembangan teknologi yang semakin pesat bisa dimanfaatkan untuk menangani masalah terutama di bidang pertanian. Oleh karena itu penulis memanfaatkan teknologi terkini seperti identifikasi gambar atau objek menggunakan metode deep learning You Only Look Once(YOLO). YOLO merupakan algoritma deep learning untuk mendeteksi objek yang dikembangkan tahun 2016. YOLO memproses sebuah gambar sekali dan membagi menjadi grid, memprediksi bounding box dan probabilitas objek[9]. Algoritma ini cepat dan akurat dalam mendeteksi objek[10][11]. YOLO mengalami seri pengembangannya, salah satunya yaitu YOLO v5 yang dikenal sebagai model yang kuat dan efisien dalam pencitraan terutama dalam bidang medis, keamanan, dan otonom kendaraan karena kemampuan memproses objek secara real-time. Generasi pengembangan YOLO v5 terus berkembang hingga versi terbarunya, YOLO v8, yang melampaui kinerja versi sebelumnya dan menunjukkan potensi besar untuk digunakan dalam berbagai bidang[12]. Dengan ini peneliti menerapkannya dalam deteksi penyakit daun tanaman tomat dengan membandingkan kinerja kedua model dengan harapan dapat membantu dalam menentukan metode mana yang paling tepat dan sesuai dengan kebutuhan.

Penelitian terkait klasifikasi dan identifikasi penyakit tanaman banyak yang telah melakukan penelitian menggunakan computer vision. Penelitian terdahulu dengan menggunakan YOLO v5 dalam mengklasifikasikan 3 kelas penyakit daun durian dengan nilai akurasi sebesar 85%[13]. Selanjutnya penelitian terdahulu dengan menggunakan YOLO v8 untuk deteksi dan klasifikasi hama potato beetle pada tanaman kentang dengan nilai mean average precision(mAP50) sebesar 81,8% yang menunjukkan tingkat akurasi deteksi yang baik[14]. Penelitian terdahulu selanjutnya dengan menggunakan convolutional neural network(CNN) dalam mendiagnosa tujuh kelas penyakit daun tanaman tomat dengan nilai akurasi sebesar 93%[7]. Penelitian selanjutnya menggunakan YOLO v5 dalam klasifikasi penyakit daun tanaman padi hasil evaluasi model menunjukkan nilai mAP tertinggi sebesar 95%, akurasi sebesar 88%, dan recall sebesar 100%. Tahap terakhir, pengujian sistem secara real-time dengan webcam, menghasilkan akurasi pengujian untuk kelas Narrow Brown Spot sebesar 93%, Hawar Daun sebesar 81%, dan Daun Padi Normal sebesar 91%[1]. Penelitian terdahulu selanjutnya dengan melakukan tiga percobaan dengan pembagian data yang berbeda. Percobaan pertama membagi data pada 70:20:10, percobaan kedua membagi data pada 75:15:10, dan percobaan ketiga membagi data pada 80:10:10. Pada percobaan ini, nilai rata-rata untuk pengujian adalah 0,947 dengan penjabaran nilai presisi, recall, dan mAP masing-masing sebesar 0.946, 0.936, dan 0.959[15]. Keberagaman dan jumlah data latih yang digunakan adalah

faktor penting dalam penelitian ini[16]. Data latih yang berkualitas dan bervariasi akan sangat mempengaruhi kinerja model. Dengan jumlah data latih yang cukup, model akan dapat mempelajari karakteristik unik dari berbagai jenis penyakit daun tomat. Jumlah epoch atau iterasi pelatihan juga penting. Terlalu sedikit waktu dapat menyebabkan model underfitting, sedangkan terlalu banyak waktu dapat menyebabkan overfitting. Epoch berarti satu melewati seluruh kumpulan data dan memberi nilai waktu ke waktu berikutnya. Pemilihan waktu yang tepat diperlukan untuk menghasilkan model yang baik untuk mendapatkan akurasi yang cukup baik[11].

Penelitian ini bertujuan untuk membandingkan kedua metode dan dapat menyimpulkan versi YOLO yang memberikan tingkat akurasi yang lebih baik dalam mendeteksi penyakit daun tanaman tomat dengan harapan dapat membantu dalam menentukan metode mana yang paling tepat dan sesuai dengan kebutuhan.

2. METODOLOGI PENELITIAN

1. Tahapan Penelitian

Proses penting dan sistematis untuk menghasilkan pengetahuan baru atau solusi masalah adalah tahapan penelitian[17]. Berikut ini adalah tahapan penelitian pada penelitian ini:

a. Perumusan Masalah

Tahapan perumusan masalah peneliti mengidentifikasi dan merumuskan masalah yang akan ditangani, serta menentukan metode atau algoritma yang akan digunakan untuk menangani masalah tersebut secara efektif. Selain itu, pada tahap ini, peneliti juga merencanakan dan menentukan model penelitian yang akan digunakan untuk membantu memecahkan masalah tersebut. Untuk memastikan penelitian yang efektif dan efisien, dan rumusan masalah harus jelas dan tepat.

b. Studi Literatur

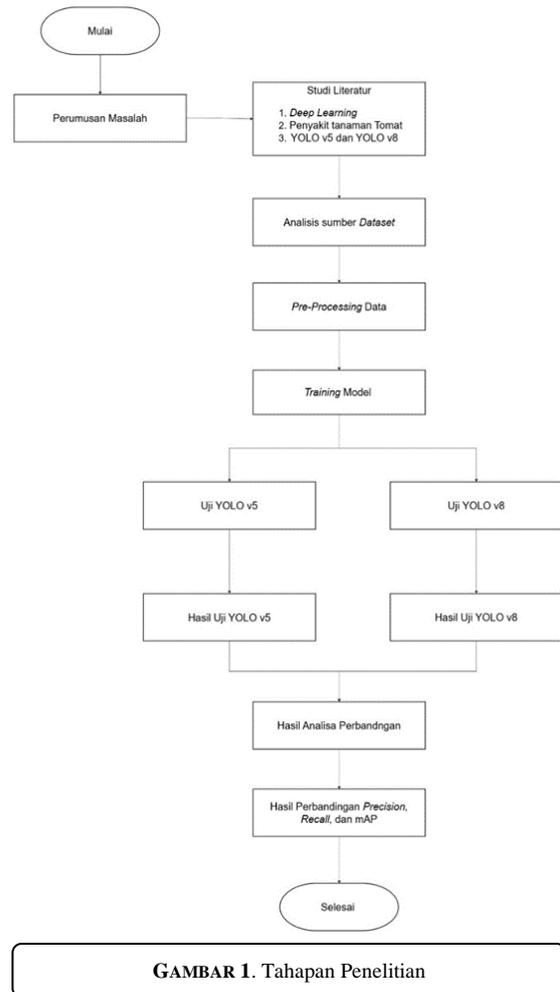
Tahapan ini merupakan proses penelitian mendalam terkait topik penelitian, dipelajari melalui peninjauan dan analisis literatur sebelumnya yang berkaitan dengan subjek penelitian. Mempelajari teori, penelitian sebelumnya tentang deep learning, deteksi objek, You Only Look Once(YOLO), dan penyakit daun tomat adalah bagian dari tahap ini. Tahapan ini sangat penting karena membantu membangun dasar teoretis yang kuat untuk penelitian dan menunjukkan kesalahan yang mungkin telah ditemukan dalam penelitian sebelumnya. Untuk mendapatkan pemahaman yang mendalam dan informasi yang relevan untuk mendukung dan memvalidasi penelitian, peneliti mempelajari berbagai sumber informasi seperti buku, skripsi, jurnal, artikel online, dan publikasi ilmiah lainnya.

c. Analisis Sumber *Dataset*

Pada penelitian ini peneliti menggunakan dataset melalui platform publik universe.roboflow.com(<https://universe.roboflow.com/personal-enkcf/tomato-leaf-disease-detection-gyozv>) yang mencakup jumlah data sebanyak 10.594 dengan ukuran 640x640 imgsz yang terdiri dari 9 kelas dimana 8 kelas penyakit daun tanaman tomat yaitu *early blight*, *late blight*, *leaf mold*, *septoria*, *mosaic virus*, *yellow leaf curl virus*, *leaf miner*, dan *spider mites* dan 1 kelas merupakan daun sehat atau normal.

d. *Pre-Processing Dataset*

Dataset public pada platform roboflow merupakan data yang sudah siap untuk diolah. Pada tahapan ini peneliti melakukan analisis dalam mempersiapkan data sebelum dilakukan proses training, pada tahapan ini peneliti menggunakan versi terakhir dari total versi yaitu 6, peneliti menggunakan versi terakhir dikarenakan pembagian data train, valid dan test yang lebih unik dari versi lainnya. Berikut ini gambar dari pembagian data *training*, *valid*, dan *test*.



GAMBAR 1. Tahapan Penelitian

e. Tahapan *Training*

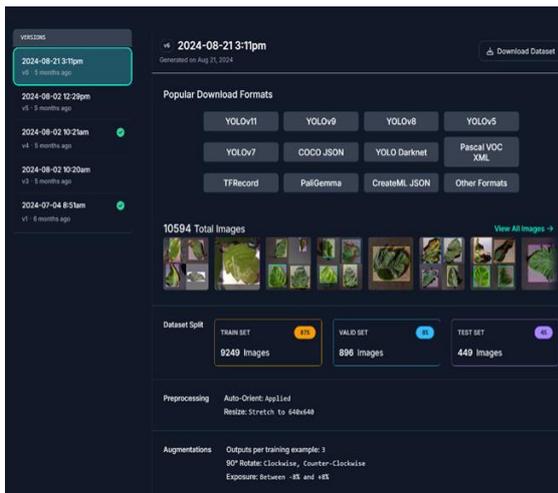
Pada tahap ini merupakan proses menguji kemampuan model yang telah dikembangkan untuk mengenali pola dan membuat prediksi. Data yang telah di-label dengan baik dimasukkan ke dalam model untuk mengajarkan sistem cara menafsirkan data tersebut.

f. Tahapan Analisa Perbandingan

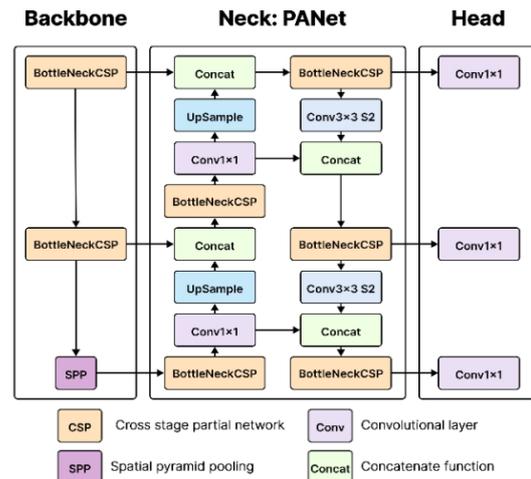
Setelah mendapatkan hasil dari training selanjutnya adalah melakukan analisis terkait kedua model untuk mendapatkan hasil dan kesimpulan.

2. YOLOv5

Algoritma YOLO (You Only Look Once) adalah algoritma deep learning untuk mendeteksi objek yang menggunakan jaringan saraf tunggal pada gambar secara keseluruhan[18]. Ini berbeda dari algoritma lainnya karena menggunakan pengklasifikasi atau pencari lokasi yang digunakan kembali untuk mendeteksi objek. Pola diterapkan pada gambar dalam berbagai posisi dan skala. Area dengan kinerja terbaik pada gambar akan di asumsikan sebagai area deteksi[1][15]. YOLO v5 adalah pembaharuan dari YOLO v4 yang dikembangkan oleh tim Ultralytics tahun 2020 dengan mengubah backbone YOLO dari darknet ke pytorch untuk meningkatkan kemampuan pendeteksian ancor box. Terdiri dari 3 bagian utama yaitu *backbone*, *neck*, dan *head*. *Backbone* terdiri dari BottleNeckCSP yang berfungsi mengekstrak fitur utama dari gambar yang di *input*, hasil dari ekstraksi ini akan menjadi dasar untuk proses deteksi. *Neck* menggunakan arsitektur PANet yang berfungsi menggabungkan fitur-fitur yang diekstrak oleh backbone sehingga menghasilkan representasi fitur yang lebih banyak dari berbagai tingkatan, sehingga model dapat mendeteksi objek dengan ukuran yang bervariasi. Hasil dari proses ini akan dilanjutkan oleh head untuk proses deteksi. *Head* terdiri dari *convolutional layer* yang menghasilkan *output* berupa tensor yang berisi informasi kotak *bounding box* dan probabilitas kelas untuk setiap sel grid pada gambar. *Output* dari *head* akan diproses lebih lanjut sehingga menghasilkan hasil akhir berupa deteksi objek. YOLOv5 masih merupakan solusi terbaik untuk deteksi objek lanskap, meskipun ada beberapa masalah yang terus-menerus, seperti redundansi peta fitur dan target yang meleset dalam situasi tertentu[19].

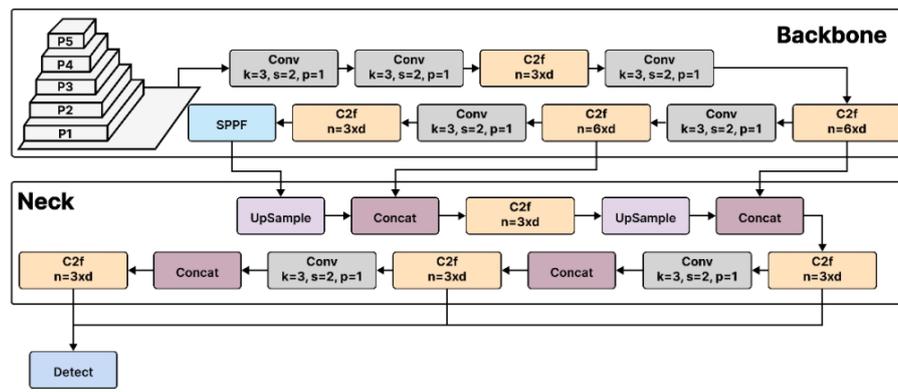


GAMBAR 2. Pembagian data training, valid dan test



GAMBAR 3. Arsitektur YOLO v5

3. YOLOv8



GAMBAR 4. Arsitektur YOLO v8

YOLO v8 salah satu peningkatannya adalah peralihan deteksi *anchor-free* dari metode *anchor-box* sebelumnya. Modifikasi ini memungkinkan prediksi langsung pusat objek, mengaktifkan proses *Non-Maximum Suppression* (NMS), dan mengurangi masalah *anchor-box* seperti kekurangan generalisasi dan masalah penanganan bentuk yang tidak beraturan[19]. YOLOv8 mengubah konvolusi 6×6 awal di stem menjadi 3×3 yang lebih efisien, dan memperbarui blok penyusun inti dengan menukar C3 dengan C2f. Selain itu, fitur-fitur dirangkai langsung di leher model tanpa memaksakan dimensi saluran yang sama ini berarti bahwa jumlah parameter dan ukuran tensor berkurang. Kemampuan model untuk mengelola objek dengan skala yang beragam ditingkatkan dengan fitur baru Spatial Pyramid Pooling Feature (SPPF)[19]. Selama pelatihan, YOLOv8 juga menggabungkan augmentasi gambar *daring*, seperti *augmentasi mosaik*. Fitur ini memungkinkan model dapat mempelajari objek pada posisi baru dan dalam berbagai kondisi, seperti oklusi parsial dan latar belakang yang bervariasi[19].

4. Metriks Evaluasi

Confusion matrix adalah matriks yang menampilkan prediksi klasifikasi aktual dan prediksi klasifikasi yang diprediksi untuk mengevaluasi kinerja model machine learning[20]. Dalam mengevaluasi kinerja model *You Only Look Once* (YOLO), penelitian ini menggunakan matriks *confusion*, yang memungkinkan untuk membandingkan nilai aktual dan nilai prediksi. Untuk mengukur performa model dalam membedakan kelas penyakit pada gambar daun tomat, metrik seperti *precision*, *recall*, dan mAP dihitung berdasarkan hasil *confusion matrix*[2]. Berikut rumus persamaan dari *confusion matrix*:

$$precision = \frac{TP}{TP + FP} \quad (1)$$

$$recall = \frac{TP}{TP + FN} \quad (2)$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \quad (3)$$

$$mAP = \frac{1}{k} \sum_i^k A_{pi} \quad (4)$$

Precision adalah rasio *True Positif* terhadap jumlah data positif yang diprediksi secara keseluruhan. Ketepatan model prediksi diukur dengan *precision*[21]. Variabel *False Positif* berfungsi sebagai pembagi pada penyebut. *Recall* menunjukkan jumlah kejadian positif yang benar-benar diidentifikasi oleh model. Ini pada dasarnya mengukur kemampuan model untuk menemukan dan mengumpulkan peristiwa objek dengan benar dalam bidang deteksi objek. *F1-score* adalah nilai rata-rata dari *precision* dan *recall*. Dimana nilai terbaik dari *F1-score* adalah 1,0 dan 0,0 adalah nilai terburuknya[19]. *Mean average precision* (mAP) adalah mean dari seluruh nilai *Average Precision*(AP) disetiap kelas. Semakin besar nilai mAP menandakan semakin bagus kinerja sebuah model

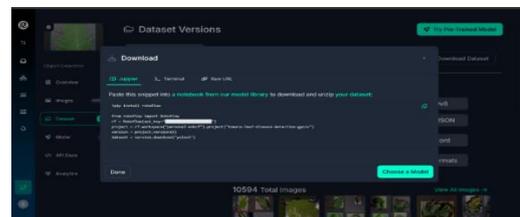
3. HASIL DAN PEMBAHASAN

1. Analisis Sumber Dataset

Data yang digunakan pada penelitian ini merupakan data yang diperoleh penulis dari dataset public <https://universe.roboflow.com/personal-enkcf/tomato-leaf-disease-detection-gyozv>. Dengan jumlah data sebanyak 10.594 dengan ukuran 640×640 imgsz yang terdiri dari 9 kelas dimana 8 kelas penyakit daun tanaman tomat yaitu early blight, late blight, leaf mold, septoria, mosaic virus, yellow leaf curl virus, leaf miner, dan spider mites dan 1 kelas merupakan daun sehat atau normal. Tabel 1 pembagian dataset train, valid dan test.

TABEL 1. Pembagian dataset train, valid dan test

Folder	Label	Presentase
Train	9.249	87%
Valid	896	8%
Test	449	4%



GAMBAR 5. Hasil code download dataset YOLOv5

2. pre-processing dataset

Setelah analisa data selanjutnya melakukan *ekspor* ke YOLO v5 dan YOLO v8 untuk melihat kode yang dapat digunakan saat menginstalasi kedua model dan menyalin *repository* untuk dilakukan training menggunakan google colab.

Setelah mendapatkan semua *source code* dari kedua model dan disimpan dalam google colab sesuai dengan folder dari masing-masing model, langkah selanjutnya adalah menginstall *requirement* dari masing-masing model. Berikut adalah *Source code* untuk menginstall YOLOv5 dan *Source code dataset* dari YOLO v5 untuk di *install*.

Source code install YOLOv5

```
#clone YOLOv5 and
!git clone https://github.com/ultralytics/yolov5 # clone repo
%cd yolov5
%pip install -qr requirements.txt # install dependencies
%pip install -q roboflow

import torch
import os
from IPython.display import Image, clear_output # to display images

print(f"Setup complete. Using torch {torch.__version__} ({torch.cuda.get_device_properties(0).name if torch.cuda.is_available() else 'CPU'})")
```

Source code instalasi Dataset YOLOv5 ke google colab

```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="9TBsAa9C1i7pbEM6BXQW")
project = rf.workspace("personal-enkcf").project("tomato-leaf-disease-detection-gyozv")
version = project.version(6)
dataset = version.download("yolov5")
```

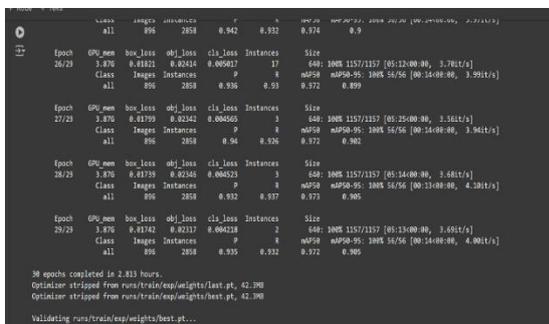
Lakukan hal yang sama pada model YOLO v8. Untuk selanjutnya melakukan proses *training dataset*.

3. *Training dataset*

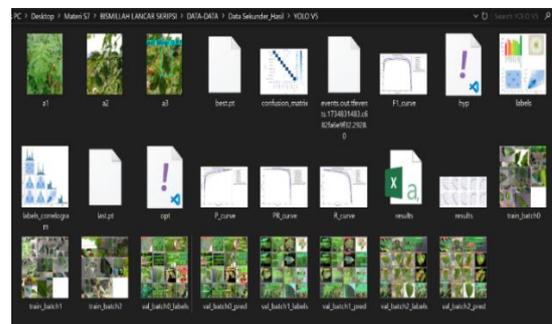
Pelatihan *dataset* menggunakan YOLO v5 dan YOLO v8 dilakukan setelah tahapan *pre-processing* selesai. Pelatihan *dataset* digunakan untuk melatih model dalam pengajaran mesin dengan tujuan mengajarkan pola atau informasi yang diperlukan model untuk membuat prediksi yang akurat ketika dihadapkan pada data baru. Source code dibawah merupakan *source code* untuk *training YOLO v5* untuk penelitian ini, model YOLOv5m dipilih karena berhubungan dengan kecepatan dan efisiensi pelatihan. Model YOLOv5m dibangun sebagai versi mini dari arsitektur YOLOv5, yang membuatnya lebih ringan dan lebih cepat dibandingkan dengan model lain seperti YOLOv5s, YOLOv5n, YOLOv5L, dan YOLOv5x. YOLOv5m jumlah parameter yang lebih sedikit dan struktur yang lebih sederhana. *Img* adalah ukuran gambar yang digunakan dan diinginkan saat pelatihan, *epoch* adalah jumlah iterasi atau berapa kali model mempelajari data pelatihan, *batch* adalah jumlah data, dan file yml berfungsi sebagai penghubung antara *dataset* gambar dan proses pelatihan. *Weights* adalah berat model awal yang akan digunakan sebagai titik awal pelatihan. Gambar 6. Hasil *training* menggunakan 30 *epoch* memerlukan waktu selama kurang lebih 2 jam 48 ment 47 detik. Sedangkan gambar 7. Merupakan folder hasil *training* yang sudah dilakukan dan peneliti unduh untuk dilakukan analisis.

Source code untuk *training YOLOv5*

```
!python train.py --img 640 --batch 8 --epochs 30 --data /content/yolov5/Tomato-Leaf-Disease-Detection-6/data.yaml --weights yolov5m.pt
```



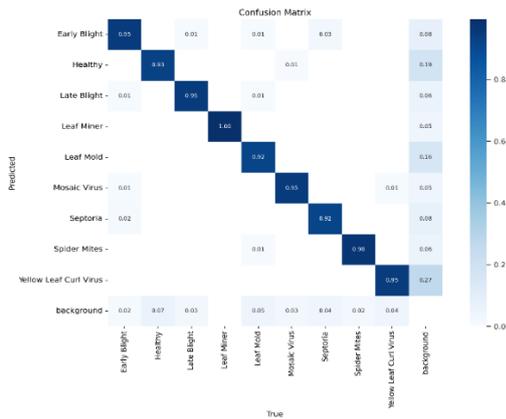
GAMBAR 6. Hasil *training YOLO v5*



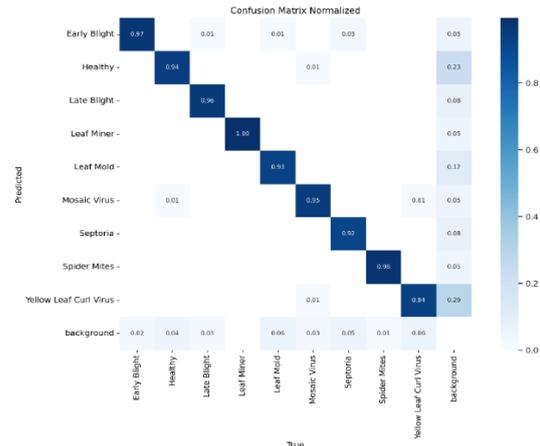
GAMBAR 7. Folder Hasil *training YOLO v5*

4. Hasil Evaluasi Model

Confusion matriks memberikan gambaran tentang kemampuan model untuk mengklasifikasikan objek dengan benar. Matriks berisi jumlah prediksi yang benar (positif benar dan negatif benar) dan kesalahan prediksi (positif palsu dan negatif palsu)[12]. *Training* model menggunakan 30 *epoch*, 8 *batch*, dan 640x640 *img*.



GAMBAR 8. Confusion matriks YOLO v5



GAMBAR 9. Confusion matriks YOLO v8

Hasil evaluasi model pada gambar diatas menunjukkan bahwa nilai prediksi yang diperoleh dalam setiap kelas memiliki nilai *true* positif yang dimana kelas *early blight*, *late blight*, *leaf mold*, *septoria*, *mosaic virus*, *yellow leaf curl virus*, *spider mites*, dan *healthy* memiliki nilai *true* positif lebih dari sama dengan 0.92 berbeda dengan kelas *leaf miner* yang menunjukkan nilai *true* positif sempurna yaitu 1.00 yang berarti bahwa model berhasil mengklasifikasikan 100% gambar *leaf miner* dengan benar. Sedangkan angka diluar diagonal utama menunjukkan kesalahan model dalam mengklasifikasi, sebagai contoh nilai 0.01 pada *early blight*(pada baris 1, kolom 3) menunjukkan sebesar 1% gambar *early blight* salah diklasifikasikan sebagai *late blight*.

Hasil evaluasi model pada gambar diatas menunjukkan bahwa nilai prediksi yang diperoleh dalam setiap kelas memiliki nilai *true* positif yang dimana kelas *early blight*, *late blight*, *leaf mold*, *septoria*, *mosaic virus*, *yellow leaf curl virus*, *spider mites*, dan *healthy* memiliki nilai *true* positif lebih dari sama dengan 0.92 dan kelas *leaf miner* menunjukkan nilai *true* positif nya sempurna sebesar 1.00 yang berarti bahwa model berhasil mengklasifikasikan 100% gambar *leaf miner* dengan benar. Sedangkan angka diluar diagonal utama menunjukkan kesalahan model dalam mengklasifikasi, sebagai contoh nilai 0.03 pada *early blight*(pada baris 1, kolom 7) menunjukkan sebesar 3% gambar *early blight* salah diklasifikasikan sebagai *septoria*. Selanjutnya adalah evaluasi performa *dataset*.

Hasil dari evaluasi performa *dataset* berupa *precision*, *recall*, *mAP* yang diperoleh dari proses *training* dengan 30 *epoch* dapat dilihat pada gambar berikut.

Class	Images	Instances	P	R	mAP50	mAP50-95; 100%
all	896	2858	0.935	0.932	0.972	0.985
Early Blight	896	251	0.918	0.941	0.97	0.983
Healthy	896	273	0.886	0.985	0.944	0.856
Late Blight	896	354	0.965	0.926	0.983	0.932
Leaf Miner	896	257	0.966	0.993	0.994	0.943
Leaf Mold	896	361	0.988	0.986	0.968	0.9
Mosaic Virus	896	367	0.97	0.946	0.975	0.93
Septoria	896	369	0.959	0.911	0.978	0.919
Spider Mites	896	292	0.967	0.869	0.932	0.926
Yellow Leaf Curl Virus	896	334	0.877	0.889	0.958	0.838

GAMBAR 10. Evaluasi Hasil Training YOLO v5

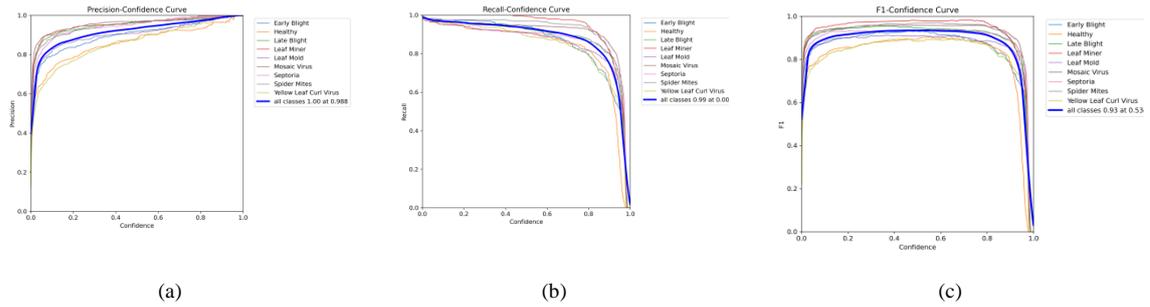
Class	Images	Instances	Box(P	R	mAP50	mAP50-95); 100%
all	896	2858	0.954	0.925	0.976	0.925
Early Blight	211	251	0.923	0.956	0.983	0.945
Healthy	73	273	0.985	0.984	0.933	0.859
Late Blight	245	354	0.963	0.95	0.987	0.95
Leaf Miner	283	257	0.974	0.992	0.995	0.964
Leaf Mold	221	361	0.961	0.878	0.971	0.921
Mosaic Virus	288	367	0.983	0.932	0.98	0.96
Septoria	236	369	0.982	0.88	0.981	0.935
Spider Mites	154	292	0.974	0.959	0.992	0.947
Yellow Leaf Curl Virus	178	334	0.923	0.877	0.958	0.845

GAMBAR 11. Evaluasi Hasil Training YOLO v8

Gambar diatas menunjukkan bahwa evaluasi performa model YOLO v5 pada *dataset* yang digunakan penelitian ini dengan mengukur hasil data *train* dengan parameter *precision*, *recall*, *mAP@IoU 0.5* dan *mAP@IoU 0.5:95* dengan nilai seluruhnya berturut-turut yaitu 0.93%, 0.93%, 0.97%, dan 0.90% menunjukkan bahwa model mampu mendeteksi objek dengan akurasi yang baik. Tabel *class* merupakan nama setiap kelas penyakit daun tomat saat dievaluasi, *images* merupakan jumlah total gambar yang sedang dievaluasi sedangkan *instances* merupakan jumlah total objek yang ada pada gambar. Nilai *mAP50-95* lebih rendah dari *mAP50*, menunjukkan bahwa model mungkin sedikit kesulitan menemukan objek dengan tingkat tumpang tindih yang sangat tinggi. Variasi dalam *dataset* atau kompleksitas kelas penyakit tertentu adalah beberapa dari berbagai penyebab hal ini.

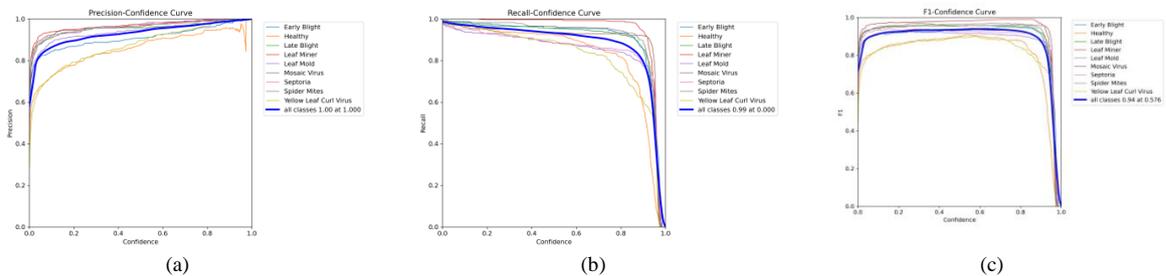
Hasil dari gambar 11. menunjukkan bahwa evaluasi performa pada *dataset* yang digunakan dalam penelitian ini mengukur hasil data latih dengan parameter *precision*, *recall*, *mAP@IoU 0.5* dan *mAP@IoU 0.5:95* dengan nilai seluruhnya berturut-turut yaitu 0.95%, 0.92%, 0.97%, dan 0.92%. Nilai tinggi *mAP50* dan *mAP50-95* menunjukkan bahwa model mendeteksi objek dan menempatkan *bounding box* dengan sangat akurat. Semakin besar nilainya atau hampir 100%, maka kinerja model YOLO semakin baik.

Pada gambar berikut gambar *Precision-Confidence Curve* (a), *Recall-Confidence Curve* (b), dan *F1-Confidence Curve* (c) diperoleh dari proses pelatihan model YOLO dengan 30 *epoch*, 8 *batch*, dan ukuran gambar 640 x 640.



GAMBAR 12. Grafik *precision-confidence curve*(a) grafik *recall-confidence curve*(b) grafik *F1-confidence curve*(c) YOLO v5

Grafik *precision-confidence curve*(a) menggambarkan hubungan antara nilai *precision* dan nilai *confidence*, grafik ini disebabkan oleh banyaknya prediksi yang benar tetapi juga banyak prediksi yang salah, grafik ini juga menunjukkan hubungan antara presisi dan tingkat kepercayaan akan bergerak dari angka nol ke angka satu. Seiring dengan meningkatnya tingkat kepercayaan, jumlah prediksi yang benar juga akan meningkat, sehingga nilai yang tepat juga akan meningkat, meskipun jumlah objek yang diprediksi akan menurun, dimana pada hasil *training* nilai presisi menghasilkan rata-rata nilai maksimal 1.00 pada nilai *confidence* 0.988, nilai ini menunjukkan bahwa pada titik *confidence* model telah mencapai tingkat presisi tertinggi yang berarti bahwa semua prediksinya benar tanpa kesalahan positif. Pada grafik *recall-confidence curve*(b) *Recall* adalah teknik untuk menghitung rasio prediksi positif yang benar dari hasil prediksi dibandingkan dengan semua nilai sebenarnya yang positif (*true positif + false negative*). Gambar yang menunjukkan hubungan antara *recall* dan tingkat kepercayaan akan menunjukkan penurunan, dengan nilai *recall* yang lebih rendah ditunjukkan oleh tingkat kepercayaan yang lebih tinggi. Dari hasil *training* didapatkan nilai rata-rata maksimal 0.99 terhadap nilai *confidence* 0.000 yang menunjukkan bahwa model dapat mengingat kembali dengan sangat baik *instance* positif yang sebenarnya pada tingkat keyakinan yang sangat rendah(0.00). Untuk grafik *F1-confidence curve*(c) merupakan hubungan antara nilai keyakinan dan skor F1 yang menunjukkan kemampuan model untuk menyeimbangkan presisi dan *recall* pada berbagai tingkat keyakinan dalam prediksi, kurva ini penting karena menunjukkan titik optimal di mana model menyelesaikan antara tidak melewatkan kelas yang diinginkan (*recall* tinggi) dan mengklasifikasikan kelas yang bukan target dengan benar (*precision* tinggi). Dari hasil *training* yang diperoleh, F1 menghasilkan nilai 0.93 pada nilai *confidence* 0.534 yang berarti bahwa model mampu menyeimbangkan antara presisi dan *recall* dan model mampu mencapai keseimbangan yang optimal.



GAMBAR 13. Grafik *precision-confidence curve*(a) grafik *recall-confidence curve*(b) grafik *F1-confidence curve*(c) YOLO v8

Hasil dari grafik *precision-confidence curve*(a) pada hasil *training* menunjukkan bahwa presisi menunjukkan nilai maksimal 1.00 pada nilai *confidence* 1.00 nilai ini menunjukkan bahwa pada titik *confidence* model telah mencapai tingkat presisi tertinggi yang berarti bahwa semua prediksinya benar tanpa kesalahan positif. Pada grafik *recall-confidence curve*(b) dari hasil *training* didapatkan nilai 0.99 terhadap nilai *confidence* 0.000 yang menunjukkan bahwa model dapat mengingat kembali dengan sangat baik dengan tingkat kepercayaan yang tinggi. Untuk grafik *F1-confidence curve*(c) dari hasil *training* yang diperoleh, F1 menghasilkan nilai 0.94 pada nilai *confidence* 0.576 yang menunjukkan model mampu menangkap sebagian besar positif benar dan mungkin juga menangkap positif palsu.

5. Hasil Analisa Perbandingan

Hasil untuk seluruh kelas penyakit daun tanaman tomat dari kedua model dapat dilihat dari tabel 2. Dari tabel diatas dapat disimpulkan bahwa nilai *precision* kedua model selisih 0.02% menunjukkan bahwa kedua model dapat menghindari klasifikasi *false positive* dengan baik, YOLOv8m sedikit lebih baik daripada YOLOv5m dalam hal ini. Nilai *recall* YOLOv8m sedikit lebih rendah selisih 0,01% daripada YOLOv5m, yang berarti bahwa YOLOv5m memiliki kemampuan yang lebih baik untuk mendeteksi semua objek positif yang ada dalam gambar. Kedua model memiliki nilai mAP@IoU 0.5 yang sama, yang menunjukkan kinerja yang seimbang dalam

mendeteksi objek dengan ambang *Intersection over Union*(IoU) 0,5. Namun, YOLOv8m sedikit lebih baik pada rentang IoU yang lebih luas (mAP@IoU 0.5:0.95) 92% yang menunjukkan bahwa model mendeteksi objek dengan IoU yang lebih tinggi sedikit sehingga lebih akurat dalam mendeteksi objek

TABEL 2. Hasil *precision, recall, mAP*

Model	Precision	Recall	mAP@IoU0.5	mAP@IoU0.5:0.95	Trainng Time(hours)
YOLOv5m	0.93%	0.93%	0.97%	0.90%	2.813
YOLOv8m	0.95%	0.92%	0.97%	0.92%	2.954

4. KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan dapat disimpulkan bahwa kedua model YOLOv5m dan YOLOv8m menunjukkan kinerja yang sangat baik dalam mendeteksi penyakit pada daun tanaman tomat. Kedua model menunjukkan nilai presisi, recall, dan mAP yang tinggi. Nilai-nilai ini menunjukkan kemampuan yang sangat baik untuk menemukan penyakit daun tomat dengan akurasi yang tinggi. YOLOv8m lebih unggul dalam presisi dengan nilai 0.95% selisih 0.02% dengan YOLOv5m dan mAP50:95 yaitu 0.92% selisih 0.02% dengan YOLOv5m yang berarti bahwa YOLOv8m lebih baik dalam mengidentifikasi objek dengan sangat tepat dan objek dengan berbagai ukuran, namun YOLOv8m membutuhkan waktu pelatihan sedikit lebih lama dibanding YOLOv5m. YOLOv8m lebih mampu mendeteksi semua objek yang ada dalam gambar dimana nilai presisi lebih unggul 0.02% dibanding YOLOv5m.

REFERENSI

- [1] A. P. Pranjaya, F. Rizki, R. Kurniawan, and N. K. Daulay, “Klasifikasi Penyakit Pada Daun Tanaman Padi Berbasis YoloV5 (You Only Look Once),” *Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 6, pp. 3127–3136, 2024.
- [2] B. Erwanto and D. H. , Afu Ichsan Pradana, “Pengembangan Sistem Deteksi Penyakit Tanaman Tomat Melalui Citra Daun dengan Metode You Only Look Once (YOLO) Berbasis Android,” *G-Tech J. Teknol. Terap.*, vol. 8, no. 3, pp. 1453–1463, 2024.
- [3] A. N. Mendrofa, N. Gea, and K. Gea, “PENGARUH PUPUK ORGANIK AMPAS KELAPA TERHADAP PERTUMBUHAN TANAMAN TOMAT (*Lycopersicum E sculentum* Mill),” *Sapta Agrica*, vol. 2, no. 1, pp. 36–49, 2023.
- [4] BPS, “Latto Lattomat,” 2023.
- [5] G. Henry, A. Panjaitan, and F. Simatupang, “Pemodelan Klasifikasi Penyakit Daun Tanaman Tomat dengan Convolutional Neural Network Algorithm,” *Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 5, pp. 2667–2675, 2024.
- [6] D. Setiawan and T. W. E. Suryawijaya, “ALGORITMA RESNET152V2 DALAM MELAKUKAN KLASIFIKASI PENYAKIT PADA DAUN TOMAT,” *J. Comput. Sci. Technol.*, vol. 3, no. 2, pp. 37–42, 2023.
- [7] A. M. N. Rohma, “Diagnosa Penyakit Tanaman Tomat pada Citra Daun Menggunakan Metode Convolutional Neural Network (CNN),” *JIMU J. Ilm. Multi Disiplin*, vol. 02, no. 03, pp. 555–567, 2024.
- [8] A. W. Putri, “IMPLEMENTASI ARTIFICIAL NEURAL NETWORK (ANN) BACKPROPAGATION UNTUK KLASIFIKASI JENIS PENYAKIT PADA DAUN TANAMAN TOMAT,” *J. Ilm. Mat.*, vol. 09, no. 02, pp. 344–350, 2021.
- [9] J. R. Yasiri, R. Prathivi, and Susanto, “Detection of Plastic Bottle Waste Using YOLO Version 5 Algorithm,” *J. dan Penelit. Tek. Inform.*, vol. 9, no. 1, pp. 20–30, 2025.
- [10] E. Naufal and D. Tanadi, “Sistem Pendeteksi Penyakit Kanker Kulit Menggunakan Convolutional Neural Network Arsitektur YOLOv8 Berbasis Website,” *Neptunus J. Ilmu Komput. dan Teknol. Inf.*, vol. 2, no. 3, 2024.
- [11] J. Jonathan and D. Hermanto, “Penentuan Epochs Hasil Model Terbaik : Studi Kasus Algoritma YOLOv8,” *Digit. Transform. Technol. (Digit.)*, no. September, 2024.

- [12] D. Hafidzulrahman, Perbandingan Algoritma You Only Look Once (YOLO) versi 5 dan versi 8 sebagai Object Detection pada Pendeteksian Hilal. 2024.
- [13] M. I. Mauladany, B. Fatkhurrozi, and R. A. Wibowo, “Deteksi Penyakit Daun Durian dengan Algoritma YOLO (You Only Look Once),” *Aviat. Electron. Inf. Technol. Telecommun. Electr. Control.*, vol. 6, no. 1, p. 73, 2024.
- [14] D. G. Manurung et al., “Deteksi Dan Klasifikasi Hama Potato Beetle Pada Tanaman Kentang Menggunakan YOLOV8,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 4, pp. 723–734, 2024.
- [15] L. S. Riva and Jayanta, “Deteksi Penyakit Tanaman Cabai Menggunakan Algoritma YOLOv5 Dengan Variasi Pembagian Data,” *J. Pengemb. IT(JPIT)*, vol. 8, no. 3, pp. 248–254, 2023.
- [16] R. Hesnananda, I. A. Noviani, and M. Zulfariansyah, “Implementasi YOLOv5 untuk Deteksi Objek Mesin EDC : Evaluasi dan Analisis,” *J. Teknol. Inf. dan Rekayasa Komput.*, vol. 5, no. 2, pp. 104–110, 2024.
- [17] S. Fatimah and T. Ardiansah, “Kombinasi Metode MOORA dan Rank Order Centroid dalam Sistem Pendukung Keputusan Pemilihan Supplier Produk Sepatu,” *Kaji. Ilm. Inform. dan Komput.*, vol. 5, no. 1, pp. 28–38, 2024.
- [18] F. H. Arby, I. Husni, and A. Amin, “Implementation of YOLO-v5 for a real-time Social Distancing Detection,” *J. Appl. Informatics Comput.*, vol. 6, no. 1, pp. 1–6, 2022.
- [19] E. Casas, L. Ramos, E. Bendek, and F. Rivas-Echeverria, “YOLOv5 vs. YOLOv8: Performance Benchmarking in Wildfire and Smoke Detection Scenarios,” *J. Image Graph.*, vol. 12, no. 2, pp. 127–136, 2024.
- [20] I. P. Sary, S. Andromeda, and E. U. Armin, “Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection using Aerial Images,” *Ultim. Comput. J. Sist. Komput.*, vol. 15, no. 1, pp. 8–13, 2023.
- [21] G. V. Agustin, M. Ayub, and S. L. Liliawati, “Deteksi dan Klasifikasi Tingkat Keparahan Jerawat : Perbandingan Metode You Only Look Once,” *J. Tek. Inform. dan Sist. Inf.*, vol. 10, pp. 468–481, 2024