

# Pemanfaatan Yolo Untuk Deteksi Hama Dan Penyakit Pada Daun Cabai Menggunakan Metode *Deep Learning*

Nadini Mardiah Yasen<sup>1</sup>, Silfia Rifka<sup>2\*</sup>, Rikki Vitria<sup>3</sup>, Yulindon<sup>4</sup>

<sup>1,2,3,4</sup> Program Studi Teknik Telekomunikasi, Jurusan Teknik Elektro, Politeknik Negeri Padang

\*Corresponding Author Email: [silfiarifka@gmail.com](mailto:silfiarifka@gmail.com)

**Abstrak**— Tanaman cabai merupakan salah satu tanaman hortikultura di Indonesia yang berpotensi besar dalam ekonomi di Indonesia. Namun, seringkali terjadi gagal panen. Salah satu faktor utama penyebab terjadinya yaitu serangan hama dan penyakit pada tanaman cabai. Hal tersebut dibutuhkan pencegahan dini yang dapat mengurangi kerugian. Berkembangnya teknologi saat ini, pencegahan dapat dilakukan dengan cara mudah dan hemat dengan menggunakan metode deep learning. YOLO adalah algoritma deep learning yang biasa digunakan untuk mendeteksi objek secara *realtime*. Terdapat 4 kelas yang akan diuji yaitu daun yang terdampak penyakit virus kuning, bercak daun, hama thrips, dan daun cabai yang sehat. Pengujian dilakukan dengan aplikasi berbasis web dibuat dengan framework flask. Hasil akurasi dari proses *training* model YOLO dengan epoch 150 yaitu 73%. Nilai *precision*, *recall*, dan *mAP* yang didapatkan yaitu 77,4%, 67,1%, dan 75,1%. Pengujian menghasilkan akurasi di atas 74%. Hasil penelitian ini masih menghasilkan akurasi yang belum cukup tinggi, namun aplikasi dapat digunakan untuk mendeteksi dengan baik dan cukup akurat.

**Kata kunci:** Tanaman cabai, Hama dan penyakit, Deep learning, YOLO

**Abstract**— Chili plants are one of the horticultural crops in Indonesia which have great potential in the Indonesian economy. However, crop failure often occurs. One of the main factors causing this is pest and disease attacks on chili plants. This requires early prevention which can reduce losses. With today's technological developments, prevention can be done easily and economically by using deep learning methods. YOLO is a deep learning algorithm that is commonly used to detect objects in real time. There are 4 classes that will be tested, namely leaves affected by yellow virus disease, leaf spot, thrips pests, and healthy chili leaves. Testing was carried out with a web-based application created with the flask framework. The accuracy results of the YOLO model training process with epoch 150 were 73%. The precision, recall and *mAP* values obtained were 77.4%, 67.1% and 75.1%. Testing produces accuracy above 74%. The results of this research still produce accuracy that is not high enough, but the application can be used to detect it well and is quite accurate.

**Keywords:** Chili plants, Pests and diseases, Deep learning, YOLO

© 2023 Elektron Jurnal Ilmiah

## I. PENDAHULUAN

Indonesia sebagai negara agraris yang mempunyai sektor pertanian yang cukup besar dan memiliki perkebunan yang banyak. Tanaman cabai merupakan tanaman hortikultura yang banyak dibudidayakan. Tanaman cabai merupakan salah satu bahan pangan atau sayuran dan komponen penting yang dibutuhkan oleh manusia dalam membuat makanan, berfungsi sebagai bumbu masakan dan sebagai penambah rasa tersendiri [1]. Selain menjadi tambahan masakan, cabai juga mempunyai nilai ekonomi yang penting. Produksinya menjadi mata pencaharian petani sehingga cabai memiliki nilai pasar yang tinggi. Tanaman cabai sangat membutuhkan pemeliharaan dan perawatan agar mendapatkan hasil panen yang bagus dan melimpah. Berdasarkan data dari Badan Pusat Statistik (BPS), pada tahun 2020 rata-rata produksi cabai nasional mencapai 2,77 ton. Produksi tertinggi tercatat pada bulan Agustus mencapai 280,78 ribu ton dengan luas panen mencapai 73,77 ribu hektar. Ini menyatakan peningkatan dari

tahun sebelumnya, di mana produksi cabai pada tahun 2019 mencapai 183,96 ribu ton atau meningkat sekitar 7,11%.

Seringkali panen cabai mengalami kegagalan seperti busuk, berulat maupun mati disebabkan oleh beberapa hal, salah satu faktor utamanya yaitu serangan penyakit dan hama yang menyerang tanaman tersebut [2]. Hal itu terjadi dapat menyebabkan kerugian yang signifikan bagi para petani. Pengendalian yang tepat dilakukan untuk menghindari kegagalan panen bukan hanya dilakukan ketika serangan pada tanaman telah terjadi, tetapi yang paling penting adalah untuk mencegahnya. Kurangnya kesadaran petani dalam mengetahui jenis penyakit menyebabkan terjadinya keterlambatan diagnosis penyakit pada tanaman [3]. Pencegahan dapat dilakukan dengan mendeteksi penyakit pada tanaman. Saat ini mendeteksi penyakit pada tanaman dilakukan secara manual dengan cara pemantauan oleh ahlinya dengan secara terus menerus. Ini membutuhkan biaya yang sangat tinggi apabila dilakukan pada pertanian ekstensif [4]. Selain itu,

Deteksi secara manual dapat memicu kesalahan dalam identifikasi jenis hama dan penyakit pada tanaman cabai. Ada beberapa petani tanaman cabai yang belum paham mengenai perbedaan jenis penyakit dan hama pada tanaman cabai. Sehingga memberikan obat ataupun tindakan pengendalian yang tidak sesuai dengan jenis penyakitnya.

Ada beberapa teknologi yang digunakan dalam mendeteksi penyakit pada tanaman cabai. Salah satunya yaitu *deep learning*. *Deep learning* digunakan untuk mendeteksi hama dan penyakit pada daun tanaman cabai melalui objek daun dan teknik pengolahan data. Metode ini murah dan mudah untuk mendeteksi sehingga sangat bermanfaat dan membantu para petani. [5], [6].

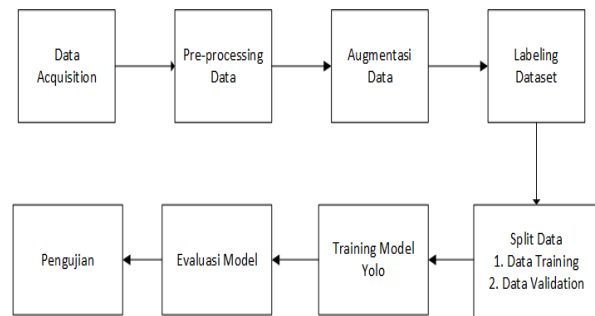
Penelitian yang telah dilakukan sebelumnya untuk mendeteksi Penelitian yang dilakukan oleh Rosalina dan Genta pada tahun 2020 dengan judul “Implementasi *Deep Learning* Dalam Mendeteksi Penyakit Pada Daun Cabai”. Penelitian ini menggunakan model neural network dan Raspberry Pi untuk mengambil gambar daun cabai dengan menambahkan aplikasi desktop untuk sistem pendeteksi. Penelitian ini mendeteksi cabai sehat atau cabai sakit. Hasil akurasi terbaik dari penelitian tersebut diperoleh 100% dengan pencahayaan yang bagus dan jarak antara kamera dengan daun tidak lebih dari 1 meter [4]. Namun, pada penelitian ini data yang dideteksi belum secara *realtime*. Data diinput terlebih dahulu dan selanjutnya akan diproses.

Dari penelitian sebelumnya, akurasi yang diperoleh cukup bagus, namun ada beberapa proses deteksi yang tidak sesuai dengan kelas aslinya dikarenakan tekstur dan warna yang hampir sama. Kemudian, proses dideteksi belum secara *realtime* sehingga dilakukan penginputan data terlebih dahulu.

Berdasarkan hal tersebut penulis mengembangkan penelitian dengan judul “Deteksi Hama Dan Penyakit Pada Daun Tanaman Cabai Menggunakan Metode *Deep Learning*”. Penelitian ini bertujuan untuk membuat aplikasi untuk mendeteksi hama dan penyakit daun tanaman cabai dengan menerapkan metode *deep learning* yang menggunakan algoritma YOLO (*You only Look Once*) yang memungkinkan untuk mendeteksi secara *real-time* sehingga dapat mencegah kegagalan panen dengan mudah. Algoritma YOLO adalah metode *deep learning* yang sering digunakan untuk deteksi objek secara cepat. Sehingga membantu petani tanaman cabai dalam mendeteksi penyakit secara dini secara mudah dan dapat melakukan pengendalian secara cepat dan tepat.

## II. METODE

### A. Rancangan



Gambar 1 Blok Diagram

Perancangan dilakukan dengan membuat model terlebih dahulu dan perangkat lunak (software). Model yang dibuat terdapat data yang diolah terlebih dahulu. Kemudian data tersebut akan dikirimkan pada perangkat lunak. Perangkat lunak nantinya berbentuk aplikasi android yang digunakan oleh user.

#### 2.1 Data Acquisition

Pada penelitian ini pengumpulan data (citra cabai) diambil secara scrapping yaitu *searching* di google, download dataset di website kaagle dengan nama *chili disease*, dan juga diambil secara manual pada kebun cabai menggunakan kamera smartphone dan pencahayaan yang jelas. Data memiliki 4 kelas yang berbeda yaitu cabai virus kuning, thrips, bercak daun, dan cabai yang sehat. Pada gambar 2 menunjukkan contoh gambar dataset perkelas.



Gambar 2. Contoh dataset

Pada umumnya penyakit yang sering menyerang tanaman cabai disebabkan oleh cendawan. Berikut penjelasan mengenai beberapa jenis penyakit yang menyerang cabai melalui fisik daun:

### 2.1.1 Penyakit Virus Kuning

*Vein clearing* yang dialami helai daun dengan tanda awal dengan timbulnya pucuk daun yang berkembang menjadi warna kuning terang, pada daun terjadi penebalan tulang dan menggulung ke atas. Hal ini disebabkan oleh infeksi dari gemini virus yang terus menerus menyebabkan daun mengecil dan berwarna kuning terang, sehingga tanaman menjadi kerdil dan cabai tidak berbuah [7].

### 2.1.2 Penyakit bercak daun

Penyakit menyebabkan kerusakan pada daun, batang dan akar dengan gejala yang ditimbulkan yaitu terlihat pada daun bercakcoklat bulat (ukuran mencapai hingga sekitar 1 inch) dan kering. Pusat bercak berwarna pucat hingga putih dengan warna tepi lebih tua. Lubang pada daun disebabkan oleh bercak yang tua [7].

### 2.1.3 Thrips

Hama jenis thrips ini menyerang tanaman pada bagian permukaan bawah daun. Gejala yang ditimbulkan dari penyakit ini pada daun cabai yaitu berupa bercak keperak-perakkan, mengeriting atau keriput, daun akan bergulung ke dalam dan adanya muncul benjolan. Hama ini sering terjadi pada saat musim kemarau karena pada musim tersebut hama thrips berkembang sangat cepat. Akibat dari hama ini mengakibatkan pertumbuhan pada tanaman cabai menjadi terhambat, kerdil hingga pucuknya menjadi mati [7].

## 2.2 Pre-processing data

Tahap preprocessing bertujuan agar citra dapat di proses lebih mudah untuk proses selanjutnya [8]. Pada tahap ini memilah citra yang sesuai untuk dijadikan dataset dan mengubah ukuran setiap citra menjadi 640 x 640 pixel. Hasil dataset awal sebanyak 460 citra.

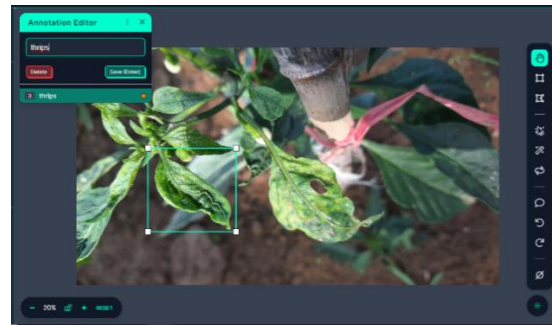
## 2.3 Data Augmentation

Data augmentation merupakan proses yang menambahkan dataset secara kuantitas untuk meningkatkan performa model [9]. Proses menambahkan data ini dengan cara flip dan rotate. Citra akan di flip secara horizontal dan rotate dengan interval 90°. Penambahan data menghasilkan data sebanyak 1226 citra.

## 2.4 Labelling data

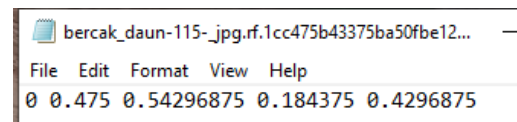
Proses labelling data ini dengan cara memberi anotasi pada citra-citra tersebut dengan menggunakan *bounding box* dan untuk menentukan kelas pada citra.

Proses anotasi dilakukan menggunakan platform roboflow.



Gambar 3. *Bounding box* pada citra

Proses labelling ini akan menghasilkan file txt yang berisi koordinat objek atau *bounding box* dan nama kelas pada objek. Skala pada koordinat dari 0 hingga 1. Kolom pertama merupakan nama kelas pada objek, kolom kedua dan ketiga menunjukkan sumbu x dan y dari kiri pada *bounding box*, kolom keempat menunjukkan lebar, dan kolom kelima menunjukkan tinggi pada *bounding box* [10]. Hasil dari labelling ditunjukkan pada gambar 4.



Gambar 4 Hasil Labelling

## 2.5 Split data

Dalam split data ini membagi dataset menjadi dua bagian, yakni data pelatihan (*training*) dan data validasi (*validation*). Data *training* dibagi menjadi 75% dari keseluruhan dataset dan data *validation* 25% dari keseluruhan dataset. Data *training* dan data *validation* ini nantinya digunakan dalam melatih dan membuat model [9].

## 2.6 Training Model

Pada penelitian ini menggunakan algoritma YOLO (*You Only Look Once*) yang mana merupakan bagian dari algoritma *deep learning* dengan proses pembelajaran yang dilakukan oleh mesin yang menyerupai cara kerja jaringan otak manusia atau yang dikenal dengan neural networks dan dapat digunakan untuk menyelesaikan permasalahan data. baik dalam bentuk analisis pola, *clustering* maupun klasifikasi. Dalam pembelajarannya algoritma ini memiliki sangat banyak jumlah struktur dan jaringan dan rinci serta secara mandiri dapat melakukan pembelajaran yang dapat dijadikan pedoman dalam pemecahan masalah [11].

Algoritma YOLO yang menggunakan *Convolutional Neural Network* (CNN) untuk deteksi objek secara real-time dengan menggunakan pendekatan yang unik dibandingkan algoritma lainnya

karena menerapkan satu jaringan saraf tunggal untuk mengidentifikasi objek di seluruh gambar. Jaringan ini memprediksi *bounding box* mana yang akan berfungsi dalam melakukan prediksi dengan menggunakan fitur dari semua gambar[12]. Metode *deep learning* ini menjadi lebih singkat karena lebih sedikit masalah hilangnya gradien pada back propagation [13]. Versi YOLO yang digunakan yaitu YOLOv5. YOLOv5 merupakan tambahan terbaru dari seri YOLO yang telah ditingkatkan dalam basis dan kecepatan dibandingkan dengan YOLOv4. YOLOv5 memiliki ukuran file yang kecil dan lebih baik dalam deteksi dan akurasi objek [14].

2.7 Evaluasi Model

Confusion matrix digunakan untuk mengevaluasi performa pengujian model terhadap dataset. Parameter performa diantaranya yaitu akurasi, presisi, dan recall[15]. Gambar 5 menunjukkan *confusion matrix*

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 5. Confusion Matrix

Akurasi merupakan persentase yang digunakan untuk membandingkan jumlah data yang akurat dengan keseluruhan data[9]. Persamaan untuk menghitung akurasi ditunjukkan pada persamaan 1.

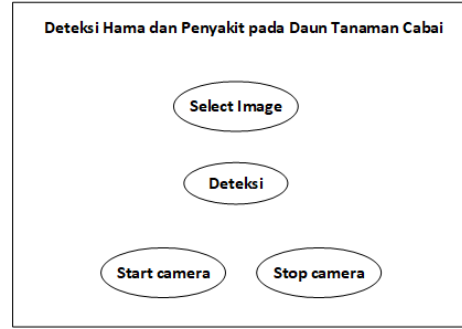
$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \dots\dots\dots (1)$$

2.8 Pengujian

Tahap pengujian ini dilakukan untuk menilai apakah aplikasi yang telah dikembangkan beroperasi dengan efektif dalam mendeteksi dan data terdistribusi dengan tepat. Aplikasi ini dibuat dengan framework flask berbasis web.

B. Pembuatan Aplikasi

Sebelum membuat aplikasi deteksi hama dan penyakit pada daun tanaman cabai, maka dirancang tampilan aplikasi seperti yang terlihat pada gambar 6.



Gambar 6. Rancangan Tampilan Aplikasi

Pembuatan aplikasi pada aplikasi deteksi ini dibuat menggunakan Visual Studio Code melalui terminal. Visual Studio Code sebelumnya di download dan di instal pada laptop. Berikut langkah-langkah membuat aplikasi:

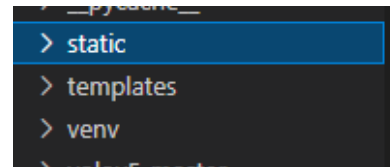
Langkah 1: Menyediakan file model custom dataset yaitu best.pt merupakan hasil dari *training* model YOLO yang telah dilakukan.

Langkah 2 :Mengaktifkan virtual instrument melalui terminal pada visual studio code.

Langkah 3 : Menginstal flask

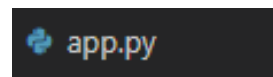
```
# pip install flask
```

Langkah 4 : Menyiapkan direktori static dan template. Static untuk menyimpan style.css dan gambar yang berhubungan dengan desain aplikasi. Sedangkan template untuk menyimpan html. Gambar 7 menunjukkan direktori yang disiapkan.



Gambar 7. Direktori

Langkah 5 : Menyiapkan file dengan format .py seperti ditunjukkan pada gambar 8.



Gambar 8. app.py

Langkah 6 : Membuat program app.py pada visual studio code seperti ditunjukkan gambar 9.



```

Run Terminal Help
* app.py - app - Visual Studio Code
app.py
index.html web.py
app.py
1 from flask import Flask, render_template, request, redirect, Response
2 import cv2
3 import torch
4 import numpy as np
5 import os
6 import io
7 import argparse
8 from PIL import Image
9 import time
10
11 app = Flask(__name__)
12
13
14 # Load YOLOv5 model
15 model = torch.hub.load('ultralytics/yolov5', 'custom', path='best.pt') # Gath menuju model Anda
16 model.eval()
17
18 camera_started = False # Inisialisasi variabel camera_started
19
20 cap = cv2.VideoCapture(0)
21
22 def detect_objects(image):
23     results = model(image) # Deteksi objek menggunakan model YOLOv5
24     output_image = results.render()[0]
    
```

Gambar 9. Program Aplikasi

Langkah 7 : Membuat program html untuk tampilan web seperti ditunjukkan gambar 10.

```

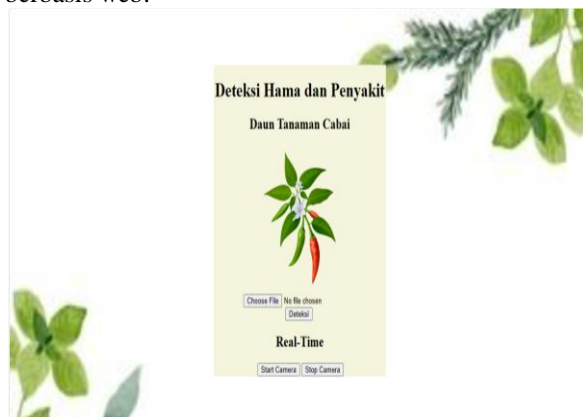
templates > index.html > html > body > div.text-center.centered-title > form.form-signin > div > video/camera-feed-video
app.py index.html x web.py
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>Aplikasi Deteksi Hama dan Penyakit Daun Tanaman Cabai</title>
8 <style>
9
10 body {
11     background-image: url('/static/bgcabai.jpg'); /* path gambar latar */
12     background-size: cover; /* Mengatur tampilan gambar latar */
13     background-repeat: no-repeat; /* Menghindari pengulangan gambar latar */
14     background-attachment: fixed; /* Menjaga gambar latar tetap pada posisi saat scrolling */
15 }
16
17 .container {
18     text-align: center;
19 }
20
21 .box {
22     display: inline-block;
23     width: 38%;
24 }
    
```

Gambar 10. Program Html

Langkah 8 : melakukan run pada terminal.

# python app.py

Berikut gambar 11 dan 12 merupakan tampilan aplikasi deteksi hama dan penyakit tanaman cabai berbasis web.



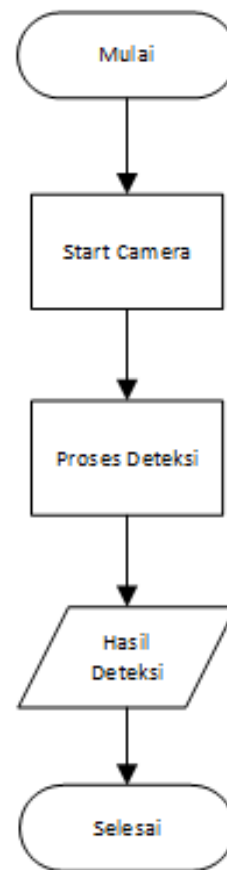
Gambar 11. Tampilan Aplikasi Web



Gambar 12. Tampilan Aplikasi Kamera Hidup

### C. Flowchart

Aplikasi dimulai dengan menginput data, proses, dan hasil data. Gambar 13 merupakan flowchart aplikasi berbasis web secara *realtime*

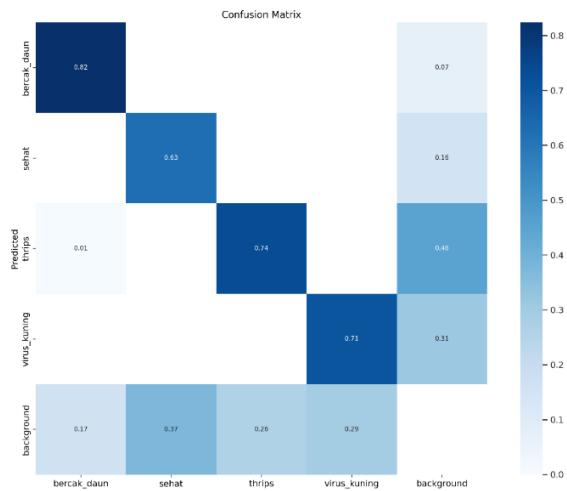


Gambar 13. Flowchart dengan secara *realtime* dan menginput data.

Flowchart aplikasi dengan klik tombol start camera kemudian aplikasi terhubung dengan kamera pada laptop, selanjutnya proses deteksi hama dan penyakit dengan mengarahkan daun tanaman cabai pada kamera hingga hasil kelas objek dan akurasi terdeteksi.

### III. HASIL DAN PEMBAHASAN

Proses *training* model dengan menggunakan 1126 citra yang terdiri dari 4 kelas yaitu virus kuning, hama thrips, bercak daun, dan daun cabai yang sehat. *Training* model menggunakan epoch 150 dan batch 32, menghasilkan confusion matrix terlihat pada gambar 14.



Gambar 14 Confusion Matrix Epoch 150

Untuk menghitung akurasi dapat menggunakan *confusion matrix* dengan menggunakan persamaan 1 sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$

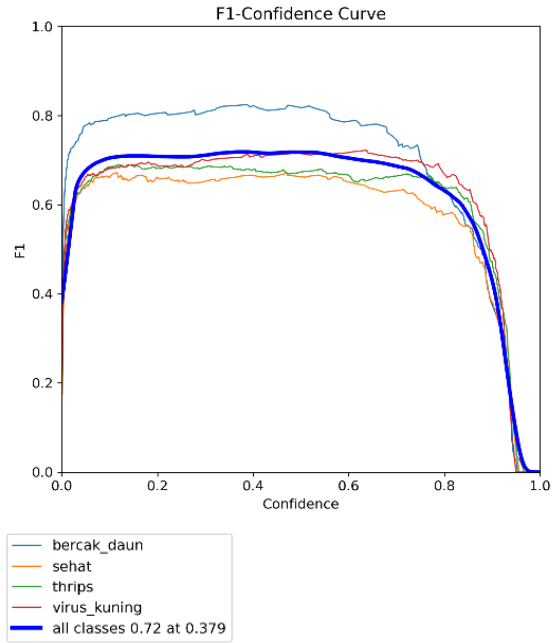
$$Accuracy = \frac{82+63+74+71}{17+37+26+29+82+63+74+71}$$

$$= \frac{290}{399} = 0,726$$

$$= 0,73 \times 100\% = 73\%$$

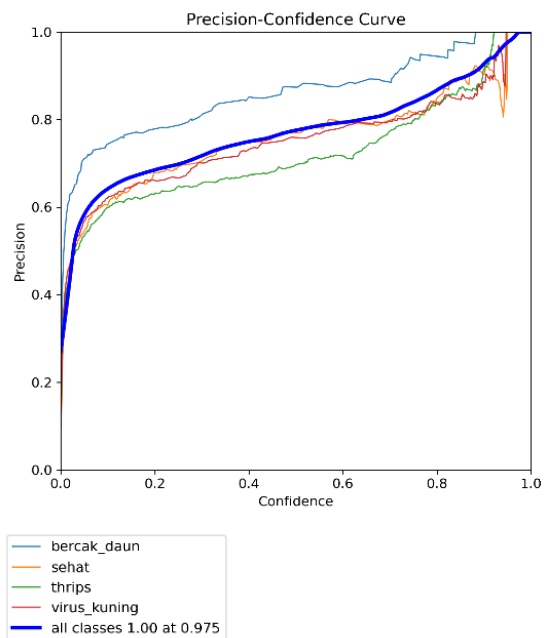
Hasil akurasi yang didapatkan untuk keseluruhan data pada proses *training* model dengan epoch 150 adalah 73%. Nilai akurasi dikatakan baik jika dihasilkan nilai 76%-100%, dikatakan cukup baik jika 55%-75%, dikatakan kurang baik jika dihasilkan nilai 40%-56%, dan tidak baik jika dihasilkan nilai <40% [16]. Ini membuktikan bahwa cukup baik dan akurat dalam mendeteksi hama dan penyakit pada daun tanaman cabai, namun akurasi masih belum terlalu tinggi.

Grafik precision, recall, dan F1 yang dihasilkan dari proses *training* model YOLO dengan epoch 150. Pada gambar 15 merupakan grafik skor F1 terhadap nilai *confidence*. Dari hasil *training* yang didapatkan, skor F1 mencapai nilai rata-rata tertinggi sebesar 0,72 terhadap nilai *confidence* 0,379.



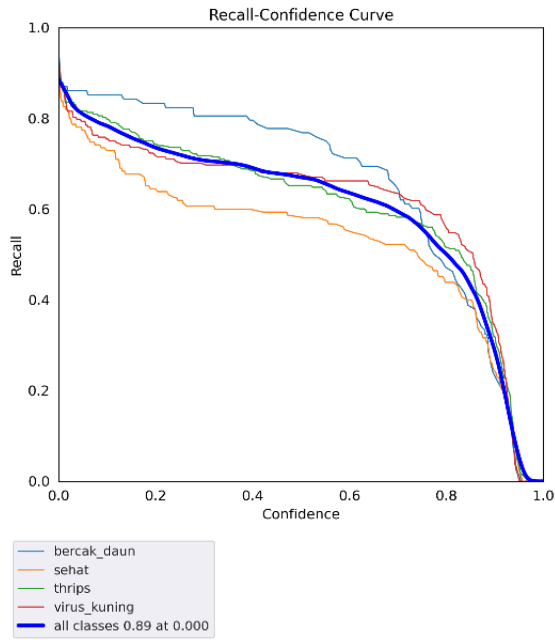
Gambar 15. F1-Confidence Curve

Pada gambar 16 merupakan grafik nilai *Precision* terhadap nilai *confidence*. Dari hasil *training* yang didapatkan, nilai *precision* menghasilkan nilai rata-rata maksimal sebesar 1,00 terhadap nilai *confidence* 0,975



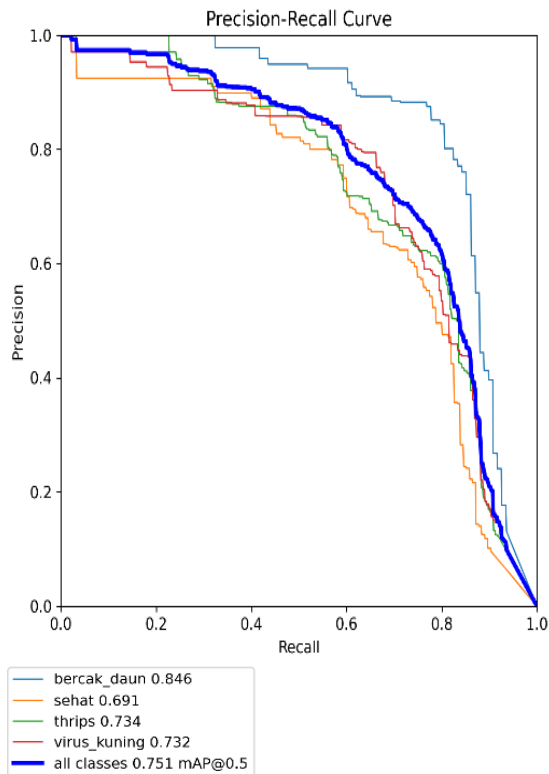
Gambar 16. Precision-Confidence Curve

Pada gambar 17 merupakan grafik nilai *Recall* terhadap nilai *confidence*. Dari hasil *training* yang didapatkan, nilai *precision* menghasilkan nilai rata-rata maksimal sebesar 0,89 terhadap nilai *confidence* 0,00.



Gambar 17. Recall--Confidence Curve

Pada gambar 18 merupakan grafik nilai *Precision-Recall* terhadap nilai *confidence*. Dari hasil *training* yang didapatkan, nilai *precision-recall* menghasilkan nilai rata-rata maksimal sebesar 0,751 terhadap nilai mAP 0,5. Semakin tinggi skor mAP, maka semakin baik performa yang didapatkan.



Gambar 18. Precision-Recall Curve

Hasil *training* dataset dengan batch 32 dan epoch 150 menghasilkan nilai *precision*, *recall*, dan mAP ditunjukkan pada tabel 1.

Tabel 1. Precision, Recall, dan mAP

Kelas	Precision	Recall	mAP
Thrips	69,5%	65,2%	73,4%
Virus kuning	75,9%	67,7%	73,2%
Bercak daun	87,4%	77%	84,6%
Sehat	77%	58,3%	69,1%
Rata-rata	77,4%	67,1%	75,1%

Dari tabel 1 dapat disimpulkan bahwa kelas bercak daun memiliki performa yang lebih bagus dibandingkan dengan kelas-kelas lainnya. Performa yang paling rendah yaitu pada daun yang sehat. Namun, semua kelas dapat mendeteksi dengan baik dan cukup akurat.

Pengujian Deteksi hama dan penyakit daun tanaman cabai secara *realtime* dengan menggunakan aplikasi web yang mana terhubung langsung dengan kamera pada laptop terlihat pada gambar 18. Namun, pengujian secara *realtime* terdapatnya delay sehingga membutuhkan waktu sedikit lebih lama. Nilai rata-rata delay yang dihasilkan yaitu 8m/s. Proses deteksi tersebut didapatkan hasil yang terlihat pada gambar 19.



Gambar 18. Pengujian Aplikasi Secara *Realtime*

Gambar 19. Hasil Deteksi Secara *Realtime*

Tabel 2. Hasil Akurasi Pengujian Aplikasi

Kelas	Akurasi	
	Percobaan 1	Percobaan 2
Sehat	84%	89%
Bercak daun	86%	92%
Thrips	85%	74%
Virus kuning	84%	77%

Percobaan deteksi daun tanaman cabai dengan masing-masing kelas 2 kali percobaan terlihat pada tabel 2 menghasilkan nilai akurasi yang cukup baik. Nilai akurasi dikatakan baik jika dihasilkan nilai 76%-100%, dikatakan cukup baik jika 55%-75%, Ini menunjukkan aplikasi cukup akurat dalam mendeteksi hama dan penyakit sehingga aplikasi dapat digunakan.

Hasil pengujian yang telah dilakukan secara realtime membutuhkan waktu yang sedikit lama karena terdapatnya *delay*. Oleh karena itu, penelitian selanjutnya disarankan menggunakan perangkat dengan spesifikasi yang lebih bagus dan menggunakan webcam sebagai kamera untuk menangkap citra.

#### IV. KESIMPULAN

Aplikasi berbasis web yang dibuat untuk deteksi hama dan penyakit pada daun tanaman cabai dengan menerapkan metode deep learning yaitu algoritma YOLO (*you only look once*) dapat digunakan dengan baik dalam mendeteksi hama dan penyakit pada daun

tanaman cabai. Akurasi dari hasil *training* yaitu 73% menunjukkan bahwa akurasi tersebut dikatakan cukup baik. Sehingga aplikasi dapat digunakan dalam masyarakat.

#### UCAPAN TERIMA KASIH

Terima kasih kepada pihak yang terkait dalam membantu penelitian ini dan paling utama kepada orang tua atas doanya dan pembimbing yang telah membimbing selama penelitian ini sehingga penulis dapat menyelesaikan penelitian ini.

#### REFERENSI

- [1] Adyakbar, "Budidaya Tanaman Cabai," *Pertanian.go.id*, 2019.
- [2] Damaiyanti, R. Yulianty, A. Marzuki, S. Kasim, and H. Rante, "ANALISIS RESIDU PESTISIDA KLORPIRIFOS PADA CABAI ( *Capsicum sp.* ) DARI DESA BUNGIN KECAMATAN BUNGIN," vol. 23, no. 3, pp. 106–108, 2020, doi: 10.20956/mff.v23i3.9401.
- [3] A. A. Muslim and R. Arnie, "Sistem Pakar Diagnosa Hama Dan Penyakit Cabai Berbasis Teorema Bayes," pp. 867–876.
- [4] Rosalina and G. Sahuri, "Implementation of Deep Learning Methods in Detecting Disease on Chili Leaf," vol. 9, no. 6, pp. 10–15, 2020.
- [5] F. Zikra, K. Usman, and R. Patmasari, "Deteksi Penyakit Cabai Berdasarkan Citra Daun Menggunakan Metode Gray Level Co-Occurrence Matrix Dan Support Vector Machine," 2021.
- [6] A. Wijaya, "Pendeteksian Penyakit pada Daun Cabai dengan Menggunakan Metode Deep Learning," vol. 6, pp. 452–461, 2020.
- [7] A. Meilin, *Hama dan Penyakit pada Tanaman Cabai serta Pengendaliannya*. Jambi: Science Innovation Networks, 2014.
- [8] S. S. Zuain, H. Fitriyah, and R. Maulana, "Deteksi Penyakit pada Daun Cabai berdasarkan Fitur HSV dan GLCM," vol. 5, no. 9, pp. 3934–3940, 2021.
- [9] A. Tsany and R. Dzaky, "Deteksi Penyakit Tanaman Cabai Menggunakan Metode Convolutional Neural Network," vol. 8, no. 2, pp. 3039–3055, 2021.
- [10] L. Rahma, H. Syaputra, A. H. Mirza, and S. D. Purnamasari, "Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO ( You Only Look Once )," vol. 2, no. 3, 2021.
- [11] S. MR, "4 Metode Deep Learning yang Digunakan dalam Data Science," *DQLab*, 2022. <https://dqlab.id/4-metode-deep-learning-yang-digunakan-dalam-data-science#:~:text=Deep Learning adalah suatu proses,pola atau clustering maupun klasifikasi.>
- [12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection".



- [13] A. A. Hania, "Mengenal Artificial Intelligence, Machine Learning, & Deep Learning," *J. Teknol. Indones.*, vol. 1, no. June, pp. 1–6, 2017, [Online]. Available: <https://amt-it.com/mengenal-perbedaan-artificial-intelligence-machine-learning-deep-learning/>
- [14] M. H. Ashar and D. Suarna, "Implementasi Algoritma YOLOv5 dalam Mendeteksi Penggunaan Masker Pada Kantor Biro Umum Gubernur Sulawesi Barat," vol. 3, no. 3, pp. 298–302, 2022.
- [15] F. Rofii, G. Priyandoko, M. I. Fanani, and A. Suraji, "Peningkatan Akurasi Penghitungan Jumlah Kendaraan dengan Membangkitkan Urutan Identitas Deteksi Berbasis YOLOv4 Deep Neural Networks," vol. 42, no. 2, pp. 169–177, 2021, doi: 10.14710/teknik.v42i2.37019.
- [16] P. K. Laut, "Implementasi certainty factor dalam mengatasi ketidakpastian pada sistem pakar diagnosa penyakit kuda laut," vol. VII, no. 1, 2020.