

Eggplant Leaf Diseases Detection and Counting System on Agricultural Robot Based on YOLOv8

Yohanes Gilang Prasaja Putra^{a1}, Ardy Seto Priambodo^{a2}

^aProgram Studi Teknik Elektronika, Fakultas Vokasi, Universitas Negeri Yogyakarta
Kulon Progo, Daerah Istimewa Yogyakarta, Indonesia
¹yohanesgilangpp@gmail.com
²ardyseto@uny.ac.id

Abstract

Eggplants play an important role in Indonesia's economic and food sectors. Production increases yearly, but diseases such as earworm, flea beetle, leaf spot, and leafhopper cause significant losses. Additionally, manual disease detection methods are time-consuming and prone to errors. This research develops an automated system using YOLOv8 and the Telegram Bot system to detect and count eggplant leaf diseases. The YOLOv8 algorithm can detect objects in videos or images quickly and accurately in real-time. The model uses the OpenVINO format for faster inference compared to PyTorch. The research results show that the YOLOv8 model achieved a mAP50 of 0.606. This system effectively detects, tracks, and counts eggplant leaf diseases and can send PDF files via email through the Telegram Bot. Accuracy evaluation shows an accuracy rate of 98% from the confusion matrix, with video testing showing the highest accuracy variation at 97.05% and the lowest at 23.07%. These results indicate that the automated eggplant leaf disease detection system has great potential to help agriculture prevent eggplant diseases more efficiently and accurately, supporting increased production and quality of harvests.

Keywords: Disease Detection, Eggplant Leaf Diseases, Sprayer Robot, Telegram Bot, YOLOv8

1. Pendahuluan

Tanaman hortikultura banyak ditanam oleh petani di Indonesia karena memberikan kontribusi besar pada ekonomi dan kesediaan pangan masyarakat. Salah satu tanaman hortikultura tersebut adalah terung yang secara sistematis masuk dalam anggota *Solanaceae*, berkerabat dengan cabai, tomat, dan kentang [1]. Produktivitas tanaman terung di Indonesia meningkat setiap tahunnya. Menurut data Badan Pusat Statistik (BPS) tahun 2022, produksi terung naik dari 676.339 ton pada tahun 2021 menjadi 691.738 ton pada tahun 2022, meningkat 2,27%. Namun, produksi ini belum memenuhi permintaan pasar karena tingginya minat masyarakat mengonsumsi terung dan harganya yang ekonomis [2].

Terung sering diserang penyakit yang menyebabkan rusaknya tanaman dan kerugian ekonomi. Beberapa penyakit muncul akibat serangan *earworm*, *flea beetle*, *leaf spot*, dan *leafhopper*, yang menyebabkan kerugian panen besar. Umumnya penyakit tanaman dapat dideteksi dari daun [3]. Oleh karena itu, pengamatan daun terung secara tepat efektif untuk memastikan kesehatan tanaman [4]. Metode deteksi penyakit manual yang dilakukan petani saat ini memakan waktu dan rawan kesalahan [5]. Oleh karena itu, diperlukan teknologi otomatis untuk mendeteksi penyakit pada terung. Sistem deteksi otomatis ini dapat memberikan informasi cepat dan tepat untuk pencegahan dan pengendalian penyakit [6].

Smart farming 4.0 dengan *Artificial Intelligence* (AI) berkembang pesat di era digital, dirancang untuk meningkatkan produktivitas petani dengan membangun budidaya pertanian yang terintegrasi, terukur, dan efisien [7]. Penelitian ini mengusulkan deteksi dan penghitungan penyakit tanaman terung berdasarkan daun menggunakan algoritma *You Only Look Once* versi 8 (YOLOv8) dan monitoring berbasis Telegram Bot. Algoritma YOLO memiliki performa terbaik dibandingkan metode lain seperti Faster-RCNN dan SSD karena resolusi tinggi dalam mendeteksi objek kecil [8]. YOLO dapat mendeteksi gambar dan memberikan *bounding box* untuk menentukan letak penyakit pada daun

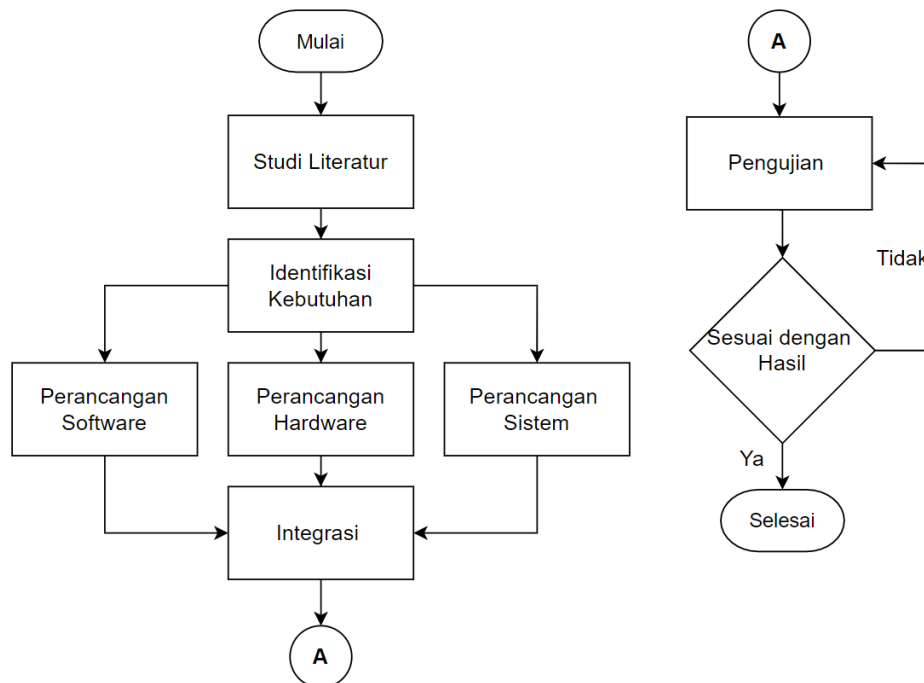
dengan lebih akurat [9]. Sistem deteksi ini dirancang dalam bentuk robot beroda yang dapat dijalankan di lahan pertanian. Penelitian ini diharapkan dapat membantu petani dalam melakukan analisis penanganan yang tepat serta meningkatkan produktivitas pertanian terung.

2. Tinjauan Literatur

Dalam beberapa tahun terakhir, penelitian mengenai deteksi penyakit tanaman telah berkembang pesat dengan memanfaatkan teknologi deep learning. Penelitian oleh Keke Zhang dkk. (2021) [10] menggunakan algoritma Faster R-CNN untuk mendeteksi penyakit pada daun soybean, meskipun sistem deteksi ini tidak bekerja secara real-time dan hanya menggunakan data tes gambar. Sementara itu, Huishan Li dkk. (2023) [11] berhasil meningkatkan deteksi penyakit secara real-time pada daun apel dengan menggunakan algoritma BTC-YOLOv5 dan dataset sebanyak 2099 gambar. Selain itu, Md Janibul Alam Soeb dkk. (2023) [12] menggunakan algoritma YOLOv7 dan dataset berjumlah 4000 gambar untuk mendeteksi lima jenis penyakit pada daun teh, menunjukkan potensi yang signifikan dalam deteksi penyakit tanaman. Sebagai perbandingan, Ruchi Gajjar dkk. (2021) [13] menggunakan algoritma CNN untuk mendeteksi penyakit pada daun tanaman apel, jagung, kentang, dan tomat secara real-time dengan jumlah dataset yang sangat besar yaitu 21.978 gambar. Pada penelitian lainnya, Muhammad Juman Jhatial dkk. (2022) [14] menggunakan algoritma YOLOv5 untuk mendeteksi empat jenis penyakit pada daun padi, namun sistem ini juga tidak bekerja secara real-time dan menggunakan dataset terdiri dari 400 gambar penyakit daun padi dari Kaggle.

3. Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini ditunjukkan pada gambar 1.



Gambar 1. Flowchart Metode Penelitian

Berdasarkan *flowchart* tersebut dapat diuraikan sebagai berikut:

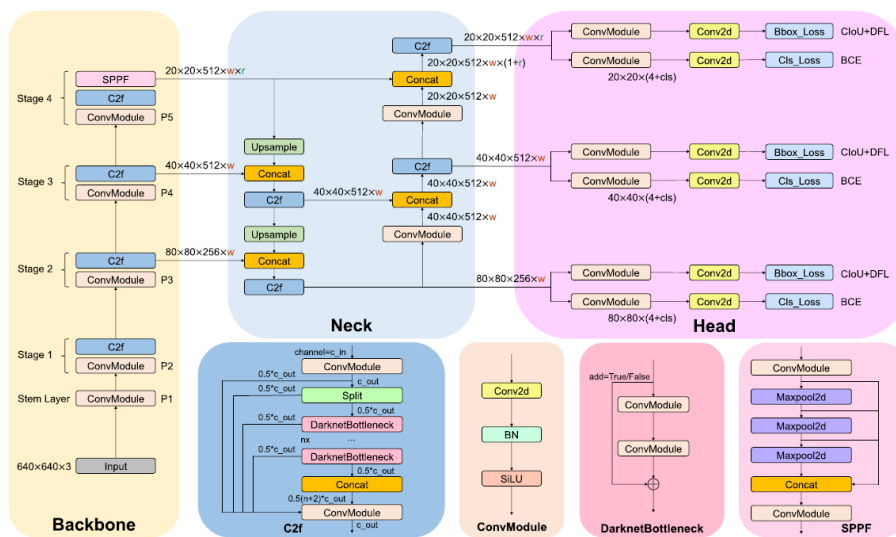
- Studi literatur: proses ini mencakup membaca, menganalisis, dan mencatat bahan penelitian yang berkaitan dengan penelitian yang dilakukan. Bahan yang didapatkan digunakan untuk menyusun penelitian agar relevan dengan penelitian yang sudah pernah dilakukan. Studi literatur yang digunakan berasal dari jurnal, skripsi, buku, maupun *website*.
- Identifikasi kebutuhan: proses ini bertujuan menentukan komponen yang digunakan dalam melakukan penelitian, sehingga pelaksanaan penelitian akan berjalan sesuai rencana. Identifikasi kebutuhan dilakukan dengan analisis dokumen dan konsultasi.
- Perancangan *software*, *hardware*, dan sistem: bertujuan untuk membuat gambaran proyek yang akan dibuat. Rancangan yang dibuat akan menjadi pedoman dalam melakukan penelitian. Pada bagian ini dilengkapi dengan flowchart, diagram blok, dan tabel penjelasan.

- d. Integrasi: mengintegrasikan masing-masing rancangan yaitu *software* (pengiriman data), *hardware* (elektronik dan mekanik), dan sistem menjadi satu kesatuan yang saling berhubungan.
- e. Pengujian: pengujian dilakukan untuk mengetahui hasil dan fungsi dari proyek yang telah dibuat apakah sesuai dengan yang diinginkan atau belum. Pengujian dilakukan dengan pengujian masing-masing komponen, pengujian sistem, dan pengujian integrasi.
- f. Hasil akhir: hasil akhir merupakan proyek yang telah dilakukan pengujian sistem. Evaluasi pada bagian ini dilakukan dengan analisis hasil dan kinerja integrasi alat. Jika belum sesuai, maka akan dilakukan pengujian kembali hingga mendapatkan hasil yang sesuai keinginan.

3.1 Identifikasi Kebutuhan

3.1.1 You Only Look Once (YOLO) V8

YOLO merupakan model deteksi objek *real time* yang digunakan untuk mendeteksi objek dalam bentuk video atau gambar. Algoritma ini dirancang untuk mengambil gambar masukan, mengolah fitur-fitur gambar melalui sistem prediksi, kemudian menggambar *bounding box* di sekitar objek-objek yang terdeteksi, sambil memberikan prediksi kelas objek tersebut [14].

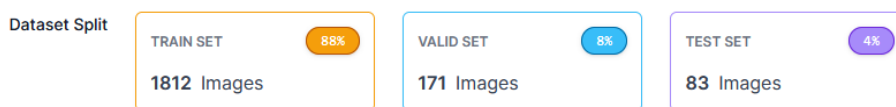


Gambar 2. Arsitektur YOLOv8

Gambar 2 merupakan arsitektur YOLOv8 yang mengimplementasikan dua neural network, yaitu Feature Pyramid Network (FPN) dan Path Aggregation Network (PAN). Adanya gabungan fitur-fitur ini dirancang untuk mempermudah proses anotasi gambar dalam *training* model, menjadikan prosesnya lebih efisien dan mudah dilakukan [15].

3.1.2 Dataset

Dataset penelitian ini diproses dalam *website* Roboflow dengan 4 kelas penyakit yaitu *earworm*, *flea beetle's damage*, *leaf spot*, dan *leafhopper*. Gambar dataset dilakukan proses *preprocessing* seperti mengubah dimensi gambar 640x640, format RGB, dan dibagi menjadi 3 bagian yaitu *training*, *validation*, dan *test*. Gambar 3 menunjukkan pembagian dataset.



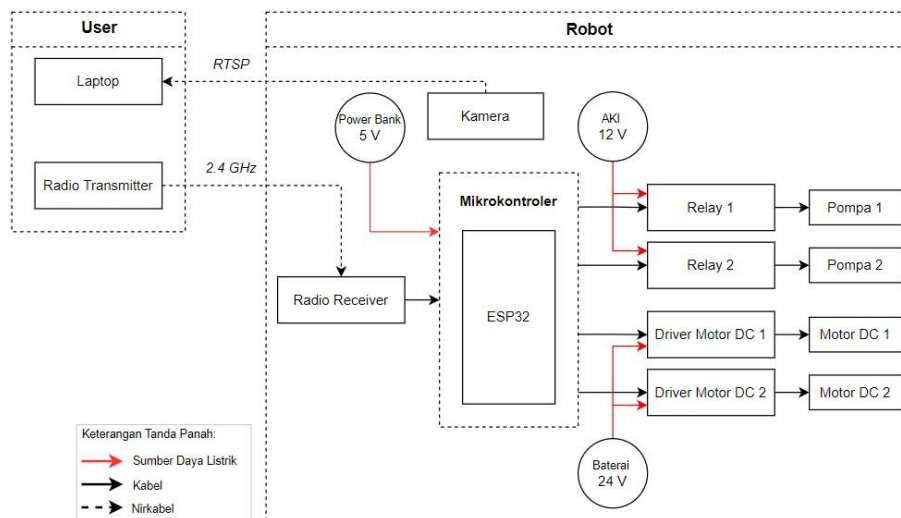
Gambar 3. Pembagian Dataset

Setelah dibagi menjadi 3 bagian, dataset *training* di lakukan proses *augmentation* untuk menambah variasi dataset sehingga model menerima lebih banyak contoh *training*. *Augmentation* yang dilakukan yaitu *flip*, *90° rotate*, *crop*, *rotation*, *brightness*, dan *blur*.

3.2 Perancangan Alat

Perancangan alat pada penelitian yang dilakukan terdiri dari hardware (elektronik dan mekanik), software (pengiriman data), dan sistem secara keseluruhan.

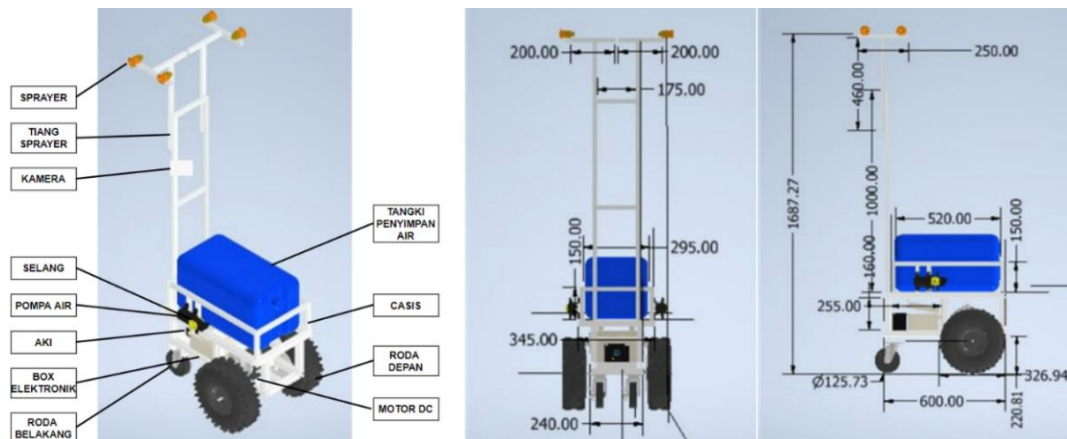
3.2.1 Rancangan Elektronik



Gambar 4. Blok Diagram Elektronik

Gambar 4 menunjukkan komponen elektronik terbagi menjadi dua bagian yaitu komponen yang dipegang dan dikendalikan oleh pengguna, seperti laptop dan radio transmitter, serta komponen yang terpasang dalam robot. Mikrokontroler ESP32 berperan sebagai pemroses sinyal dan data yang juga mengendalikan outputnya. Sumber daya listrik untuk ESP32 dan komponen lain yang membutuhkan tegangan 5V dipasok oleh power bank, sementara motor DC sebagai penggerak robot menggunakan driver BTS7960 dengan tegangan 24V dari baterai LiFePo4. Sistem semprot robot menggunakan relay dan pompa 12V, serta sistem deteksi menggunakan laptop sebagai pemroses deteksi objek dan kamera smartphone untuk menangkap video dengan *Real Time Streaming Protocol* (RTSP).

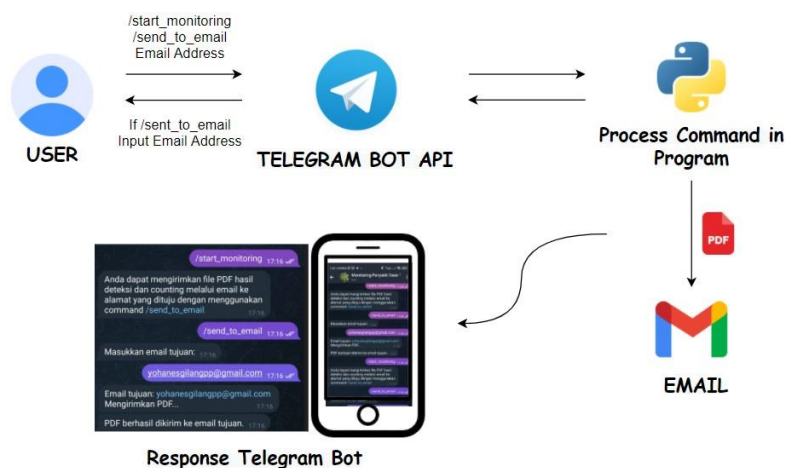
3.2.2 Rancangan Mekanik



Gambar 5. Desain Mekanik Robot

Pada desain gambar 5, robot memiliki ukuran panjang 600 mm, lebar 500 mm, dan tinggi tiang sprayer yang dapat diatur antara 750 mm hingga 1687,27 mm dengan lebar 175 mm. Peletakan setiap komponen diatur agar mudah diakses, termasuk kotak elektronik yang tahan air dan jeriken air atau pestisida berkapasitas 30 liter yang dapat diturunkan untuk diisi ulang. Sistem gerak robot memungkinkan belokan di tempat tanpa sistem kemudi, dengan roda bergerak berlawanan arah untuk belok kanan atau kiri. Selain itu, smartphone android ditempatkan di salah satu sisi robot dan peletakan kabel juga memperhatikan aspek keamanan K3 untuk memastikan keselamatan pengguna.

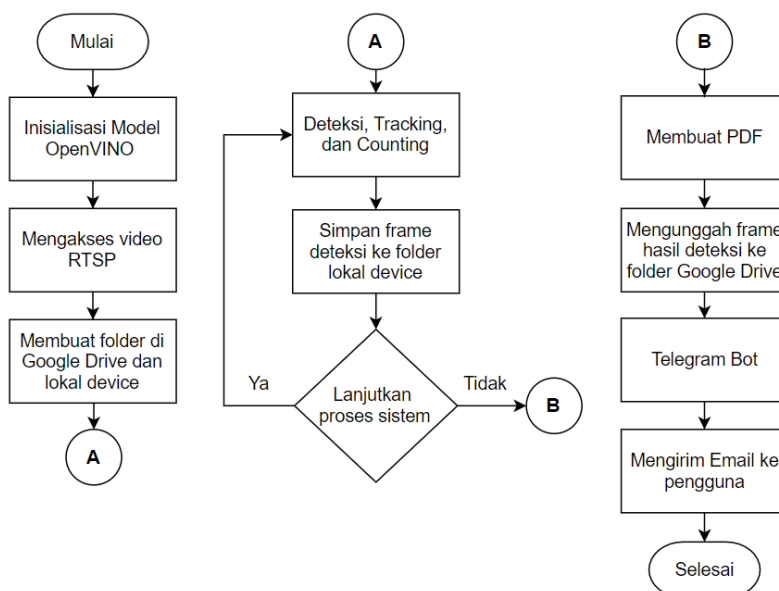
3.2.3 Rancangan Pengiriman Data



Gambar 6. Alur Pengiriman Data Menggunakan Telegram Bot

Pengiriman data ke email diproses menggunakan Telegram Bot yang bernama "Monitoring Penyakit Daun Terung". Data yang dikirim berupa file PDF yang berisi informasi hasil deteksi dan penghitungan penyakit pada daun tanaman terung. Pengguna dapat masuk ke Telegram Bot dan memasukkan perintah `/start_monitoring` untuk menampilkan cara mengirimkan data ke email. Selanjutnya, dengan menggunakan perintah `/send_to_email`, pengguna diminta memasukkan alamat email tujuan. Jika berhasil, file PDF akan terkirim ke email yang telah dimasukkan, dan pengguna akan menerima notifikasi bahwa PDF berhasil dikirim ke email. Seluruh proses ini menggunakan Telegram Bot API dan diimplementasikan melalui program Python seperti pada gambar 6.

3.2.4 Rancangan Sistem



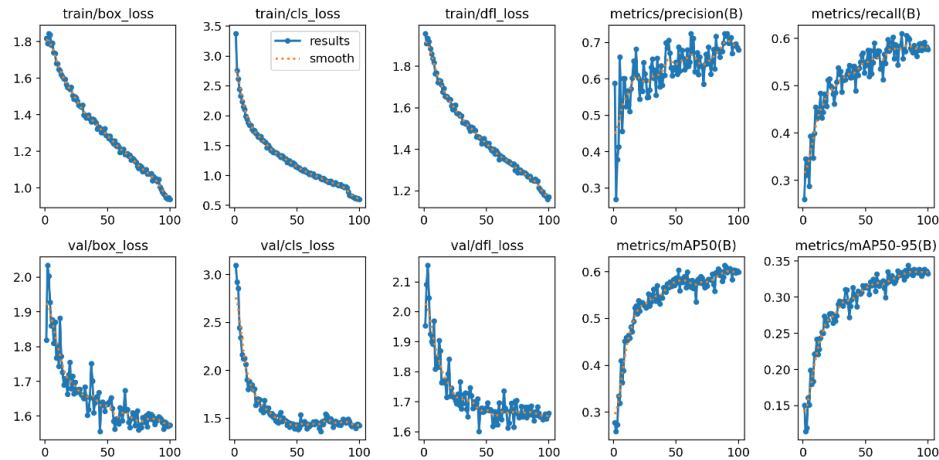
Gambar 7. Flowchart Keseluruhan Sistem

Proses program ditunjukkan pada gambar 7, dimulai dengan mengimpor *library* dan modul yang diperlukan serta memuat model deteksi OpenVINO dari penyimpanan lokal. Video *live streaming* dari smartphone Android diakses menggunakan IP Webcam dan protokol RTSP. Proses kemudian membuat folder di penyimpanan lokal dan Google Drive untuk menyimpan frame yang terdeteksi penyakit. Selanjutnya proses deteksi menggunakan YOLOv8, pelacakan objek dengan *Simple Online*

and Realtime Tracking (SORT), serta penghitungan penyakit pada daun tanaman terung. Frame yang terdeteksi penyakit disimpan di folder lokal. Setelah program deteksi berhenti, file PDF dibuat dan disimpan di folder lokal, kemudian frame hasil deteksi diunggah ke Google Drive. Pengguna dapat menggunakan Telegram Bot untuk mengirim file PDF ke email yang ditentukan.

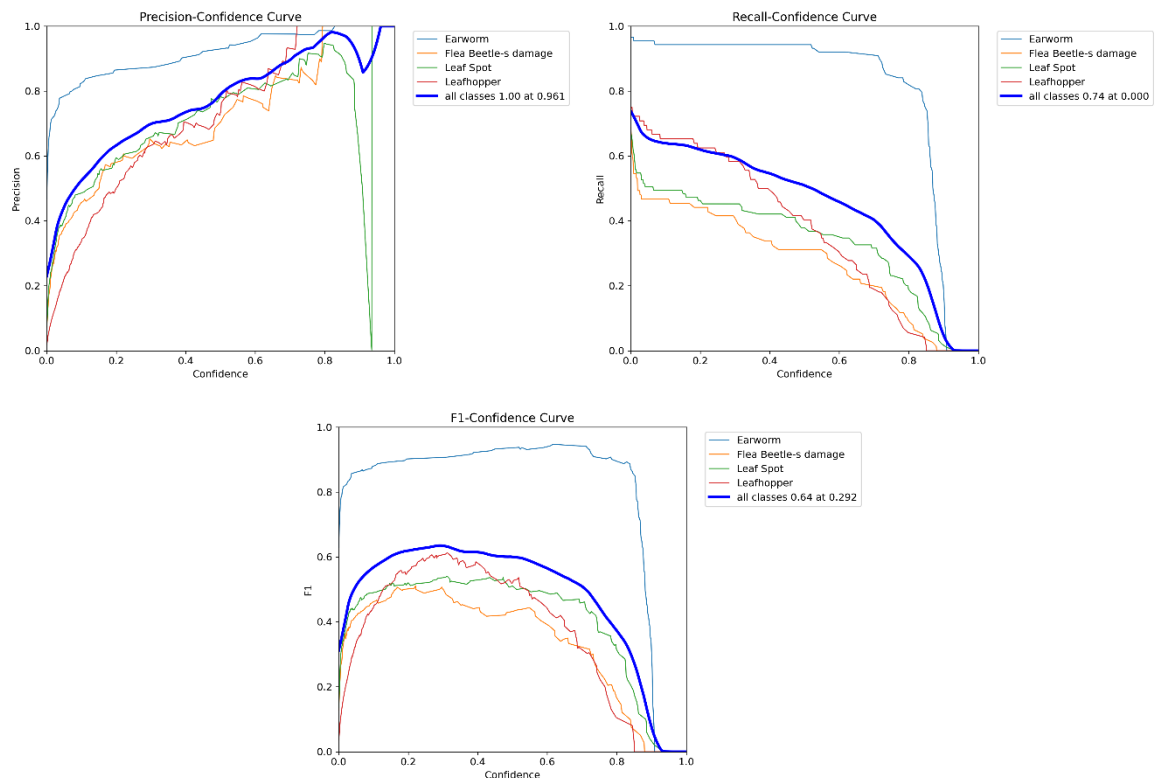
4. Hasil dan Pembahasan

4.1. Training Dataset



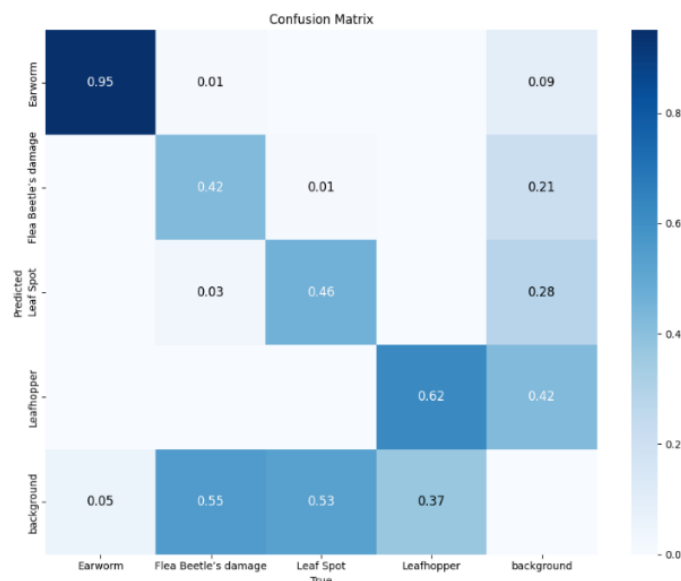
Gambar 8. Hasil Training Dataset

Training dataset dilakukan dengan menggunakan Google Colaboratory karena dapat mempercepat proses dengan GPU. Kategori YOLOv8 yang digunakan yaitu YOLOv8n karena paling ringan dan tercepat dibanding kategori lainnya. Format model yang dihasilkan Pytorch (.pt). Terdapat beberapa parameter lain seperti 100 epoch, ukuran *image* 640, waktu *training* 1,133 jam. Gambar 8 merupakan grafik hasil keseluruhan proses *training* yang memuat informasi data *loss* (*training* dan *validation*), mAP50, mAP50-95, *precision*, dan *recall*. Grafik *loss* menunjukkan adanya penurunan setiap epoch sehingga model yang dihasilkan semakin baik dalam melakukan prediksi. mAP50 model sebesar 0,606 dan mAP50-95 model sebesar 0,344 untuk keseluruhan kelas penyakit.



Gambar 9. Precision, Recall, dan F1 Score

Precision menunjukkan jika model memiliki *precision* 1.00 pada *threshold* 0.961 untuk semua kelas. *Recall* yang dihasilkan adalah 0.74 di *threshold* 0.00 yang menunjukkan jika model dapat mendeteksi 74% dari semua data positif yang dideteksi dengan benar. *F1 score* mendapatkan hasil 0,64 di *threshold* 0,292. Hal tersebut menunjukkan jika pada *threshold* 0,292, model memiliki kinerja baik dalam melakukan prediksi objek. Hasil grafik tersebut ditampilkan pada gambar 9.



Gambar 10. Confusion Matrix

Akurasi model dihitung berdasarkan *confusion matrix* pada gambar 10 dengan menggunakan persamaan berikut.

$$\begin{aligned}
 \text{Akurasi} &= \frac{TP+TN}{TP+TN+FN+TN} \\
 &= \frac{0,95+0,42+0,46+0,62}{0,95+0,42+0,46+0,62+0,03+0,01+0,01} \\
 &= \frac{2,45}{2,5} \\
 &= 0,98
 \end{aligned}$$

Pada penelitian ini untuk mendeteksi penyakit daun terung memperoleh akurasi model sebesar 0,98 atau 98% menunjukkan model dapat mengklasifikasi objek dengan akurasi yang baik.

4.2. Pengiriman Data Telegram Bot

Data file PDF yang dikirimkan ke email melalui Telegram Bot terdapat *time delay* saat pengiriman. Selain itu, juga mengetahui apakah PDF berhasil terkirim atau tidak. Jika terkirim, dalam Telegram Bot akan ada pesan “PDF berhasil dikirim ke email tujuan”.

Tabel 1. Hasil Pengujian *Time Delay* Pengiriman Data

No	<i>Time Delay</i> PDF Terkirim	Status
1	11,94 detik	Terkirim
2	12,06 detik	Terkirim
3	12,77 detik	Terkirim
4	11,47 detik	Terkirim
5	11,70 detik	Terkirim

Hasil pengujian pada table 1 menunjukkan bahwa pengiriman file PDF melalui Telegram Bot memiliki rata-rata *time delay* 11,99 detik. Perbedaan *time delay* dipengaruhi kekuatan sinyal smartphone. Setiap pengiriman berhasil tanpa kegagalan, menunjukkan keandalan dan konsistensi sistem.

4.3. Deteksi YOLOv8 Menggunakan Video

Pengujian dilakukan untuk menghitung penyakit daun tanaman terung dengan merekam video dari sisi kanan dan kiri pada tiga kelompok yang terdiri dari 3, 5, dan 10 tanaman. Data yang diambil dari rekaman video tersebut kemudian dianalisis untuk mendeteksi jumlah penyakit pada daun. Hasil deteksi (prediksi) dibandingkan dengan total penyakit aktual yang ada pada tanaman untuk menghitung tingkat akurasi sistem.

Tabel 2. Hasil Pengujian Menggunakan 3 Tanaman Terung

No	Video			Jumlah		Akurasi
	Sisi	Waktu	FPS	Aktual	Prediksi	
1	kanan	1 menit 11 detik	10,72	Earworm: 0 Flea Beetle's damage: 20 Leaf Spot: 3 Leafhopper: 0	Earworm: 0 Flea Beetle's damage: 22 Leaf Spot: 3 Leafhopper: 1	86,9%
2	kanan	51 detik	12,8	Earworm: 0 Flea Beetle's damage: 11 Leaf Spot: 2 Leafhopper: 0	Earworm: 0 Flea Beetle's damage: 14 Leaf Spot: 0 Leafhopper: 0	92,3%
3	kiri	59 detik	8,55	Earworm: 0 Flea Beetle's damage: 9 Leaf Spot: 0 Leafhopper: 0	Earworm: 0 Flea Beetle's damage: 10 Leaf Spot: 0 Leafhopper: 2	66,6%
4	kiri	1 menit 5 detik	9,04	Earworm: 0 Flea Beetle's damage: 11 Leaf Spot: 2 Leafhopper: 0	Earworm: 0 Flea Beetle's damage: 22 Leaf Spot: 0 Leafhopper: 1	23,07%

Berdasarkan hasil tabel 2 akurasi deteksi tertinggi terdapat pada percobaan ke-2 pengambilan video dari sisi kanan dengan akurasi 92,3% dan waktu pemrosesan video 51 detik, selain itu juga mendapatkan FPS tertinggi 12,8 FPS. Sedangkan terendah ada pada percobaan ke-4 dengan pengambilan video dari sisi kiri mendapat akurasi 23,07 dan 9,04 FPS.

Tabel 3. Hasil Pengujian Menggunakan 5 Tanaman Terung

No	Video			Jumlah		Akurasi
	Sisi	Waktu	FPS	Aktual	Prediksi	
1	kanan	1 menit 30 detik	12,16	Earworm: 0 Flea Beetle's damage: 17 Leaf Spot: 5 Leafhopper: 0	Earworm: 0 Flea Beetle's damage: 32 Leaf Spot: 5 Leafhopper: 0	31,8%
2	kanan	1 menit 34 detik	11,3	Earworm: 0 Flea Beetle's damage: 23 Leaf Spot: 5 Leafhopper: 0	Earworm: 0 Flea Beetle's damage: 32 Leaf Spot: 0 Leafhopper: 0	85,7%
3	kiri	1 menit 40 detik	8,81	Earworm: 0 Flea Beetle's damage: 15 Leaf Spot: 5 Leafhopper: 0	Earworm: 0 Flea Beetle's damage: 22 Leaf Spot: 8 Leafhopper: 5	75%
4	kiri	1 menit 52 detik	8,93	Earworm: 0 Flea Beetle's damage: 17 Leaf Spot: 5 Leafhopper: 0	Earworm: 0 Flea Beetle's damage: 28 Leaf Spot: 2 Leafhopper: 3	50%

Berdasarkan hasil tabel 3 akurasi deteksi tertinggi terdapat pada percobaan ke-2 dengan akurasi 85,7% dan waktu 1 menit 34 detik. FPS akhir dari proses deteksi yaitu 11,3 FPS. Sedangkan akurasi terendah pengujian ke-1 mendapatkan 31,8%, namun mendapat FPS tertinggi yaitu 12,16.

Tabel 4. Hasil Pengujian Menggunakan 10 Tanaman Terung

No	Video			Jumlah		Akurasi
	Sisi	Waktu	FPS	Aktual	Prediksi	
1	kanan	2 menit 3 detik	11,68	Earworm: 0 Flea Beetle's damage: 37 Leaf Spot: 5 Leafhopper: 0	Earworm: 0 Flea Beetle's damage: 37 Leaf Spot: 2 Leafhopper: 1	95,2%
2	kanan	2 menit 50 detik	8,72	Earworm: 0 Flea Beetle's damage: 33 Leaf Spot: 6 Leafhopper: 0	Earworm: 1 Flea Beetle's damage: 42 Leaf Spot: 6 Leafhopper: 1	71,7%
3	kiri	3 menit 29 detik	8,7	Earworm: 0 Flea Beetle's damage: 32 Leaf Spot: 8 Leafhopper: 0	Earworm: 1 Flea Beetle's damage: 38 Leaf Spot: 1 Leafhopper: 4	90%
4	kiri	3 menit 6 detik	8,58	Earworm: 0 Flea Beetle's damage: 27 Leaf Spot: 7 Leafhopper: 0	Earworm: 1 Flea Beetle's damage: 31 Leaf Spot: 0 Leafhopper: 1	97,05%

Berdasarkan hasil tabel 4 akurasi deteksi tertinggi terdapat pada percobaan ke-4 dengan akurasi 97,05% dan waktu 3 menit 6 detik serta mendapat 8,58 FPS. Sedangkan akurasi terendah pada pengujian ke-2 dengan nilai 71,7% dan waktu 2 menit 50 detik serta mendapat 8,72 FPS.

Dari tiga pengujian YOLOv8 menggunakan video, akurasi deteksi bervariasi dari 23,07% hingga 97,05%, hal tersebut dipengaruhi oleh kondisi pencahayaan, pergerakan daun, dan fokus serta stabilitas kamera. Kondisi pencahayaan yang kurang optimal, pergerakan daun akibat angin, dan fokus serta stabilitas kamera sangat mempengaruhi kualitas gambar dan akurasi deteksi. FPS selama pengujian juga bervariasi, dengan tertinggi 12,8 dan terendah 8,55. Pada awal pengujian FPS cenderung meningkat dan setelah mencapai kondisi maksimal mengalami kondisi naik-turun atau tidak stabil. Optimasi pencahayaan, meminimalkan pergerakan daun, dan memastikan kamera yang stabil serta fokus yang tepat dapat meningkatkan akurasi dan stabilitas FPS. Meskipun ada tantangan, model YOLOv8 tetap mampu mendeteksi gejala penyakit dengan baik.

Contoh hasil frame yang terdeteksi penyakit daun tanaman terung ditunjukkan pada gambar 11.



Gambar 11. Frame Terdeteksi Penyakit

5. Kesimpulan

Sistem deteksi menggunakan YOLOv8 dapat melakukan deteksi dan penghitungan empat jenis penyakit daun tanaman terong dengan video dan secara *real time*. Selain itu, pengiriman data dalam bentuk file PDF melalui Telegram Bot dapat dilakukan dengan lancar. *Training* model YOLOv8, diperoleh mAP50 sebesar 0,606 dengan 100 epoch dan waktu training selama 1,133 jam. Akurasi model yang diukur menggunakan *confusion matrix*, mencapai 98%. Saat mendeteksi penyakit pada video, akurasi tertinggi yang dicapai adalah 97,05%, sedangkan akurasi terendah sebesar 23,07%. Hasil tersebut dapat ditingkatkan dengan menambah variasi dataset, misalnya melalui teknik augmentasi. Selain itu, memaksimalkan penggunaan GPU selama proses *inference* akan meningkatkan FPS dan penggunaan gimbal pada smartphone yang dipasang di robot akan mengurangi getaran video selama proses deteksi, sehingga hasil deteksi lebih stabil dan akurat.

Daftar Pustaka

- [1] A. Al Fandi, R. Muchtar, and N. Notarianto, "Pengaruh media tanam terhadap pertumbuhan tanaman terong (*Solanum melongena* L.) dengan sistem hidroponik," *Jurnal Ilmiah Respati*, vol. 11, no. 2, pp. 114-127, 2020.
- [2] Badan Pusat Statistik, "Produksi Tanaman Sayuran," 2023. [Online]. Available: <https://www.bps.go.id/id/statistics-table/2/NjEjMg==/produksi-tanaman-sayuran.html>. [27 November 2023].
- [3] M. G. Jayanthi, D. R. Shashikumar, and S. Preethi "Eggplant leaf disease detection and segmentation using adaptively regularized multi Kernel-Based Fuzzy C-Means and Optimal PNN classifier," *Indian Journal of Computer Science and Engineering*, vol. 13, no. 5, pp. 1543-1558, 2022.
- [4] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks," *IEEE Access*, vol. 7, pp. 59069–59080, 2019.
- [5] M. Shoaib, B. Shah, S. El-Sappagh, A. Ali, A. Ullah, F. Alenezi, T. Gechev, T. Hussain, and F. Ali, "An advanced deep learning models-based plant disease detection: A review of recent research," *Frontiers in Plant Science*, vol. 14, p. 1158933, 2023.
- [6] M. R. Haque and F. Sohel, "Deep network with score level fusion and inference-based transfer learning to recognize leaf blight and fruit rot diseases of eggplant," *Agriculture*, vol. 12, no. 8, p. 1160, 2022.
- [7] Rachmawati, R.R. "Smart Farming 4.0 untuk Mewujudkan Pertanian Indonesia Maju, Mandiri, dan Modern," *Forum Penelitian Agro Ekonomi*, vol. 38, no. 2, hal. 137-154, 2020.
- [8] J.-a. Kim, J.-Y. Sung, and S.-h. Park, "Comparison of Faster R-CNN, YOLO, and SSD for Real-Time Vehicle Type Recognition," in *2020 IEEE International Conference on Consumer Electronics-Asia*, pp. 1–4, 2020.
- [9] P. K. Sethy, N. K. Barpanda, A. K. Rath, dan S. K. Behera, "Deep feature based rice leaf disease identification using support vector machine," *Computers and Electronics in Agriculture*, vol. 175, pp. 105527, 2020.
- [10] K. Zhang, Q. Wu, and Y. Chen, "Detecting soybean leaf disease from synthetic image using multi-feature fusion faster R-CNN," *Computers and Electronics in Agriculture*, vol. 183, p. 106064, 2021
- [11] H. Li, L. Shi, S. Fang, and F. Yin, "Real-time detection of apple leaf diseases in natural scenes based on YOLOv5," *Agriculture*, vol. 13, no. 4, p. 878, 2023.
- [12] M. J. A. Soeb, M. F. Jubayer, T. A. Tarin, M. R. Al Mamun, F. M. Ruhad, A. Parven, ... and I. M. Meftaul, "Tea leaf disease detection and identification based on YOLOv7 (YOLO-T)," *Scientific Reports*, vol. 13, no. 1, p. 6078, 2023.
- [13] R. Gajjar, N. Gajjar, V. J. Thakor, N. P. Patel, and S. Ruparelia, "Real-time detection and identification of plant leaf diseases using convolutional neural networks on an embedded platform," *The Visual Computer*, pp. 1-16, 2022.
- [14] M. J. Jhatial, R. A. Shaikh, N. A. Shaikh, S. Rajper, R. H. Arain, G. H. Chandio, A. Q. Bhangwar, H. Shaikh, and K. H. Shaikh, "Deep learning-based rice leaf diseases detection using yolov5," *Sukkur IBA Journal of Computing and Mathematical Sciences*, vol. 6, no. 1, pp. 49-61, 2022.
- [15] I. P. Sary, S. Andromeda, and E. U. Armin, "Performance comparison of yolov5 and yolov8 architectures in human detection using aerial images," *Ultima Computing: Jurnal Sistem Komputer*, vol. 15, no. 1, pp. 8-13, 2023.