

DETEKSI KONDISI DAUN TANAMAN JAGUNG MENGUNAKAN MODEL YOLOV9M BERBASIS ANDROID

Murni¹, Rendra Soekarta², Suhardi Aras³, Bakti Prasetyo⁴

^{1,2,3,4}Program Studi Teknik Informatika, Universitas Muhammadiyah Sorong

e-mail: murni44@um-sorong.ac.id¹, rsoekarta@um-sorong.ac.id², suhardi.aras@um-sorong.ac.id³, baktiprasetyo@gmail.com⁴

Abstrak

Tanaman jagung merupakan salah satu komoditas pertanian penting yang berperan dalam memenuhi kebutuhan pangan dan mendukung perekonomian masyarakat. Salah satu kendala dalam meningkatkan produktivitas jagung adalah serangan penyakit pada daun, seperti hawar daun dan karat daun, yang dapat menurunkan kualitas dan hasil panen. Proses identifikasi kondisi daun yang masih dilakukan secara manual melalui pengamatan visual membutuhkan waktu dan berpotensi menghasilkan identifikasi yang kurang akurat, terutama bagi pengguna yang belum berpengalaman dalam mengenali gejala penyakit. Penelitian ini bertujuan untuk menerapkan model YOLOv9m dalam membangun sistem deteksi kondisi daun tanaman jagung secara real-time berbasis Android serta mengetahui kinerja model dalam mengidentifikasi tiga kelas, yaitu daun sehat, hawar daun, dan karat daun. Dataset yang digunakan berjumlah 3.000 citra yang terdiri atas tiga kelas tersebut. Pengembangan sistem menggunakan metode Extreme Programming (XP), sedangkan pengujian dilakukan menggunakan Black Box Testing dan Usability Testing. Model YOLOv9m dilatih dan kemudian dikonversi ke format TensorFlow Lite untuk diintegrasikan ke dalam aplikasi Android menggunakan Android Studio. Hasil penelitian menunjukkan bahwa model YOLOv9m mampu mendeteksi dan mengidentifikasi kondisi daun tanaman jagung dengan tingkat akurasi sebesar 94,4%. Hasil Black Box Testing menunjukkan bahwa seluruh fungsi aplikasi berjalan sesuai dengan kebutuhan, sedangkan Usability Testing memperoleh persentase sebesar 85,46% yang termasuk dalam kategori sangat setuju. Dengan demikian, aplikasi yang dikembangkan dinilai layak digunakan sebagai alat bantu untuk mendeteksi kondisi daun tanaman jagung secara real-time melalui perangkat Android.

Kata Kunci: Tanaman Jagung, Deteksi Daun, YOLOv9m, Deep learning, Android.

1. PENDAHULUAN

Jagung merupakan salah satu komoditas pertanian yang memiliki peranan penting dalam memenuhi kebutuhan pangan, pakan ternak, dan bahan baku industri [1]. Di Papua Barat Daya, jagung juga menjadi salah satu komoditas yang berpotensi mendukung ketahanan pangan dan perekonomian masyarakat. Berdasarkan data Badan Pusat Statistik, produksi jagung pipilan kering dengan kadar air 14 persen di Papua Barat Daya pada tahun 2025 mencapai 2,73 ribu ton, meningkat sebesar 130,10 persen dibandingkan tahun 2024 yang mencapai 1,19 ribu ton [2]. Peningkatan produksi tersebut menunjukkan bahwa komoditas jagung memiliki potensi untuk terus dikembangkan sehingga berbagai faktor yang dapat memengaruhi produktivitas tanaman perlu mendapat perhatian.

Seiring dengan pengembangan produksi jagung, berbagai faktor yang dapat memengaruhi kesehatan dan produktivitas tanaman juga perlu mendapat perhatian, salah satunya adalah serangan penyakit pada daun [3]. Beberapa penyakit yang dapat menyerang tanaman jagung adalah hawar daun dan karat daun. Hawar daun ditandai dengan munculnya bercak memanjang berwarna kecokelatan pada permukaan daun, sedangkan karat daun ditandai dengan munculnya pustula berwarna oranye hingga kecokelatan [4], [5]. Serangan penyakit tersebut dapat mengganggu proses fotosintesis dan memengaruhi pertumbuhan tanaman [6]. Proses identifikasi penyakit yang dilakukan secara manual melalui pengamatan visual membutuhkan pengetahuan dan pengalaman dalam mengenali gejala [7]. Kondisi tersebut dapat menyebabkan kesalahan identifikasi, terutama ketika gejala yang muncul memiliki kemiripan. Kebutuhan akan proses

identifikasi yang cepat dan tepat mendorong pemanfaatan teknologi untuk membantu mengenali kondisi daun tanaman jagung [8].

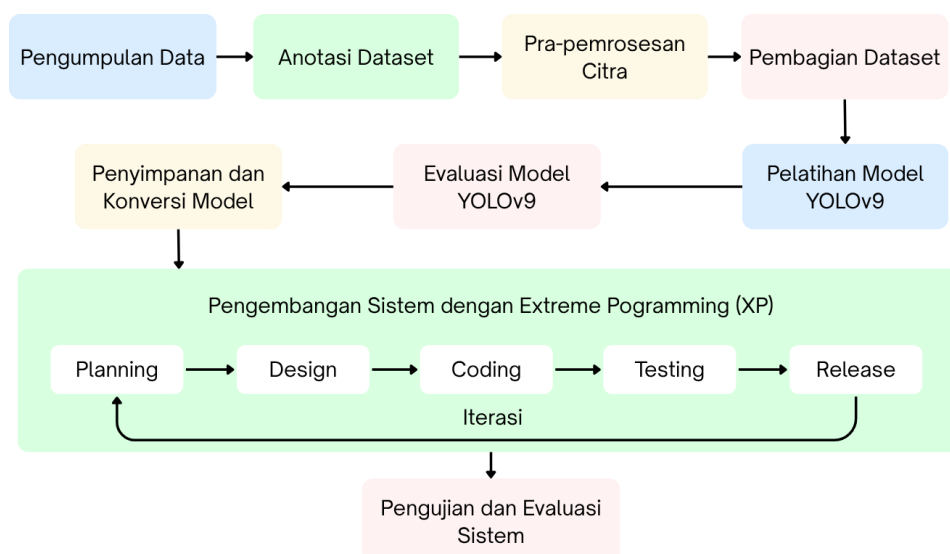
Perkembangan teknologi *computer vision* dan *deep learning* memberikan peluang untuk membantu proses identifikasi penyakit tanaman berdasarkan citra [9]. Salah satu algoritma yang dapat digunakan untuk melakukan deteksi objek adalah YOLO (*You Onl Look Once*) [10]. Algoritma YOLO mampu melakukan deteksi objek secara cepat dalam satu proses. Berbagai penelitian sebelumnya telah menerapkan model berbasis YOLO untuk mendeteksi penyakit pada tanaman berdasarkan citra, termasuk penyakit pada daun tanaman jagung [11]. Penelitian [12] menggunakan model YOLOv8n untuk mendeteksi penyakit pada daun tanaman jagung, sedangkan penelitian [13], [14] menerapkan model YOLO untuk mendeteksi penyakit tanaman berdasarkan karakteristik visual pada citra. Hasil penelitian tersebut menunjukkan bahwa model berbasis YOLO memiliki potensi untuk membantu proses deteksi penyakit tanaman secara otomatis.

Berdasarkan permasalahan tersebut, penelitian ini menerapkan model YOLOv9m untuk mendeteksi kondisi daun tanaman jagung yang terdiri atas tiga kelas, yaitu daun sehat, hawar daun, dan karat daun. Model yang telah dilatih kemudian diintegrasikan ke dalam aplikasi berbasis Android agar proses deteksi dapat dilakukan secara langsung melalui perangkat seluler. Aplikasi juga dilengkapi dengan informasi mengenai karakteristik setiap kondisi daun untuk membantu pengguna memahami hasil deteksi. Penelitian ini mengevaluasi kinerja model YOLOv9m dalam mendeteksi ketiga kelas tersebut serta menguji fungsi dan tingkat penerimaan aplikasi menggunakan *Black Box Testing* dan *Usability Testing*. Penelitian ini diharapkan dapat menghasilkan alat bantu yang mampu melakukan deteksi kondisi daun tanaman jagung secara cepat dan dapat digunakan dengan mudah melalui perangkat Android.

2. METODE PENELITIAN

2.1 Alur Penelitian

Alur penelitian terdiri atas beberapa tahapan utama, yaitu pengumpulan data, anotasi *dataset*, pra-pemrosesan citra, pembagian *dataset*, pelatihan model YOLOv9m, evaluasi model YOLOv9m, penyimpanan dan konversi model, pengembangan sistem menggunakan metode *Extreme Programming* (XP), serta pengujian dan evaluasi sistem. Tahapan tersebut dilakukan secara berurutan untuk menghasilkan model yang mampu mendeteksi daun sehat, hawar daun, dan karat daun pada tanaman jagung serta mengintegrasikannya ke dalam aplikasi berbasis Android. Alur penelitian yang dilakukan ditunjukkan pada Gambar 1.



Gambar 1. Alur Penelitian

2.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data primer yang diperoleh melalui pengumpulan citra daun tanaman jagung. *Dataset* berjumlah 3.000 citra yang terdiri atas tiga kelas, yaitu daun sehat, hawar daun, dan karat daun.

2.3 Anotasi Dataset

Anotasi *dataset* dilakukan menggunakan *platform* Roboflow dengan memberikan *bounding box* dan label kelas pada setiap objek daun yang akan dideteksi [15]. Setiap citra diberi label berdasarkan tiga kelas, yaitu daun sehat, hawar daun, dan karat daun. Hasil anotasi kemudian digunakan sebagai *dataset* untuk proses pelatihan model YOLOv9m.

2.4 Pra-pemrosesan Dataset

Dataset diproses menggunakan Roboflow dengan mengubah ukuran citra (*resize*) menjadi 512×512 piksel serta menerapkan augmentasi berupa *horizontal flip* dan *vertical flip* [16]. *Dataset* kemudian dibuat dengan tiga varian augmentasi sehingga menghasilkan 7.163 citra.

2.5 Pembagian Dataset

Dataset hasil pra-pemrosesan dan augmentasi dibagi menggunakan Roboflow menjadi data *training*, *validation*, dan *testing*. Pembagian *dataset* dilakukan untuk mempersiapkan data dalam proses pelatihan, validasi, dan pengujian model YOLOv9m.

2.6 Model YOLOv9m

Model YOLOv9m (*You Only Look Once* version 9) merupakan algoritma object detection berbasis *deep learning* yang dapat mendeteksi dan mengklasifikasikan objek dalam satu proses [17]. Pada penelitian ini, YOLOv9m digunakan untuk mendeteksi kondisi daun tanaman jagung yang terdiri atas tiga kelas, yaitu daun sehat, hawar daun, dan karat daun. Model dilatih menggunakan *dataset training* yang telah melalui proses anotasi, pra-pemrosesan, augmentasi, dan pembagian *dataset*.

2.7 Evaluasi Model

Kinerja model YOLOv9m dievaluasi menggunakan *precision*, *recall*, *F1-score*, *score Precision* (mAP) [18], dan *confusion matrix*. Evaluasi dilakukan untuk mengetahui kemampuan model dalam mendeteksi dan mengklasifikasikan kondisi daun tanaman jagung pada tiga kelas, yaitu daun sehat, hawar daun, dan karat daun. Nilai *precision* untuk setiap kelas dihitung berdasarkan jumlah *True Positive* (TP), *False Positive* (FP), dan *False Negative* (FN) pada kelas tersebut menggunakan Persamaan (1).

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \quad (1)$$

Nilai *recall* untuk setiap kelas dihitung menggunakan Persamaan (2).

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \quad (2)$$

F1-score untuk setiap kelas dihitung berdasarkan keseimbangan antara *precision* dan *recall* menggunakan Persamaan (3).

$$F1_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (3)$$

Selanjutnya, *Average Precision* (AP) dihitung untuk setiap kelas berdasarkan kurva *precision-recall*. Nilai AP untuk kelas ke-*i* dapat dinyatakan menggunakan Persamaan (4).

$$AP_i = \int_0^1 P_i(R) dR \quad (4)$$

Di mana $P_i(R)$ merepresentasikan *precision* pada kelas ke-*i* berdasarkan nilai *recall* *R*, sedangkan *dR* menunjukkan perubahan kecil pada nilai *recall*. Karena penelitian menggunakan tiga kelas, nilai mAP (*mean Average Precision*) diperoleh dengan menghitung rata-rata AP dari ketiga kelas menggunakan Persamaan (5).

$$mAP = \frac{1}{3} (AP_{Sehat} + AP_{Hawar} + AP_{Karat}) \quad (5)$$

2.8 Save dan Konversi Model

Model terbaik hasil pelatihan disimpan dalam format .pt, kemudian dikonversi ke format TensorFlow Lite (.tflite) [19]. Model hasil konversi selanjutnya diintegrasikan ke dalam aplikasi Android menggunakan Android Studio untuk mendukung proses deteksi kondisi daun tanaman jagung secara *real-time*.

2.7 Pengembangan Sistem

Pengembangan aplikasi Android dilakukan menggunakan metode *Extreme Programming* (XP) [20]. Metode ini digunakan sebagai pendekatan dalam proses pengembangan perangkat lunak yang meliputi tahapan *Planning*, *Design*, *Coding*, *Testing*, dan *Release*.

2.7.1 Planning

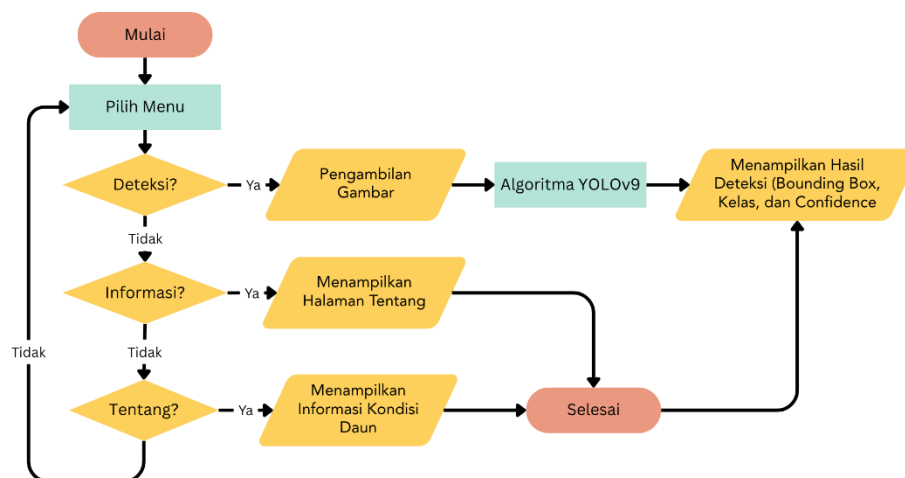
Tahap *Planning* dilakukan untuk mengidentifikasi kebutuhan perangkat keras dan perangkat lunak yang digunakan dalam pengembangan aplikasi. Perangkat keras yang digunakan berupa laptop, sedangkan perangkat lunak yang digunakan meliputi Android Studio, Google Colaboratory, dan Figma.

2.7.2 Design

Tahap *Design* dilakukan untuk merancang alur kerja dan tampilan aplikasi. Perancangan sistem divisualisasikan menggunakan *Flowchart* dan *Use Case Diagram*, sedangkan rancangan antarmuka aplikasi dibuat menggunakan Figma.

a. Flowchart Sistem

Flowchart digunakan untuk menggambarkan alur proses dan interaksi pengguna dengan aplikasi dalam mendeteksi kondisi daun tanaman jagung. *Flowchart* sistem ditunjukkan pada Gambar 2.

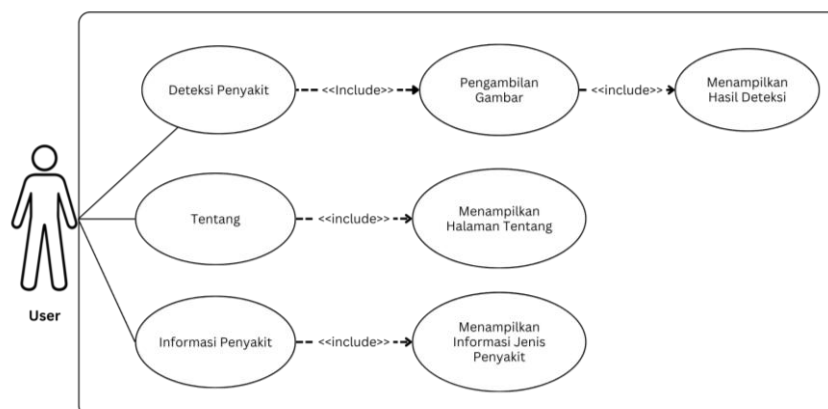


Gambar 2. Flowchart Sistem

Flowchart digunakan untuk menggambarkan alur proses dan interaksi pengguna dalam aplikasi deteksi kondisi daun tanaman jagung. Gambar 2 menunjukkan alur mulai dari pemilihan menu hingga proses deteksi dan penyajian informasi kepada pengguna.

b. UseCase Sistem

Use Case Diagram digunakan untuk menggambarkan interaksi pengguna dengan fungsi-fungsi utama yang tersedia dalam aplikasi. *Use Case Diagram* sistem ditunjukkan pada Gambar 3.



Gambar 3. Usecase Sistem

Use Case Diagram digunakan untuk menggambarkan interaksi pengguna dengan fungsi utama yang tersedia dalam aplikasi. Gambar 3 menunjukkan tiga fungsi utama, yaitu Deteksi Kondisi Daun, Tentang, dan Informasi Kondisi Daun.

2.7.3 Coding

Tahap *Coding* dilakukan untuk mengimplementasikan rancangan sistem ke dalam aplikasi Android. Pengembangan aplikasi dilakukan menggunakan Android Studio, sedangkan Google Colaboratory digunakan untuk proses pelatihan dan evaluasi model YOLOv9m. Model YOLOv9m yang telah dikonversi ke format TensorFlow Lite kemudian diintegrasikan ke dalam aplikasi untuk mendeteksi kondisi daun tanaman jagung secara *real-time*.

2.7.4 Testing

Tahap *Testing* dilakukan untuk memastikan aplikasi berjalan sesuai dengan kebutuhan yang telah ditentukan. Pengujian dilakukan menggunakan *Black Box Testing* untuk menguji fungsionalitas aplikasi dan *Usability Testing* untuk mengetahui tingkat kemudahan serta penerimaan pengguna terhadap aplikasi.

2.7.5 Release

Tahap *Release* merupakan tahap akhir dalam pengembangan sistem menggunakan metode XP. Pada tahap ini, aplikasi yang telah melalui proses pengembangan dan pengujian disiapkan untuk digunakan oleh pengguna.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengumpulan Dataset

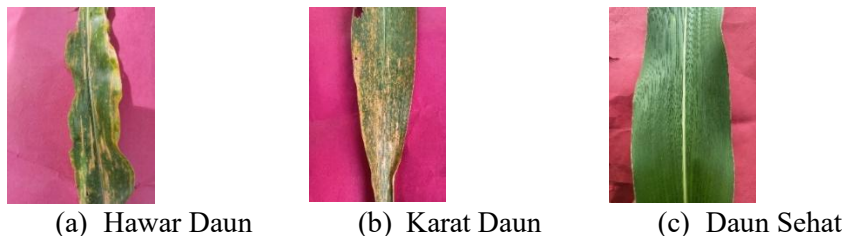
Dataset yang digunakan dalam penelitian ini merupakan data primer berupa citra daun tanaman jagung yang dikumpulkan melalui observasi. *Dataset* terdiri atas tiga kelas, yaitu hawar daun, karat daun, dan daun sehat, dengan masing-masing kelas berjumlah 1.000 citra. Rincian jumlah *dataset* ditunjukkan pada Tabel 1. *Dataset* kemudian divalidasi oleh Bapak Sutardi selaku Kepala Laboratorium Pengamatan Hama dan Penyakit Tanaman Kabupaten Sorong.

Tabel 1. Jumlah *Dataset*

No	Kelas	Jumlah
1.	Hawar Daun	1.000 citra
2.	Karat Daun	1.000 citra
3	Daun Sehat	1.000 citra
Jumlah		3.000 citra

3.2 Hasil Anotasi Dataset

Anotasi *dataset* dilakukan menggunakan Roboflow dengan memberikan *bounding box* dan label pada setiap objek daun sesuai dengan kelasnya. Hasil anotasi terdiri atas tiga kelas, yaitu daun sehat, hawar daun, dan karat daun. Contoh hasil anotasi *dataset* ditunjukkan pada Gambar 4.



Gambar 4. Hasil Anotasi *Dataset*: (a) Hawar Daun, (b) Karat Daun, dan (c) Daun Sehat.

Berdasarkan Gambar 4, citra yang digunakan terdiri atas tiga kelas dengan karakteristik visual yang berbeda. Daun sehat memiliki kondisi daun normal tanpa gejala penyakit, sedangkan hawar daun dan karat daun menunjukkan karakteristik kerusakan atau perubahan warna pada permukaan daun.

3.3 Hasil Pra-pemrosesan dan Augmentasi

Setelah proses pra-pemrosesan dan augmentasi menggunakan Roboflow, jumlah *dataset* meningkat dari 3.000 menjadi 7.163 citra. Perbandingan jumlah *dataset* sebelum dan setelah proses tersebut ditunjukkan pada Tabel 2.

Tabel 2. Hasil Pra-pemrosesan dan Augmentasi

Tahapan	Jumlah Citra
<i>Dataset Awal</i>	3.000
<i>Dataset Hasil Augmentasi</i>	7.163

Berdasarkan Tabel 2, jumlah citra meningkat dari 3.000 menjadi 7.163 citra setelah proses pra-pemrosesan dan augmentasi. Peningkatan tersebut menghasilkan variasi citra yang lebih beragam untuk digunakan dalam proses pelatihan model YOLOv9m.

3.4 Pembagian Dataset

Dataset hasil pra-pemrosesan dan augmentasi kemudian dibagi menjadi data *training*, *validation*, dan *testing* menggunakan Roboflow. Pembagian *dataset* dilakukan untuk mempersiapkan data dalam proses pelatihan dan evaluasi model YOLOv9m. Rincian pembagian *dataset* ditunjukkan pada Tabel 3.

Tabel 3. Hasil Pembagian *Dataset*

Jenis Data	Jumlah Citra	Persentase
<i>Training</i>	6263	87,44%
<i>Validation</i>	600	8,38%
<i>Testing</i>	300	4,19%

Berdasarkan Tabel 3, sebagian besar *dataset* digunakan sebagai data *training*, sedangkan data *validation* dan *testing* digunakan untuk proses validasi dan pengujian model YOLOv9m.

3.5 Hasil Pelatihan Model YOLOv9m

Pada tahap pelatihan, *dataset* sebanyak 7.163 citra digunakan untuk melatih model YOLOv9m dengan *epoch* sebanyak 50 dan ukuran citra masukan (imgsz) sebesar 640×640

piksel. Proses pelatihan dilakukan menggunakan *framework* Ultralytics dan menghasilkan model terlatih yang selanjutnya digunakan pada tahap evaluasi. Model yang diperoleh kemudian diuji untuk mengetahui kemampuan deteksi terhadap tiga kelas, yaitu Daun Sehat, Hawar Daun, dan Karat Daun. Hasil pelatihan model ditunjukkan pada Gambar 5.

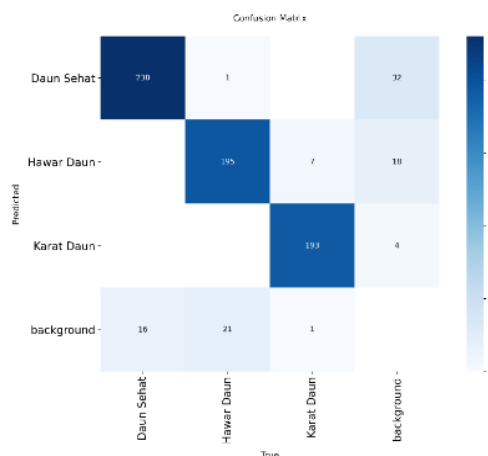
epoch	GPU mem	loss	cls loss	dfl loss	Instances	Size	train	val
45/50	8.22G	0.1874	0.137	0.0883	8	640	100%	312/312 1.611/s 4:09
Class								
all		0.000	0.000	0.000	0.000	0.000	0.000	19/19 2.111/s 9.2s
Epoch	GPU mem	loss	cls loss	dfl loss	Instances	Size		
45/50	8.22G	0.183	0.1305	0.0792	7	640	100%	312/312 1.611/s 4:09
Class								
all		0.000	0.000	0.000	0.000	0.000	0.000	19/19 2.011/s 9.4s
Epoch	GPU mem	loss	cls loss	dfl loss	Instances	Size		
46/50	8.22G	0.1821	0.1323	0.0845	8	640	100%	312/312 1.611/s 4:08
Class								
all		0.000	0.000	0.000	0.000	0.000	0.000	19/19 2.011/s 9.3s
Epoch	GPU mem	loss	cls loss	dfl loss	Instances	Size		
47/50	8.22G	0.177	0.1300	0.0793	7	640	100%	312/312 1.611/s 4:08
Class								
all		0.000	0.000	0.000	0.000	0.000	0.000	19/19 2.011/s 9.4s
Epoch	GPU mem	loss	cls loss	dfl loss	Instances	Size		
48/50	8.22G	0.1804	0.1375	0.0843	7	640	100%	312/312 1.611/s 4:09
Class								
all		0.000	0.000	0.000	0.000	0.000	0.000	19/19 2.011/s 9.3s
Epoch	GPU mem	loss	cls loss	dfl loss	Instances	Size		
49/50	8.22G	0.1814	0.1307	0.0811	7	640	100%	312/312 1.611/s 4:09
Class								
all		0.000	0.000	0.000	0.000	0.000	0.000	19/19 2.111/s 9.2s
Epoch	GPU mem	loss	cls loss	dfl loss	Instances	Size		
50/50	8.22G	0.1825	0.1358	0.081	9	640	100%	312/312 1.611/s 4:08
Class								
all		0.000	0.000	0.000	0.000	0.000	0.000	19/19 2.011/s 9.4s

Gambar 5. *Epoch Training*

Berdasarkan Gambar 5, proses pelatihan model menunjukkan perubahan nilai *loss* dan metrik evaluasi pada setiap *epoch*. Pada akhir pelatihan, model memperoleh *precision* sebesar 0,961, *recall* sebesar 0,922, *mAP@50* sebesar 0,952, dan *mAP@50-95* sebesar 0,691. Model hasil pelatihan selanjutnya digunakan pada tahap evaluasi untuk mengetahui kemampuan deteksi terhadap tiga kelas, yaitu Daun Sehat, Hawar Daun, dan Karat Daun.

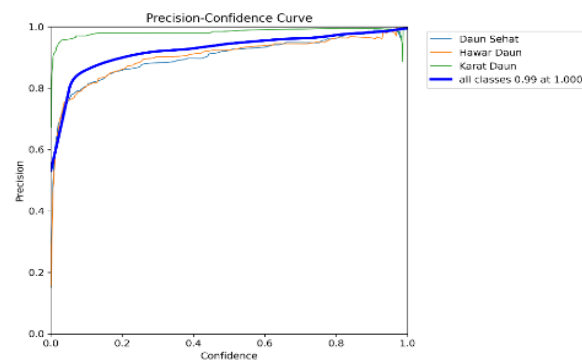
3.6 Evaluasi Model YOLOv9m

Pada tahap evaluasi, model YOLOv9m yang telah dilatih menghasilkan beberapa keluaran evaluasi untuk mengetahui kinerja model dalam mendeteksi tiga kelas kondisi daun tanaman jagung, yaitu Daun Sehat, Hawar Daun, dan Karat Daun. Hasil evaluasi berupa *confusion matrix* ditunjukkan pada Gambar 6, *precision-confidence curve* pada Gambar 7, *recall-confidence curve* pada Gambar 8, *precision-recall curve* pada Gambar 9, dan *F1-confidence curve* pada Gambar 10. Sementara itu, hasil evaluasi berdasarkan nilai *precision*, *recall*, *mAP@50*, dan *mAP@50-95* pada masing-masing kelas disajikan pada Tabel 4.



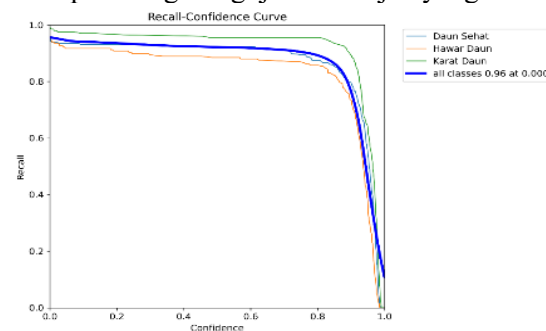
Gambar 6. *Confusion matrix* Model YOLOv9m

Berdasarkan Gambar 6, *confusion matrix* menunjukkan kemampuan model dalam mendeteksi Daun Sehat, Hawar Daun, dan Karat Daun. Nilai pada diagonal menunjukkan deteksi yang sesuai dengan kelas sebenarnya, sedangkan nilai di luar diagonal menunjukkan kesalahan deteksi antarkelas. *Background* digunakan untuk menunjukkan objek yang tidak terdeteksi atau prediksi objek pada area yang tidak memiliki objek sesuai anotasi.



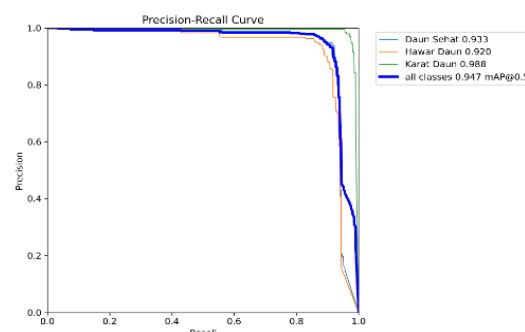
Gambar 7. *Precision-Confidence Curve* Model YOLOv9m

Berdasarkan Gambar 7, kurva *precision-confidence* menunjukkan hubungan antara nilai *precision* dan *confidence threshold* pada masing-masing kelas. Secara umum, nilai *precision* meningkat seiring dengan meningkatnya *confidence threshold*. Kelas Karat Daun menunjukkan performa paling tinggi dan relatif stabil pada sebagian besar rentang *confidence*. Sementara itu, kelas Daun Sehat dan Hawar Daun menunjukkan pola peningkatan *precision* secara bertahap. Pada *confidence* mendekati 1,00, nilai *precision* gabungan seluruh kelas mencapai sekitar 0,99. Kurva ini menunjukkan bahwa peningkatan *confidence threshold* dapat menghasilkan prediksi yang lebih tepat, meskipun dapat mengurangi jumlah objek yang terdeteksi.



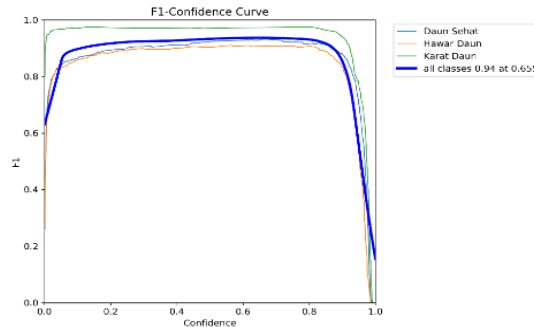
Gambar 8. *Recall-Confidence Curve* Model YOLOv9m

Berdasarkan Gambar 8, kurva *recall-confidence* menunjukkan hubungan antara nilai *recall* dan *confidence threshold* pada masing-masing kelas. Pada *confidence* rendah, model memiliki nilai *recall* yang relatif tinggi karena lebih banyak objek yang dapat terdeteksi. Seiring meningkatnya *confidence threshold*, nilai *recall* cenderung menurun karena model menjadi lebih selektif terhadap prediksi yang dihasilkan. Kelas Karat Daun menunjukkan nilai *recall* yang relatif lebih tinggi dibandingkan kelas lainnya pada sebagian besar rentang *confidence*. Secara keseluruhan, nilai *recall* mencapai sekitar 0,96 pada *confidence* 0,000.



Gambar 9. *Precision-Recall Curve* Model YOLOv9m

Berdasarkan Gambar 9, kurva *precision-recall* menunjukkan hubungan antara nilai *precision* dan *recall* pada masing-masing kelas. Nilai AP yang diperoleh untuk kelas Daun Sehat sebesar 0,933, Hawar Daun sebesar 0,920, dan Karat Daun sebesar 0,988. Nilai mAP@0.5 secara keseluruhan mencapai 0,947. Kelas Karat Daun memiliki nilai AP tertinggi dibandingkan kelas lainnya, sedangkan Hawar Daun memiliki nilai AP terendah. Hasil tersebut menunjukkan bahwa model YOLOv9m memiliki kemampuan deteksi yang baik pada ketiga kelas, dengan performa terbaik pada kelas Karat Daun.



Gambar 10. F1-Confidence Curve Model YOLOv9m

Berdasarkan Gambar 10, kurva *F1-confidence* menunjukkan hubungan antara nilai *F1-score* dan *confidence threshold* pada masing-masing kelas. Nilai *F1-score* meningkat pada awal peningkatan *confidence*, kemudian relatif stabil sebelum menurun pada *confidence* yang mendekati 1,00. Berdasarkan kurva, nilai *F1-score* gabungan seluruh kelas mencapai sekitar 0,94 pada *confidence threshold* 0,655. Kelas Karat Daun menunjukkan nilai *F1-score* tertinggi dibandingkan kelas Daun Sehat dan Hawar Daun. Nilai *confidence threshold* 0,655 dapat digunakan sebagai titik yang memberikan keseimbangan antara *precision* dan *recall* pada model.

Tabel 4. Hasil Evaluasi Model YOLOv9m pada Data Validasi

Kelas	Images	Instances	Precision	Recall	mAP@50	mAP@50-95
Daun Sehat	200	246	0.946	0.915	0.933	0.898
Hawar Daun	200	217	0.945	0.914	0.920	0.872
Karat Daun	200	201	0.992	0.995	0.988	0.974
All	600	664	0.961	0.915	0.947	0.915

Berdasarkan Tabel 4, hasil evaluasi model YOLOv9m pada data validasi menunjukkan bahwa model mampu mendeteksi ketiga kelas kondisi daun dengan performa yang baik. Kelas Karat Daun memperoleh hasil terbaik dengan *precision* sebesar 0,992, *recall* sebesar 0,995, mAP@50 sebesar 0,988, dan mAP@50-95 sebesar 0,974. Sementara itu, kelas Hawar Daun memperoleh nilai mAP@50 terendah sebesar 0,920. Secara keseluruhan, model memperoleh *precision* sebesar 0,961, *recall* sebesar 0,915, mAP@50 sebesar 0,947, dan mAP@50-95 sebesar 0,915. Hasil tersebut menunjukkan bahwa model YOLOv9m memiliki kemampuan yang baik dalam mendeteksi Daun Sehat, Hawar Daun, dan Karat Daun pada data validasi.

3.7 Hasil Konversi dan Implementasi Model

Setelah proses pelatihan dan evaluasi, model terbaik (*best model*) YOLOv9m yang tersimpan dalam format .pt dikonversi ke format TensorFlow Lite (.tflite) menggunakan *framework* Ultralytics. Proses konversi dilakukan menggunakan fungsi *export* dengan format tflite seperti ditunjukkan pada Gambar 11.

```
from ultralytics import YOLO
model = YOLO("/content/drive/MyDrive/bakti-2/detect/train/weights/best.pt")
# Export the model to TFLite format
model.export(format="tflite")
```

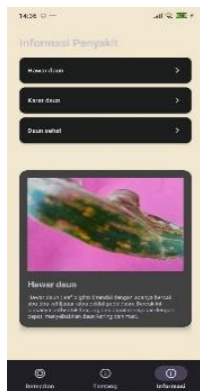
Gambar 11. Proses Konversi Model YOLOv9m ke Format TensorFlow Lite

3.8 Hasil Implementasi Aplikasi Android

Setelah model YOLOv9m dikonversi ke format *TensorFlow Lite* (.tflite), model diintegrasikan ke dalam aplikasi Android menggunakan Android Studio. Aplikasi yang dikembangkan memiliki tiga menu utama, yaitu Deteksi, Informasi, dan Tentang. Hasil implementasi antarmuka aplikasi ditunjukkan pada Gambar 12.



(a) Halaman Deteksi



(b) Halaman Informasi



(c) Halaman Tentang

Gambar 12. Implementasi Antarmuka Aplikasi Android

Berdasarkan Gambar 12, aplikasi menyediakan halaman Deteksi untuk melakukan deteksi kondisi daun tanaman jagung, halaman Informasi untuk menampilkan informasi mengenai jenis kondisi daun, serta halaman Tentang yang berisi informasi mengenai aplikasi.

3.9 Hasil Pengujian Sistem

Setelah aplikasi Android selesai diimplementasikan, dilakukan pengujian untuk mengetahui apakah fungsi-fungsi aplikasi berjalan sesuai dengan kebutuhan yang telah ditentukan. Pengujian dilakukan menggunakan *Black box Testing* dan *Usability Testing*. Pengujian *black box* dilakukan dengan menguji fungsi utama pada aplikasi tanpa melihat kode program. Hasil pengujian *Black box* ditunjukkan pada Tabel 5.

Tabel 5. *Black box Testing*

Uji Kasus	Skenario Uji Coba	Hasil yang Diharapkan	Status
Menjalankan sistem	Mengakses dan menjalankan aplikasi deteksi kondisi daun jagung	Aplikasi dapat diakses dan dijalankan dengan lancar	Sesuai
Menekan Tombol Kamera	Menekan tombol kamera untuk mengaktifkan kamera sebagai input citra	Aplikasi dapat mengakses kamera untuk mengambil citra yang akan dideteksi	Sesuai
Mengenal Objek	Mengarahkan kamera pada daun jagung untuk melakukan deteksi	Aplikasi dapat mendeteksi kondisi daun jagung sesuai dengan objek yang diberikan	Sesuai
Menampilkan Output	Menampilkan hasil deteksi kondisi daun jagung	Sistem menampilkan hasil deteksi berupa <i>bounding box</i> dan label kelas pada objek yang terdeteksi	Sesuai
Tombol informasi	Menekan tombol Informasi untuk menampilkan informasi kondisi daun	Sistem dapat menampilkan informasi mengenai kondisi daun jagung	Sesuai
Tombol Tentang Aplikasi	Menekan tombol Tentang untuk menampilkan informasi aplikasi	Sistem dapat menampilkan halaman informasi tentang aplikasi	Sesuai

Berdasarkan Tabel 5, seluruh skenario pengujian dapat dijalankan sesuai dengan hasil yang diharapkan. Fitur utama aplikasi, mulai dari menjalankan sistem, mengakses kamera, melakukan deteksi, menampilkan hasil deteksi, hingga mengakses halaman Informasi dan Tentang Aplikasi, memperoleh status sesuai. Dengan demikian, fungsi utama aplikasi dapat berjalan dengan baik berdasarkan hasil *Black box Testing*. Setelah fungsi aplikasi dinyatakan berjalan dengan baik, pengujian dilanjutkan dengan *usability testing* untuk mengetahui tingkat kemudahan penggunaan aplikasi dari sisi pengguna. Pengujian dilakukan menggunakan kuesioner dengan lima pernyataan yang dinilai menggunakan skala *Likert* 1–5, mulai dari Sangat Tidak Setuju hingga Sangat Setuju. Responden memberikan penilaian setelah menggunakan aplikasi. Hasil *usability testing* disajikan pada Tabel 6.

Tabel 6. Hasil *Usability Testing*

No.	Pernyataan	Jumlah Responden	Skor Diperoleh	Skor Maksimal	Persentase
1	Aplikasi berjalan dengan baik	32	144	160	90%
2	Tampilan aplikasi mudah dipahami	32	138	160	86,25%
3	Aplikasi dapat mendeteksi kondisi daun jagung	32	138	160	86,25%
4	Aplikasi berguna bagi petani maupun orang yang ingin menanam jagung	32	136	160	85%
5	Fitur yang digunakan pada aplikasi berjalan dengan baik	32	136	160	85%
Rata-rata		32			86,5%

Berdasarkan Tabel 6, hasil *Usability Testing* terhadap 32 responden menunjukkan bahwa persentase penilaian tertinggi terdapat pada pernyataan bahwa aplikasi berjalan dengan baik, yaitu sebesar 90%. Sementara itu, persentase terendah sebesar 85% terdapat pada aspek kebermanfaatan aplikasi dan kinerja fitur. Secara keseluruhan, hasil penilaian dari lima pernyataan memperoleh rata-rata persentase sebesar 86,5%. Hasil tersebut menunjukkan bahwa aplikasi memperoleh penilaian yang baik dari responden dalam aspek kemudahan penggunaan, fungsi deteksi, tampilan, kebermanfaatan, dan kinerja fitur.

4. KESIMPULAN

Berdasarkan hasil penelitian dan pengembangan yang telah dilakukan, dapat disimpulkan bahwa:

1. Penelitian berhasil mengimplementasikan model deteksi kondisi daun tanaman jagung menggunakan YOLOv9m dengan tiga kelas, yaitu Daun Sehat, Hawar Daun, dan Karat Daun. *Dataset* awal sebanyak 3.000 citra diproses menggunakan Roboflow melalui *resize* dan *augmentasi* sehingga menghasilkan 7.163 citra yang kemudian dibagi menjadi data *training*, *validation*, dan *testing*.
2. Model YOLOv9m berhasil dilatih dengan konfigurasi 50 *epoch* dan ukuran citra masukan 640 × 640 piksel. Berdasarkan hasil evaluasi pada data validasi, model memperoleh *precision*

sebesar 0,961, *recall* sebesar 0,915, mAP@50 sebesar 0,947, dan mAP@50–95 sebesar 0,915. Kelas Karat Daun menunjukkan performa deteksi tertinggi dengan mAP@50 sebesar 0,988.

3. Model YOLOv9m berhasil dikonversi ke format TensorFlow Lite (.tflite) dan diintegrasikan ke dalam aplikasi Android. Aplikasi menyediakan fitur utama berupa deteksi kondisi daun, informasi mengenai kondisi daun, dan informasi tentang aplikasi.
4. Hasil *Black box Testing* menunjukkan bahwa seluruh fungsi utama aplikasi dapat berjalan sesuai dengan yang diharapkan. Selanjutnya, hasil *Usability Testing* terhadap 32 responden memperoleh nilai rata-rata sebesar 86,5%, yang menunjukkan bahwa aplikasi memperoleh penilaian yang baik dari pengguna dalam aspek kemudahan penggunaan, tampilan, fungsi deteksi, kebermanfaatan, dan kinerja fitur.

Secara keseluruhan, penelitian ini berhasil menghasilkan aplikasi Android berbasis YOLOv9m yang dapat digunakan untuk membantu mendeteksi kondisi daun tanaman jagung, khususnya Daun Sehat, Hawar Daun, dan Karat Daun.

5. SARAN

Berdasarkan hasil dan kesimpulan penelitian yang telah dilakukan, saran yang dapat diberikan adalah:

1. Menambah jumlah dan keberagaman *dataset* agar model dapat mengenali kondisi daun pada berbagai situasi.
2. Menggunakan data *testing* yang benar-benar terpisah untuk memperoleh evaluasi model yang lebih objektif.
3. Membandingkan YOLOv9m dengan model lain dan melakukan optimasi agar lebih ringan dan efisien pada Android.
4. Menambahkan fitur riwayat deteksi, penyimpanan hasil, serta rekomendasi penanganan.
5. Menambah jumlah responden dan menggunakan instrumen *usability* yang lebih teruji.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan, bantuan, dan kontribusi selama proses penelitian dan penyusunan artikel ini. Ucapan terima kasih secara khusus disampaikan kepada Bapak Sutardi selaku Kepala Laboratorium Pengamatan Hama dan Penyakit Tanaman Kabupaten Sorong yang telah membantu dalam proses validasi *dataset*. Penulis juga menyampaikan terima kasih kepada seluruh responden yang telah berpartisipasi dalam proses pengujian aplikasi. Semoga penelitian ini dapat memberikan manfaat dan menjadi referensi bagi pengembangan penelitian selanjutnya.

DAFTAR PUSTAKA

- [1] P. A. Beding, N. E. Lewaherilla, R. H. Lestari, and S. Tirajoh, “Analisis Potensi Pengembangan Komoditas Jagung Di Wilayah Perbatasan Nkri-Png Kabupaten Keerom Papua,” *J. Sos. Ekon. Pertan. dan Agribisnis*, vol. 20, no. 2, pp. 162–170, 2023, doi: 10.20961/sepa.v20i2.52339.
- [2] Badan Pusat Statistik Provinsi Papua Barat, “Pada 2025, luas panen jagung pipilan di Provinsi Papua Barat Daya mencapai 0,43 ribu hektare. Produksi jagung pipilan kering dengan kadar air 14 persen pada 2025 sebanyak 2,73 ribu ton.” [Online]. Available: <https://papuabar.bps.go.id/id/pressrelease/2026/03/02/pada-2025--luas-panen-jagung-pipilan-di-provinsi-papua-barat-daya-mencapai-0-43-ribu-hektare--produksi-jagung-pipilan-kering-dengan-kadar-air-14-persen-pada-2025-sebanyak-2-73-ribu-ton-.html>.
- [3] D. Putri and U. Andalas, “Interaksi Faktor Lingkungan dan Sistem Budidaya terhadap Serangan Penyakit Karat Daun pada Tanaman Jagung,” *J. Kridatama Sains dan Teknol.*, vol. 7, no. 01, pp. 383–392, 2025, doi: 10.53863/kst.v7i01.1636.
- [4] D. Fivalianda and A. Desiani, “Penerapan Sistem Pakar Menggunakan Metode Certainty

-
- Factor untuk Diagnosis Penyakit pada Tanaman Jagung,” *J. Math. Math. Educ.*, vol. 6, no. 2, pp. 129–137, 2024, doi: <http://dx.doi.org/10.21580/square.2024.6.2.16042>.
- [5] Arsi, F. N. Kurnia, and Suparman, “Evaluasi Pengelolaan Penyakit Tanaman Terpadu pada Petani Jagung (*Zea mays* L.) Di desa Suka Menang, Kecamatan Gelumbang, Kabupaten Muara Enim, Sumatera Selatan,” *J. Tanam. Pangan dan Hortik.*, vol. 6, no. April, pp. 1–19, 2024, doi: <https://doi.org/10.25181/jplantasimbiosa.v6i1.2978>.
- [6] A. A. Sidik, B. Nugroho, and U. Sudadi, “Pengaruh Bulai pada Perubahan Indeks Kadar Klorofil, Serapan Fosforus dan Boron pada Jagung Manis,” *J. Ilmu Pertan. Indones.*, vol. 29, no. 4, pp. 667–675, 2024, doi: [10.18343/jipi.29.4.667](https://doi.org/10.18343/jipi.29.4.667).
- [7] I. G. N. A. Santika, N. G. A. P. H. Saptarini, I. P. A. Prayudha, and G. B. Subiksa, “Evaluasi Model Convolutional Neural Network (CNN) Dalam Klasifikasi Penyakit Daun Jagung Berbasis Web Menggunakan Citra Digital,” *J. Manaj. Inform. Komputerisasi Akunt.*, vol. 10, no. 1, pp. 98–106, 2026, doi: [10.46880/jmika.Vol10No1.pp98-106](https://doi.org/10.46880/jmika.Vol10No1.pp98-106).
- [8] W. R. Hakim, A. A. Ilyas, B. Riswanto, M. H. Somaída, and D. Yusuf, “Prototipe Aplikasi Identifikasi Penyakit Pada Daun Jagung Berbasis Android Menggunakan Metode Scrum,” *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 3, 2025, doi: [10.23960/jitet.v13i3S1.7688](https://doi.org/10.23960/jitet.v13i3S1.7688).
- [9] R. H. J. Notonegoro, D. A. Rahayu, D. Ikasari, and R. Kosasih, “Teknologi Ai Pada Budidaya Vanili Menuju Pertanian Pintar : Review,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 13, no. 2, pp. 309–318, 2026, doi: [10.25126/jtiik.132](https://doi.org/10.25126/jtiik.132).
- [10] A. Pangestu and M. P. K. Putra, “YOLOv9-Based Traffic Sign Detection Under Varying Lighting,” *J. Tek. Inform.*, vol. 6, no. 1, pp. 291–300, 2025, doi: [10.52436/1.jutif.2025.6.1.3917](https://doi.org/10.52436/1.jutif.2025.6.1.3917).
- [11] D. W. V. Likadja, D. R. Sina, and B. S. Djahi, “Deteksi Hama Fall Armyworm Pada Tanaman Jagung Menggunakan YOLO Di Desa Oesao, Kabupaten Kupang, Nusa Tenggara Timur,” *J. Mhs. Tek. Inform.*, vol. 10, no. 3, pp. 4450–4456, 2026, doi: [10.36040/jati.v10i3.18227](https://doi.org/10.36040/jati.v10i3.18227).
- [12] E. Tenda, S. C. W. Ngangi, C. A. J. Soewoeh, and E. Ketaren, “Sistem Deteksi Hama Pada Tanaman Jagung (*Zea Mays*) Berbasis Kecerdasan Buatan Dan Internet Of Things (Iot),” *J. TIMES*, vol. XIV, no. 2, pp. 225–233, 2025, doi: [10.51351/jtm.14.2.2025879](https://doi.org/10.51351/jtm.14.2.2025879).
- [13] H. P. Putro, B. Tumanggor, and D. Supriatna, “Deteksi Penyakit Daun Padi Menggunakan Model Deteksi Objek YOLO,” *J. Ilm. Sist. Inf.*, vol. 5, no. 2, pp. 664–673, 2026, doi: [10.51903/hn9qqt19](https://doi.org/10.51903/hn9qqt19).
- [14] M. I. Aulia, A. Yudhana, and Sunardi, “Optimasi Deteksi Hama Tanaman Melon Berbasis YOLOv9,” *J. Inform. Technol. Commun.*, vol. 10, no. 1, pp. 150–161, 2026, doi: [10.36596/jitu.v10i1.2265](https://doi.org/10.36596/jitu.v10i1.2265).
- [15] L. Mahdiyah, S. Oktamuliani, and W. L. Putri, “Penerapan Algoritma Deep Learning YOLOv8 pada Platform Roboflow untuk Segmentasi Citra Panoramik,” *J. Fis. Unand*, vol. 14, no. 3, pp. 228–234, 2025, doi: [10.25077/jfu.14.3.228-234.2025](https://doi.org/10.25077/jfu.14.3.228-234.2025).
- [16] E. F. Pangestu and B. Irawan, “Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Convolutional Neural Network Dengan Arsitektur Mobilnetv2,” *J. Inform. dan Tek. Elektro Terap.*, vol. 14, no. 1, 2026, doi: [10.23960/jitet.v14i1.8900](https://doi.org/10.23960/jitet.v14i1.8900).
- [17] R. Sapkota, Z. Meng, M. Churuvija, X. Du, Z. Ma, and M. Karkee, “Comprehensive performance evaluation of YOLOv12, YOLO11, YOLOv10, YOLOv9 and YOLOv8 on detecting and counting fruitlet in complex orchard environments,” *Agric. Commun.*, vol. 4, no. 1, p. 100125, 2026, doi: [10.1016/j.agrcom.2026.100125](https://doi.org/10.1016/j.agrcom.2026.100125).
- [18] M. Barr, “Coupling the Power of YOLOv9 with Transformer for Small Object Detection in Remote-Sensing Images,” *Comput. Model. Eng. Sci.*, vol. 143, no. 1, 2025, doi: [10.32604/cmcs.2025.062264](https://doi.org/10.32604/cmcs.2025.062264).
- [19] Muhammad and I. Susilawati, “Model Cnn Berbasis Efficientnet-Lite Untuk Deteksi Retinopati Diabetik Menggunakan Tensorflow Lite,” *J. Mhs. Tek. Inform.*, vol. 9, no. 5, pp. 8754–8761, 2025, doi: [10.36040/jati.v9i5.15192](https://doi.org/10.36040/jati.v9i5.15192).
- [20] M. Edoazani, S. Rizal, N. Adha, O. Saputri, and M. S. Putra, “Implementasi Aplikasi Android Deteksi Penyakit Tanaman Tomat Berbasis Cloud menggunakan MVVM,” *J.*
-

Ilm. MEDIA SISFO, vol. 19, no. 2, pp. 202–219, 2025, doi:
10.33998/mediasisfo.2025.19.2.2564.