

## Perbandingan Kinerja YOLOv11 pada Konfigurasi *Baseline* dan Optimasi *Hyperparameter* untuk Deteksi Hama Tanaman Cabai

### *Performance Comparison of YOLOv11 under Baseline and Hyperparameter Optimization Configurations for Chili Pest Detection*

ATTIYA DIANTI FADLI<sup>1\*</sup>, ERNAWATI<sup>1</sup>, DESI ANDRESWARI<sup>1</sup>

#### Abstrak

Tanaman cabai (*Capsicum annuum* L.) merupakan komoditas hortikultura strategis di Indonesia, namun produktivitasnya sering menurun akibat serangan hama. Deteksi dini terhadap serangan hama masih menjadi kendala, terutama karena pendekatan konvensional yang digunakan umumnya memiliki keterbatasan dalam mengidentifikasi gejala secara cepat dan akurat. Perkembangan *computer vision* membuka peluang penerapan sistem deteksi hama otomatis berbasis citra digital untuk mendukung pertanian presisi. Penelitian ini bertujuan untuk membandingkan kinerja YOLOv11 pada konfigurasi *baseline* dan optimasi *hyperparameter* dalam mendeteksi hama tanaman cabai. *Dataset* publik yang digunakan terdiri atas 4957 citra yang sudah memiliki anotasi dengan empat kelas hama utama. Model dilatih selama 300 *epoch* dengan arsitektur dan *dataset* yang sama pada kedua konfigurasi. Evaluasi menggunakan metrik *precision*, *recall*, *mAP@0.5*, dan *mAP@0.5–0.95*. Pada data validasi, model *baseline* memperoleh *precision* 86%, *recall* 74%, *mAP@0.5* sebesar 78%, dan *mAP@0.5–0.95* sebesar 34%. Model dengan optimasi *hyperparameter* mencapai *precision* sebesar 85%, *recall* 78%, *mAP@0.5* sebesar 80.2%, dan *mAP@0.5–0.95* sebesar 37%. Hasil pengujian dengan data uji pada model *baseline* memperoleh *precision* 89%, *recall* 81%, *mAP@0.5* sebesar 83%, dan *mAP@0.5–0.95* sebesar 38%. Hasil pengujian dengan data uji pada model optimasi menunjukkan *precision* sebesar 85%, *recall* 82%, *mAP@0.5* sebesar 81%, dan *mAP@0.5–0.95* sebesar 38%. Hal ini menunjukkan bahwa kinerja keduanya relatif sebanding. Analisis kurva pelatihan dan *confusion matrix* menunjukkan adanya perbedaan dinamika konvergensi antara kedua konfigurasi, namun pada evaluasi akhir keduanya menunjukkan kemampuan generalisasi yang relatif setara.

Kata Kunci: cabai, *computer vision*, deteksi, *hyperparameter*, YOLOv11

#### Abstract

*The chili pepper (Capsicum annuum L.) is a key horticultural commodity in Indonesia with significant economic value. Its productivity is frequently reduced by pest infestations, for which early-stage detection remains challenging when relying solely on conventional approaches. Recent advances in computer vision have enabled the development of automated, image-based pest detection systems to support precision agriculture. This study aims to compare the performance of the YOLOv11 model under baseline and hyperparameter-optimized configurations for chili pest detection. The analysis utilizes a public dataset consisting of 4957 annotated images covering four major pest classes. Both models were trained for 300 epochs using identical architectures and datasets. Model performance was evaluated using precision, recall, mAP@0.5, and mAP@0.5–0.95 metrics. On the validation set, the baseline model achieved 86% precision, 74% recall, 78.0% mAP@0.5, and 34% mAP@0.5–0.95, while the optimized model achieved 85% precision, 78% recall, 80.2% mAP@0.5, and 37% mAP@0.5–0.95. On the test set, the baseline model obtained 89% precision, 81% recall, 83% mAP@0.5, and 38% mAP@0.5–0.95, whereas the optimized model achieved 85% precision, 82% recall, 81% mAP@0.5, and 38% mAP@0.5–0.95, indicating comparable performance between the two configurations. Furthermore, analysis of training curves and confusion matrices reveals differences in convergence behavior across configurations; however, both models demonstrate similar generalization capabilities on the test set.*

Keywords: chili peppers, *computer vision*, detection, *hyperparameter*, YOLOv11

<sup>1</sup>Jurusan Informatika, Fakultas Teknik, Universitas Bengkulu, Bengkulu, Indonesia.

\*Penulis Korespondensi: Surel: attiyadiantifadli@gmail.com

## PENDAHULUAN

Tanaman cabai (*Capsicum annuum L.*) dikategorikan sebagai komoditas hortikultura utama di Indonesia dengan tingkat intensitas budidaya yang tinggi, serta memiliki signifikansi ekonomi bagi petani karena tingginya kebutuhan konsumsi domestik dan perannya dalam menjaga stabilitas pasokan pangan di tingkat lokal dan nasional (Utomo *et al.* 2024). Data Badan Pusat Statistik (BPS) menunjukkan bahwa produksi cabai nasional masih tergolong tinggi dalam kelompok sayuran semusim. Berdasarkan tabel produksi tanaman sayuran dan buah-buahan semusim menurut jenis tanaman tahun 2024 dari BPS, produksi cabai rawit mencapai sekitar 1.57 juta ton, cabai keriting sekitar 1.09 juta ton, dan cabai besar sekitar 384 ribu ton. Fakta ini menempatkan cabai sebagai salah satu komoditas hortikultura strategis dengan kontribusi signifikan terhadap produksi sayuran nasional (BPS 2024). Kondisi tersebut menunjukkan bahwa keberlanjutan produksi cabai memiliki pengaruh langsung terhadap stabilitas harga dan pengendalian inflasi pangan.

Selain itu, cabai merupakan komoditas strategis yang berperan dalam meningkatkan pendapatan petani, mengurangi kemiskinan, membuka kesempatan kerja, serta mendukung ekspor dan mengurangi impor komoditas hortikultura (Kristiana *et al.* 2025). Produktivitas cabai yang stabil juga menjadi faktor penting dalam menjaga pasokan dan menstabilkan harga di pasar lokal, sehingga memberikan kontribusi terhadap ketahanan pangan dan stabilitas ekonomi di tingkat komunitas pertanian (Irjang *et al.* 2024).

Meskipun demikian, tingkat produktivitas tanaman cabai masih menghadapi tantangan serius akibat serangan Organisme Pengganggu Tanaman (OPT), khususnya hama. Beberapa jenis hama utama yang umum menyerang tanaman cabai antara lain *Myzus persicae*, *Bemisia tabaci*, *Thrips*, *Spodoptera litura*, dan *Helicoverpa armigera* dilaporkan menyebabkan penurunan hasil panen yang signifikan di berbagai sentra produksi cabai di Indonesia (Ekawati *et al.* 2023). Serangan hama tidak hanya menurunkan kuantitas hasil panen, tetapi juga berdampak pada penurunan kualitas produk apabila tidak dikendalikan secara tepat (Yasen *et al.* 2023). Dalam skala yang lebih luas, penurunan produksi akibat serangan hama berkontribusi terhadap fluktuasi pasokan dan ketidakstabilan harga cabai di pasar (Rohmat *et al.* 2020).

Upaya pengendalian hama yang diterapkan di lapangan hingga saat ini masih didominasi oleh penggunaan pestisida kimia dan pengamatan visual secara manual. Sejumlah penelitian melaporkan bahwa penggunaan pestisida kimia secara intensif berpotensi menimbulkan dampak negatif terhadap kesehatan manusia serta degradasi kualitas lingkungan dalam jangka panjang (Jamin *et al.* 2024). Di sisi lain, pendekatan pengamatan visual manual memiliki keterbatasan dalam mendeteksi keberadaan hama secara dini dan konsisten, terutama pada kondisi objek berukuran kecil dan latar belakang visual yang kompleks. Keterbatasan ini menunjukkan bahwa pendekatan manual yang digunakan saat ini belum sepenuhnya mampu mendukung pengendalian hama secara presisi dan berkelanjutan.

Perkembangan teknologi *computer vision* dan *machine learning* membuka peluang penerapan sistem pertanian presisi melalui pemanfaatan analisis citra digital. Algoritma *You Only Look Once* (YOLO) merupakan metode deteksi objek satu tahap yang mampu melakukan inferensi secara *real-time* dengan tingkat akurasi yang kompetitif (Santoso *et al.* 2025). Penerapan YOLO dalam bidang pertanian telah dilaporkan mampu meningkatkan performa deteksi hama tanaman, termasuk melalui adaptasi YOLOv5 yang efektif dalam menangani variasi skala objek dan kompleksitas latar belakang pada *dataset* hama (Dong *et al.* 2024). Keunggulan tersebut menjadikan YOLO sebagai salah satu pendekatan yang potensial untuk mendukung pengembangan sistem deteksi hama otomatis.

Penerapan YOLO secara lebih spesifik pada tanaman cabai dilaporkan pada penelitian Riva dan Jayanta tahun 2023, yang menggunakan YOLOv5 untuk mendeteksi penyakit tanaman cabai berbasis citra digital. Penelitian tersebut menunjukkan performa deteksi yang tinggi dengan nilai *precision* sebesar 0.946, *recall* 0.936, dan *mean Average Precision* (mAP) sebesar 0.959 pada konfigurasi pembagian data terbaik. Hasil ini mengindikasikan bahwa YOLOv5 efektif dalam menangani objek visual pada tanaman cabai. Namun, penelitian

tersebut masih berfokus pada versi YOLO generasi sebelumnya serta tidak mengeksplorasi pengaruh pengaturan *hyperparameter* pelatihan terhadap stabilitas dan generalisasi kinerja model.

Penelitian lain yang relevan dilaporkan pada penelitian Agustian *et al.* (2023), yang mengembangkan model deteksi hama tanaman cabai berbasis *fine-tuned* YOLOv5 menggunakan *dataset* lapangan berskala besar dengan karakteristik objek berukuran kecil. Penelitian tersebut melaporkan nilai mAP@0.5 sebesar 81.3% pada data uji, dengan performa maksimum mAP@0.5 sebesar 82.6% dan kecepatan inferensi 20 ms per citra. Meskipun hasil tersebut menunjukkan kemampuan YOLOv5 dalam mendeteksi hama cabai secara efektif, penelitian tersebut masih terbatas pada konfigurasi arsitektur dan parameter pelatihan tertentu, serta belum mengkaji secara sistematis pengaruh *hyperparameter* utama sebagai variabel kunci dalam peningkatan kinerja model.

Seiring dengan perkembangan model deteksi objek, optimasi *hyperparameter* menjadi faktor penting dalam meningkatkan kinerja model berbasis *deep learning*. Penelitian Zulkarnain dan Kusri (2025) menunjukkan bahwa optimasi *hyperparameter* pada YOLOv11, seperti pengaturan *learning rate*, *momentum*, dan *weight decay* yang dikombinasikan dengan teknik *augmentasi* adaptif, mampu meningkatkan performa deteksi secara signifikan, dengan capaian mAP@0.5 hingga 0.97 pada skenario kondisi visual yang menantang. Hasil tersebut menegaskan bahwa kinerja YOLOv11 tidak hanya ditentukan oleh arsitektur, tetapi juga sangat bergantung pada strategi pelatihan yang diterapkan.

Temuan serupa mengenai pentingnya *hyperparameter* juga dilaporkan pada berbagai konteks aplikasi *computer vision*. Studi yang dilakukan oleh Isa dan Junedi (2022) menegaskan bahwa pemilihan nilai *epoch* yang tepat berpengaruh langsung terhadap stabilitas pelatihan dan akurasi model, di mana konfigurasi *epoch* optimal menghasilkan peningkatan akurasi pelatihan dan validasi secara signifikan. Dalam domain pertanian, penelitian Singh *et al.* (2023) menunjukkan bahwa variasi teknik optimasi dan parameter pelatihan dapat menghasilkan perbedaan kinerja yang nyata pada model *deep learning*, meskipun menggunakan *dataset* dan arsitektur yang sama. Sementara itu, penelitian Lokku *et al.* (2022) membuktikan bahwa optimasi *hyperparameter* pada jaringan CNN mampu meningkatkan ketahanan model terhadap gangguan visual seperti *noise* dan oklusi, dengan tingkat akurasi mencapai 97.2%. Studi yang dilakukan oleh Neshat *et al.* (2024) juga menegaskan bahwa proses *fine-tuning* dan pengaturan parameter pelatihan berperan penting dalam meningkatkan akurasi dan efisiensi komputasi pada model pembelajaran mendalam.

Meskipun berbagai penelitian tersebut menegaskan pentingnya *hyperparameter* dalam meningkatkan kinerja model *deep learning*, penelitian yang secara khusus mengevaluasi pengaruh optimasi *hyperparameter* pada model YOLO generasi terbaru, khususnya YOLOv11, masih terbatas dalam konteks deteksi hama tanaman cabai. Padahal, karakteristik objek hama yang berukuran kecil, variasi bentuk, serta latar belakang pertanian yang kompleks menuntut konfigurasi pelatihan yang optimal. Oleh karena itu, penelitian ini menjadi relevan untuk mengisi celah tersebut dengan menganalisis pengaruh optimasi *hyperparameter* terhadap kinerja YOLOv11 dalam deteksi hama pada tanaman cabai.

## **METODE**

### **Alur Penelitian**

Alur penelitian ini diawali dengan identifikasi masalah dan studi literatur untuk memperoleh landasan teoritis yang relevan. Selanjutnya, dilakukan akuisisi *dataset* publik yang diproses melalui tahap *preprocessing* dan pembagian data menjadi data *train*, data *valid*, dan data *test*. Model YOLO digunakan sebagai model dasar dengan konfigurasi *hyperparameter baseline*, kemudian dilakukan pelatihan dengan berbagai variasi *hyperparameter*. Model hasil pelatihan selanjutnya diuji menggunakan data uji, dan kinerjanya dievaluasi berdasarkan metrik evaluasi yang sesuai. Alur penelitian disajikan dalam bentuk diagram alir sebagaimana ditunjukkan pada Gambar 1.



Gambar 1 Alur penelitian

a. Identifikasi Masalah

Identifikasi masalah bertujuan untuk merumuskan fokus penelitian berdasarkan kondisi nyata di lapangan serta temuan dari penelitian sebelumnya. Salah satu permasalahan utama dalam budidaya tanaman cabai adalah tingginya risiko serangan hama yang sulit dideteksi secara awal dan akurat melalui metode konvensional, seperti pengamatan visual manual. Metode ini memiliki keterbatasan dalam konsistensi, akurasi dan efisiensi, terutama ketika objek hama berukuran kecil dan berada pada latar belakang yang rumit.

Di sisi lain, meskipun model deteksi objek berbasis *computer vision* seperti YOLO menunjukkan performa menjanjikan dalam berbagai aplikasi pertanian, kinerjanya masih sangat bergantung pada pengaturan pelatihan yang diterapkan. Penggunaan *hyperparameter* standar tidak selalu memberikan hasil terbaik pada citra hama tanaman cabai yang beragam dalam skala, pencahayaan, dan kepadatan objek. Oleh karena itu, penting untuk melakukan penelitian yang secara khusus meneliti dan mengevaluasi dampak pengaturan *hyperparameter* terhadap kinerja model YOLO dalam mendeteksi hama pada tanaman cabai.

b. Studi Literatur

Studi literatur dilakukan untuk mendapatkan pemahaman menyeluruh mengenai kemajuan penelitian yang berhubungan dengan deteksi hama tanaman menggunakan citra digital dan penerapan model deteksi yang berbasis *computer vision*. Penelitian ini mencakup penelitian-penelitian terkait deteksi hama dan penyakit tanaman, serta analisis kinerja berbagai varian YOLO yang telah ada.

Selain itu, penelitian ini juga menganalisis faktor-faktor yang memengaruhi kinerja model *computer vision*, dengan fokus pada peran *hyperparameter* pelatihan dalam memengaruhi stabilitas konvergensi dan kemampuan generalisasi model. Temuan dari studi literatur ini menjadi acuan untuk menentukan pendekatan penelitian, memilih arsitektur YOLO, serta merancang skema optimasi *hyperparameter* yang akan diterapkan dalam penelitian ini.

c. Pengumpulan Dataset

*Dataset* yang digunakan dalam penelitian ini merupakan *dataset* publik yang diperoleh dari Roboflow, yang menyediakan citra hama tanaman beserta anotasi *bounding box* untuk keperluan deteksi objek berbasis *computer vision*. Pemilihan *dataset* ini didasarkan pada kesesuaiannya dengan tujuan penelitian, yaitu mendeteksi hama pada tanaman cabai yang memiliki karakteristik objek berukuran kecil, variasi pose, serta latar belakang visual yang kompleks.

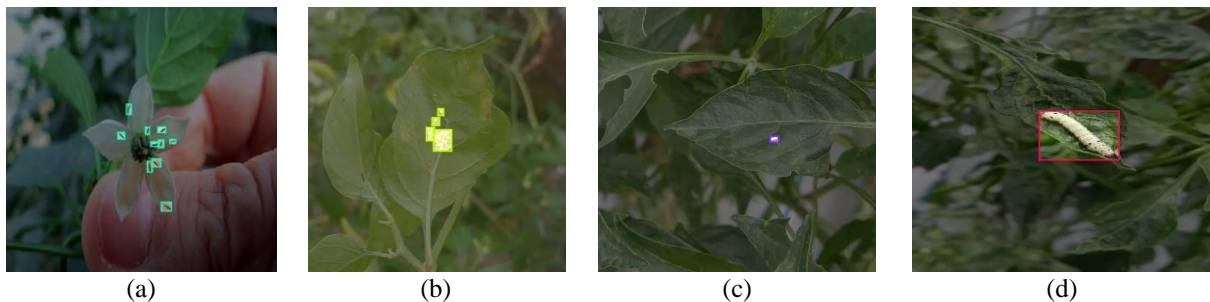
*Dataset* ini terdiri atas beberapa kelas hama utama yang umum ditemukan pada tanaman cabai, yaitu *thrips*, kutu daun, kutu kebul, dan ulat. Setiap citra dalam *dataset* telah melalui proses anotasi manual berupa *bounding box* yang menunjukkan lokasi objek hama. Informasi terperinci mengenai jumlah total citra, jumlah kelas hama, serta jumlah objek yang dianotasi pada setiap kelas disajikan dalam Tabel 1.

Tabel 1 Jumlah *dataset*

Nama Hama	Nama Kelas	Jumlah Citra	Jumlah Objek
<i>Thrips</i>	T	766	2740
Kutu Daun	MP	568	3589
Kutu Kebul	BT	1465	1727
Ulat	C	2158	2505
Total		4957	10,561

Distribusi jumlah citra dan objek pada Tabel 1 menunjukkan adanya ketidakseimbangan (*class imbalance*) antar kelas, di mana kelas tertentu seperti ulat (C) dan kutu daun (MP) memiliki jumlah citra dan objek lebih besar dibandingkan kelas lainnya. Ketidakseimbangan ini berpotensi memengaruhi proses pelatihan model, khususnya dalam meningkatkan bias prediksi terhadap kelas dominan. Untuk meminimalkan dampak tersebut, penelitian ini menerapkan *augmentasi* data pada *subset* data latih secara acak dan merata pada seluruh kelas. Pendekatan ini bertujuan meningkatkan variasi data dan membantu model mempelajari karakteristik visual setiap kelas secara lebih seimbang tanpa melakukan perubahan distribusi data secara langsung seperti *oversampling* atau *undersampling*.

Selain aspek numerik, karakteristik visual *dataset* juga ditampilkan melalui contoh citra dari setiap kelas hama. Gambar 2 menunjukkan contoh citra *dataset* beserta anotasi *bounding box* untuk masing-masing kelas, yang menggambarkan variasi ukuran objek, perbedaan kondisi pencahayaan, serta kompleksitas latar belakang dalam lingkungan pertanian.

Gambar 2 *Dataset* beserta anotasi *bounding box* (a) *Thrips*; (b) Kutu daun; (c) Kutu kebul; (d) Ulat

*Dataset* yang diperoleh dari Roboflow telah melalui proses pembagian (*split*) menjadi tiga subset: data *train*, data *valid*, dan data *test*. Pembagian ini bertujuan untuk mendukung proses pelatihan, penyesuaian model, dan pengujian kinerja secara terpisah. Rincian mengenai pembagian *dataset* dapat dilihat pada Tabel 2.

Tabel 2 Pembagian *dataset*

<i>Subset</i>	Persentase	Jumlah Citra
<i>Train</i>	70%	3496
<i>Valid</i>	20%	972
<i>Test</i>	10%	489
Total	100%	4957

#### d. *Preprocessing* Data

Tahap *preprocessing* data dilakukan untuk mempersiapkan *dataset* agar memenuhi kebutuhan model deteksi objek, serta untuk meningkatkan kualitas dan keragaman data *train*. Dalam penelitian ini, tidak dilakukan pembagian ulang *dataset* pada tahap *preprocessing*, karena *dataset* yang diperoleh dari Roboflow telah dibagi sebelumnya menjadi data *train*, data *valid*, dan data *test*.

Proses *preprocessing* difokuskan pada penyesuaian citra dan peningkatan kualitas visual agar memenuhi spesifikasi input model. Teknik yang diterapkan meliputi perubahan ukuran citra, penyesuaian kontras, serta peningkatan kualitas visual lainnya. Rincian teknik *preprocessing* yang digunakan disajikan pada Tabel 3.

Tabel 3 Teknik *preprocessing data*

<i>Preprocessing</i>	Keterangan
<i>Auto-Orient</i>	<i>Applied</i>
<i>Resize</i>	<i>Stretch to 640x640</i>
<i>Auto-Adjust Contrast</i>	<i>Using adaptive equalization</i>

Selain *preprocessing*, *augmentasi* data juga diterapkan pada *subset data train* untuk meningkatkan variasi citra dan mengurangi kemungkinan *overfitting*. *Augmentasi* ini digunakan pada kedua konfigurasi model (*baseline* dan optimasi *hyperparameter*) sehingga perbandingan dilakukan pada kondisi *dataset* yang sama. Selain itu, *augmentasi* berfungsi sebagai strategi mitigasi terhadap ketidakseimbangan (*class imbalance*) dengan memperkaya variasi sampel pada setiap kelas tanpa mengubah distribusi asli *dataset*. Setiap data *train* menghasilkan tiga variasi citra (*outputs per training example = 3*). Teknik *augmentasi* yang digunakan dirincikan dalam Tabel 4.

Tabel 4 Teknik *augmentasi data*

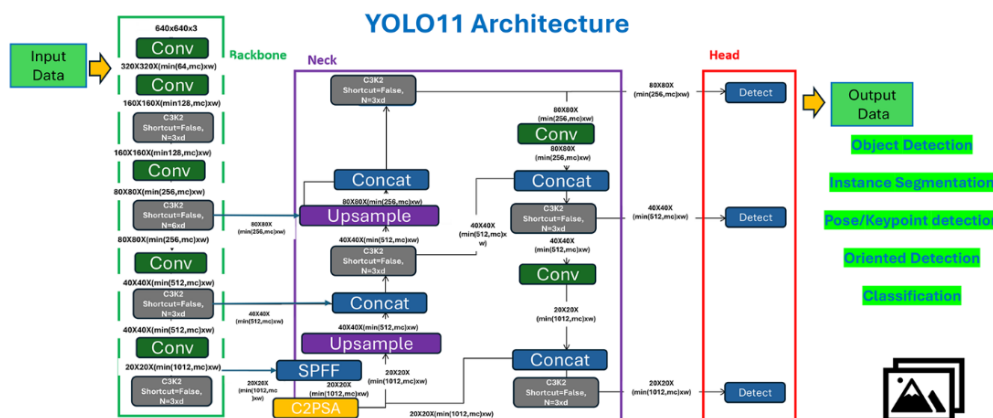
<i>Augmentasi</i>	Rentang/Keterangan
<i>Flip</i>	Horizontal
<i>Rotation</i>	-15° hingga +15°
<i>Saturation</i>	-25% hingga +25%
<i>Brightness</i>	-15% hingga +15%
<i>Noise</i>	Hingga 0.1% dari jumlah piksel

e. Arsitektur YOLO

*You Only Look Once* (YOLO) adalah algoritma deteksi objek satu tahap yang secara langsung memprediksi lokasi *bounding box* dan kelas objek dalam satu proses inferensi, sehingga dapat melakukan deteksi secara *real-time* dengan efisiensi komputasi yang tinggi (Ultralytics 2024). Pendekatan ini memformulasikan deteksi objek sebagai suatu permasalahan regresi yang mencakup prediksi lokasi objek, klasifikasi kelas, serta tingkat kepercayaan deteksi.

Secara umum, arsitektur YOLO tersusun atas tiga komponen utama, yaitu *backbone*, *neck*, dan *head*. *Backbone* berfungsi mengekstraksi fitur dari citra masukan, *neck* menggabungkan fitur multi-skala melalui operasi *upsampling* dan *concatenation* untuk meningkatkan kemampuan deteksi pada objek dengan ukuran berbeda, sedangkan *head* menghasilkan prediksi *bounding box*, kelas objek, dan skor kepercayaan. Struktur arsitektur YOLO yang digunakan dalam penelitian ini ditunjukkan pada Gambar 3.

Secara matematis, proses pelatihan YOLO bertujuan untuk mengoptimalkan fungsi kerugian (*loss function*) yang menggabungkan kesalahan prediksi lokasi objek, klasifikasi, dan kepercayaan deteksi. Fungsi kerugian tersebut dirumuskan pada Persamaan 1.



Gambar 3 Arsitektur YOLOv11 (Sapkota et al. 2025)

$$\mathcal{L} = \mathcal{L}_{box} + \mathcal{L}_{cls} + \mathcal{L}_{obj} \quad (1)$$

Model yang digunakan dalam penelitian ini adalah YOLOv11s. Varian ini dipilih karena memiliki keseimbangan antara akurasi dan efisiensi komputasi, sehingga sesuai untuk eksperimen perbandingan konfigurasi pelatihan. Proses pelatihan dilakukan menggunakan Google Colab dengan dukungan GPU.

#### f. Optimasi *Hyperparameter*

Tahap ini dilakukan untuk menganalisis pengaruh pengaturan *hyperparameter* terhadap kinerja model YOLOv11 dalam mendeteksi hama pada tanaman cabai. Proses ini menggunakan data *train* dan data *valid*, sementara data *test* hanya digunakan pada tahap evaluasi akhir.

Pada konfigurasi *baseline*, model dilatih selama 300 *epoch* dengan *batch size* 4. Optimizer yang digunakan adalah *default* (SGD), dengan *learning rate* awal mengikuti konfigurasi bawaan sistem. Konfigurasi ini digunakan sebagai acuan tanpa modifikasi *hyperparameter* utama. Sedangkan pada konfigurasi optimasi, *batch size* ditingkatkan menjadi 32. Optimizer diganti menjadi AdamW dengan *learning rate* awal 0.001, *learning rate factor* (lrf) 0.1, *momentum* 0.937, dan *weight decay* 0.0005. Optimizer yang diterapkan adalah AdamW, yang bertujuan untuk menjaga stabilitas dan keefektifan selama proses pelatihan. Seluruh proses pelatihan dilakukan menggunakan GPU.

Proses optimasi *hyperparameter* dalam penelitian ini dilakukan secara eksperimental melalui pendekatan *manual tuning* berdasarkan studi literatur dan referensi penelitian sebelumnya. Parameter yang disesuaikan meliputi ukuran citra (*imgsz*), *batch size*, jenis *optimizer*, *learning rate* awal (*lr0*), serta *learning rate factor* (*lrf*). Pemilihan nilai parameter dilakukan melalui tiga kali percobaan awal untuk memperoleh konfigurasi yang menunjukkan konvergensi stabil pada data *valid*. Pendekatan ini dipilih karena mempertimbangkan keterbatasan sumber daya komputasi dan fokus penelitian yang bertujuan membandingkan pengaruh konfigurasi pelatihan, bukan melakukan pencarian *hyperparameter* secara otomatis seperti *grid search* atau *Bayesian optimization*.

#### g. Pengujian Model dan Evaluasi

Pengujian model dilakukan untuk menilai kinerja model. Pada tahap ini, model diuji menggunakan data *test* yang tidak pernah digunakan dalam *train* dan *valid*, sehingga hasil yang diperoleh mencerminkan kemampuan generalisasi model secara objektif terhadap data baru. Evaluasi kinerja model deteksi objek dilakukan dengan menggunakan metrik *precision*, *recall*, dan *mean Average Precision* (mAP), yang sering diterapkan dalam penelitian deteksi objek berbasis *computer vision*. Metrik *precision* mengukur akurasi prediksi model, sedangkan *recall* menilai kemampuan model dalam mendeteksi semua objek target. Nilai *precision* dihitung menggunakan Persamaan 2, sedangkan *recall* dirumuskan pada Persamaan 3.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

*TP* adalah *True Positive*, *FP* adalah *False Positive*, dan *FN* adalah *False Negative* (Setiaji *et al.* 2025). Untuk memberikan penilaian kinerja model secara keseluruhan pada semua kelas objek, metrik *mean Average Precision* (mAP) digunakan. Nilai mAP dihitung sebagai rata-rata dari *Average Precision* (AP) untuk setiap kelas, yang diperoleh berdasarkan luas area di bawah kurva *precision-recall*. Perhitungan mAP dalam penelitian ini mengacu pada Persamaan 4.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (4)$$

*N* menunjukkan jumlah kelas objek yang dievaluasi (Sirisha *et al.* 2023).

## HASIL DAN PEMBAHASAN

Pelatihan model dilakukan dengan dua konfigurasi, YOLOv11 *baseline* dan YOLOv11 dengan optimasi *hyperparameter*. Kedua model ini menggunakan arsitektur dan *dataset* yang sama serta dilatih selama 300 *epoch*. Pendekatan ini bertujuan untuk memastikan bahwa perbedaan dalam performa yang dihasilkan murni disebabkan oleh strategi pengaturan *hyperparameter*, bukan oleh perbedaan dalam durasi pelatihan atau struktur jaringan.

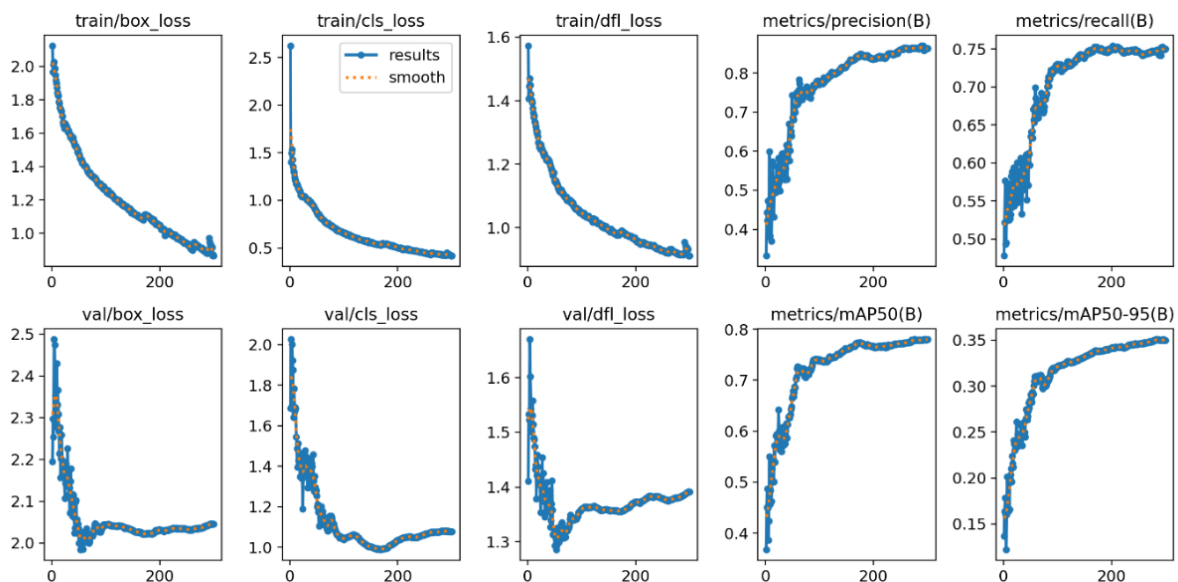
Evaluasi awal dilakukan pada data *train* dan data *valid* untuk mengamati perilaku konvergensi dan stabilitas selama proses pelatihan. Hasil perbandingan kinerja kedua model pada tahap ini dapat dilihat pada Tabel 5.

Tabel 5 Perbandingan kinerja YOLOv11 pada data *train* dan data *valid*

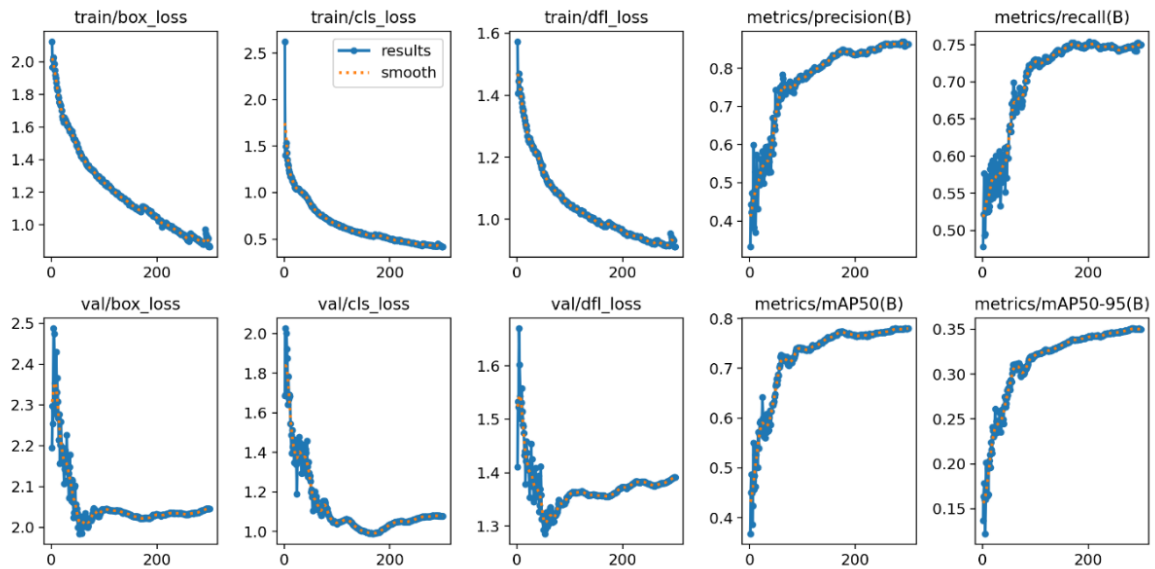
Nilai	<i>Baseline</i>	<i>Hyperparameter</i>
Precision	86%	85%
Recall	74%	78%
mAP@0.5	78.0%	80.2%
mAP@0.5–0.95	34%	37%

Berdasarkan Tabel 5, model dengan optimasi *hyperparameter* lebih efektif dalam mendeteksi lebih banyak objek hama secara akurat. Hal ini menunjukkan adanya perbedaan pola deteksi dibandingkan model *baseline*, meskipun perbedaan tersebut tidak sepenuhnya tercermin dalam metrik kuantitatif pada data *test*. Pada metrik *recall*, mAP@0.5, dan mAP@0.5–0.95 dibandingkan dengan model *baseline*. Kenaikan nilai *recall* menunjukkan bahwa model ini lebih efektif dalam mendeteksi lebih banyak objek hama secara akurat, terutama pada kelas objek yang berukuran kecil. Selain itu, peningkatan nilai mAP@0.5–0.95 mencerminkan perbaikan kualitas deteksi pada berbagai ambang *Intersection over Union* (IoU), yang menunjukkan kemampuan lokalisasi *bounding box* yang lebih akurat.

Selanjutnya, analisis dilakukan untuk memahami peningkatan kinerja tersebut terhadap perilaku pelatihan dan pola konvergensi model melalui kurva. Visualisasi perubahan nilai *loss* dan *mean Average Precision* (mAP) selama proses pelatihan dapat dilihat pada Gambar 4 dan Gambar 5.



Gambar 4 Kurva pelatihan YOLOv11 *baseline*



Gambar 5 Kurva pelatihan dengan optimasi *hyperparameter*

Gambar 4 menampilkan hasil pelatihan model YOLOv11 *baseline* yang dilatih selama 300 *epoch*. Kurva *training loss*, yang mencakup *box loss*, *classification loss*, dan distribusi *focal loss* menunjukkan penurunan yang konsisten sejak awal pelatihan, dengan laju penurunan bertahap hingga mendekati kondisi stabil awal akhir *epoch*. Pola serupa terlihat pada kurva *validation loss*, di mana nilai *loss* berkurang secara bertahap sebelum akhirnya mendatar dengan perubahan kecil. Hal ini menunjukkan bahwa proses pembelajaran berlangsung stabil dan model dapat menyesuaikan parameternya tanpa menunjukkan indikasi *overfitting* yang jelas. Dalam metrik evaluasi, nilai *precision* dan *recall* meningkat seiring bertambahnya *epoch*, diikuti oleh peningkatan mAP@0.5 dan mAP@0.5–0.95 hingga mencapai titik optimal. Model terbaik pada konfigurasi ini tercapai pada *epoch* ke-276, saat nilai mAP pada data validasi mencapai puncaknya sebelum menunjukkan kecenderungan stagnasi, menandakan bahwa performa optimal model tercapai mendekati akhir pelatihan.

Sementara itu, Gambar 5 menunjukkan hasil pelatihan model optimasi *hyperparameter* yang juga dilatih selama 300 *epoch*. Kurva *training loss* dan *validation loss* menunjukkan tren penurunan yang stabil dengan tingkat perubahan relatif lebih kecil selama pelatihan, mencerminkan proses optimasi parameter yang lebih konsisten. Jarak antara kurva *training* dan *validation loss* yang tetap terjaga menunjukkan kemampuan generalisasi model yang baik. Dalam metrik evaluasi, nilai *precision*, *recall*, mAP@0.5, dan mAP@0.5–0.95 meningkat secara bertahap hingga mencapai maksimum di tahap pertengahan pelatihan, sebelum cenderung mendatar dengan variasi kecil hingga akhir *epoch*. Model terbaik dalam konfigurasi ini diperoleh pada *epoch* ke-111, saat nilai mAP validasi mencapai titik tertinggi sebelum mengalami perubahan ringan pada *epoch-epoch* berikutnya. Hasil ini menunjukkan bahwa meskipun pelatihan berlanjut hingga 300 *epoch*, performa optimal model telah tercapai lebih awal dan dapat dipertahankan secara stabil hingga akhir pelatihan.

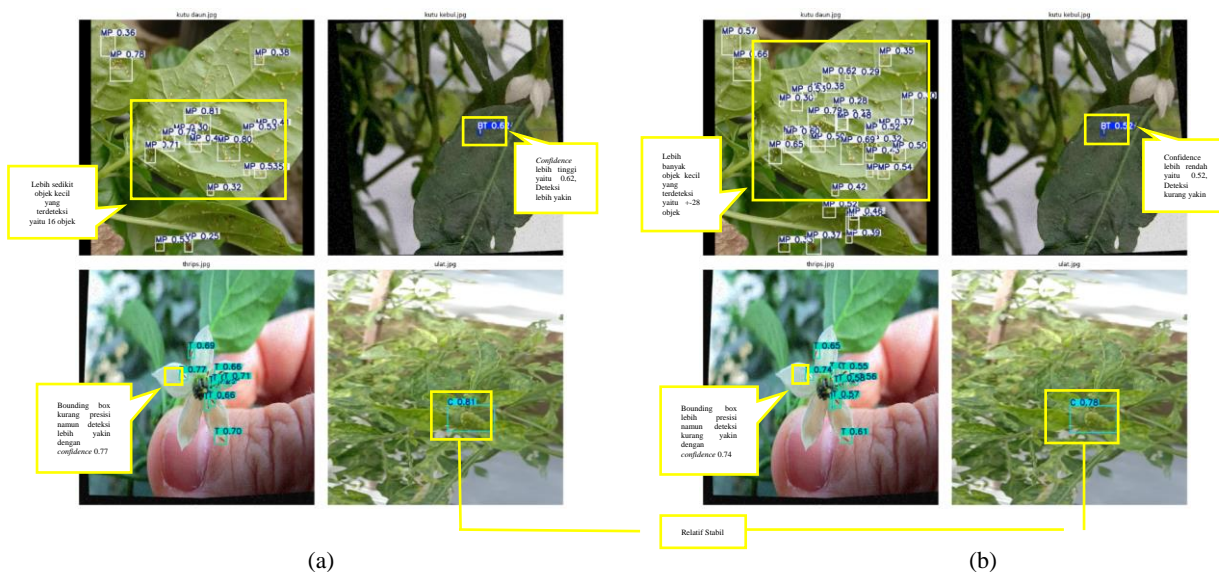
Setelah pelatihan, kedua model dievaluasi menggunakan data *test* untuk melihat performa pada data yang belum pernah dilihat sebelumnya. Hasil pengujian ditampilkan pada Tabel 6.

Tabel 6 Perbandingan kinerja YOLOv11 pada data *test*

Nilai	<i>Baseline</i>	<i>Hyperparameter</i>
<i>Precision</i>	89%	85%
<i>Recall</i>	81%	82%
mAP@0.5	83%	81%
mAP@0.5–0.95	38%	38%

Berdasarkan Tabel 6, model *baseline* menunjukkan nilai *precision* sebesar 89%, lebih tinggi dibandingkan model dengan optimasi *hyperparameter* yang memperoleh 85%. Hal ini menunjukkan bahwa model *baseline* lebih baik dalam meminimalkan prediksi positif yang salah (*false positive*). Pada metrik *recall*, model optimasi *hyperparameter* memperoleh nilai 82%, sedikit lebih tinggi dibandingkan *baseline* sebesar 81%. Namun, selisih 1% tersebut tergolong kecil dan belum menunjukkan perbedaan yang signifikan. Untuk *mAP@0.5*, model *baseline* kembali lebih unggul dengan nilai 83%, sedangkan model optimasi memperoleh 81%. Sementara itu, pada metrik *mAP@0.5–95*, kedua model menunjukkan nilai yang sama, yaitu 38%, yang mengindikasikan kemampuan deteksi pada berbagai ambang *IoU* relatif setara.

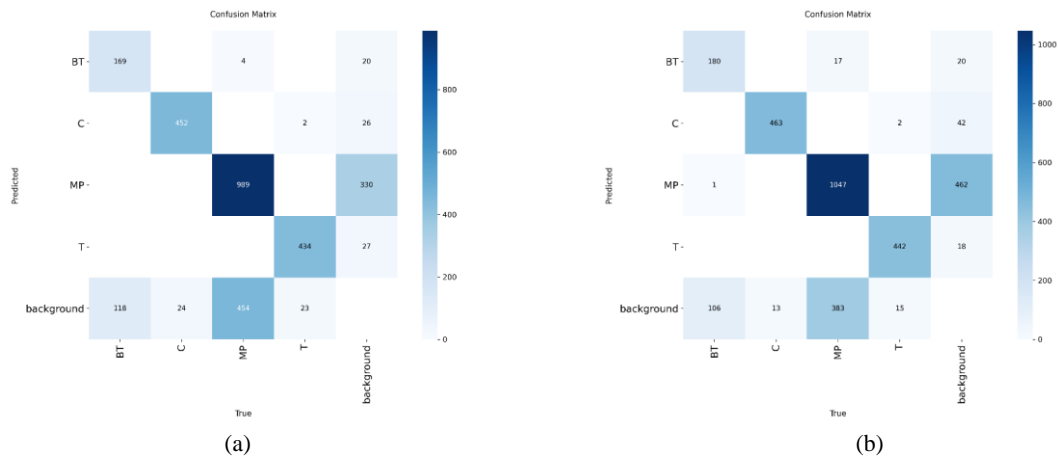
Hasil pada data *test* menunjukkan bahwa tidak terdapat peningkatan signifikan akibat optimasi *hyperparameter*. Model *baseline* cenderung lebih baik pada *precision* dan *mAP@0.5*, sedangkan model optimasi hanya menunjukkan peningkatan tipis pada *recall*. Dengan demikian, performa generalisasi kedua model dapat dikatakan relatif sebanding. Sebagai gambaran kualitatif mengenai perbedaan kinerja tersebut, dilakukan visualisasi hasil deteksi pada beberapa contoh citra. Hasil deteksi dari kedua model ditampilkan pada Gambar 6.



Gambar 6 Hasil deteksi YOLOv11 (a) *Baseline*; (b) *Hyperparameter*

Visualisasi hasil deteksi YOLOv11 *baseline* pada Gambar 6(a) menunjukkan bahwa model mampu mendeteksi objek sasaran, namun masih terdapat beberapa keterbatasan. Pada beberapa citra dengan kepadatan objek tinggi atau objek berukuran kecil, deteksi yang dihasilkan kurang konsisten. *Confidence score* yang dihasilkan juga menunjukkan variasi yang cukup besar, serta pada beberapa kasus masih ditemukan indikasi *missed detection*, khususnya pada objek kecil atau yang saling berdekatan.

Sebaliknya, pada Gambar 6(b), menunjukkan pola deteksi yang berbeda dibandingkan *baseline*, khususnya dalam jumlah objek kecil yang terdeteksi. Namun, perbedaan tersebut tidak sepenuhnya tercermin dalam peningkatan metrik kuantitatif pada data *test*. *Bounding box* yang dihasilkan tampak lebih konsisten dalam mencakup objek, dengan *confidence score* yang relatif lebih stabil. Selain itu, jumlah objek yang terdeteksi pada beberapa citra meningkat, termasuk objek berukuran kecil dan pada latar belakang yang lebih kompleks. Hal ini menunjukkan peningkatan kemampuan model dalam melakukan diskriminasi dan generalisasi dibandingkan model *baseline*. Sebagai evaluasi konsistensi klasifikasi antar kelas, analisis dilakukan menggunakan *confusion matrix*, sebagaimana ditunjukkan pada Gambar 7.



Gambar 7 Confusion matrix YOLOv11 (a) Baseline; (b) Hyperparameter

Analisis *confusion matrix* yang ditampilkan pada Gambar 7 menunjukkan bahwa kesalahan klasifikasi paling sering terjadi pada kelas MP-*background*, di mana objek MP sering tertukar dengan latar belakang. Hal ini dipengaruhi oleh ukuran objek MP yang terlalu kecil serta kemiripan warna dan tekstur dengan daun dalam citra lapangan, sehingga pemisahan objek dari latar belakang menjadi sulit.

Pada Gambar 7(a), tingkat kesalahan klasifikasi pada kelas MP tergolong tinggi, ditandai oleh jumlah prediksi MP yang salah sebagai *background* dan sebaliknya. Selain itu, masih ada kesalahan prediksi pada kelas T yang tertukar dengan *background*, meskipun dalam jumlah yang lebih sedikit. Ini menunjukkan bahwa model belum sepenuhnya berhasil mengekstraksi fitur pembeda yang kuat antara objek hama kecil dan latar belakang yang kompleks.

Sebaliknya, pada Gambar 7(b), terdapat peningkatan secara numerik pada nilai diagonal *confusion matrix*, khususnya untuk kelas BT, MP, dan T, yang menunjukkan peningkatan jumlah prediksi yang benar. Jumlah kesalahan MP-*background* menurun dibandingkan model *baseline*, mengindikasikan bahwa pengaturan *hyperparameter* yang lebih optimal dapat meningkatkan kemampuan model dalam membedakan objek hama kecil dan latar belakang.

## SIMPULAN

Penelitian ini menunjukkan bahwa optimasi *hyperparameter* memengaruhi dinamika pelatihan model YOLOv11 dan memberikan peningkatan numerik pada data *valid*, khususnya pada metrik *recall* dan  $mAP@0.5-0.95$ . Namun, pada evaluasi data *test*, performa kedua konfigurasi relatif sebanding. Model *baseline* menunjukkan nilai *precision* dan  $mAP@0.5$  yang lebih tinggi, sedangkan model optimasi hanya menunjukkan peningkatan tipis pada *recall*. Temuan ini mengindikasikan bahwa optimasi *hyperparameter* dalam skenario ini tidak secara konsisten meningkatkan kemampuan generalisasi model, tetapi berperan dalam membentuk stabilitas proses pelatihan. Hasil perbandingan model menunjukkan bahwa konfigurasi optimasi *hyperparameter* memberikan performa yang kompetitif dan cenderung lebih baik pada tahap pelatihan dan validasi, meskipun pada evaluasi akhir performanya relatif sebanding dengan konfigurasi *baseline*. Temuan ini mengindikasikan bahwa pengaturan *hyperparameter* berperan dalam memengaruhi dinamika pelatihan model, namun tidak selalu menghasilkan peningkatan signifikan pada seluruh metrik evaluasi.

Untuk pengembangan lebih lanjut, penelitian ini masih memiliki ruang perluasan, khususnya penggunaan *dataset* dengan variasi kondisi lapangan yang lebih beragam, seperti perbedaan fase pertumbuhan tanaman, intensitas pencahayaan, dan kepadatan hama. Selain itu, eksplorasi metode optimasi *hyperparameter* yang lebih adaptif atau otomatis, serta pengujian dalam skenario implementasi nyata di lapangan, berpotensi meningkatkan ketahanan dan kesiapan sistem untuk diterapkan secara operasional.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada seluruh tim penelitian atas kerja sama, dedikasi, dan kontribusi yang telah diberikan selama pelaksanaan penelitian ini, sehingga penelitian dapat diselesaikan dengan baik.

## DAFTAR PUSTAKA

- Agustian I, Faurina R, Ihsani S, Putra F, Dinata K, Daratha N. 2023. Deep learning pest detection on Indonesian red chili pepper plant based on fine-tuned YOLOv5. *Int J Adv Intell Informatics*. 9(3):383–401. doi:10.26555/ijain.v9i3.864.
- BPS. 2024. Produksi Tanaman Sayuran dan Buah–Buahan Semusim Menurut Jenis Tanaman. <https://www.bps.go.id/id/statistics-table/3/VFV4MmQxaG9kakZrVUdWeEx6aDFUMnN6WmpocVp6MDkjMw%3D%3D/produksi-tanaman-sayuran-dan-buahbuahan-semusim-menurut-jenis-tanaman?year=2024>.
- Dong Q, Sun L, Han T, Cai M, Gao C. 2024. PestLite: a novel YOLO-based deep learning technique for crop pest detection. *Agric*. 14(2). doi:10.3390/agriculture14020228.
- Ekawati ED, Wiharti T, Nugroho AA. 2023. Identification of pests and diseases in red chili plants (*Capsicum Annum L.*) in nangsrri, manisrenggo, klaten. *J Biol Trop*. 23(4):507–512. doi:10.29303/jbt.v23i4.5650.
- Irjang, linnaninengseh, Aulia MR. 2024. Analisa pertumbuhan dan produksi tanaman cabai merah (*Capsicum annum L.*) dengan aplikasi berbagai jenis insektisida nabati. *J Agroterpadu*. 3(2):143. doi:10.35329/ja.v3i2.5674.
- Isa IGT, Junedi B. 2022. Hyperparameter tuning epoch dalam meningkatkan akurasi data latih dan data validasi pada citra pengendara. *Pros Semin Nas Sains dan Teknol*. 12(1):231–237. doi:10.36499/psnst.v12i1.6697.
- Jamin FS, Auliani DMKR, Rusli M, Pramono SA. 2024. Penggunaan pestisida dalam pertanian: resiko kesehatan dan alternatif ramah lingkungan. *J Kolaboratif Sains*. 7(11):4151–4159. doi:10.56338/jks.v7i11.6342.
- Kristiana D, Mukson M, Santoso SI. 2025. Increasing the income of chili farmers by implementation of internal company institution. *J Tek Pertan Lampung (Journal Agric Eng*. 14(1):319. doi:10.23960/jtep-l.v14i1.319-329.
- Lokku G, Reddy GH, Prasad MNG. 2022. OPFaceNet: OPTimized Face Recognition Network for noise and occlusion affected face images using Hyperparameters tuned Convolutional Neural Network. *Appl Soft Comput*. 117:1–21. doi:10.1016/j.asoc.2021.108365.
- Neshat M, Ahmed M, Askari H, Mirjalili MTS. 2024. Hybrid inception architecture with residual connection: fine-tuned inception-resnet deep learning model for lung inflammation diagnosis from chest radiographs. *Procedia Comput Sci*. 235:1841–1850. doi:10.1016/j.procs.2024.04.175.
- Riva LS, Jayanta. 2023. Deteksi penyakit tanaman cabai menggunakan algoritma YOLOv5 dengan variasi pembagian data. *J Inform J Pengemb IT*. 8(3):248–254.
- Rohmat C lukman, Suarna N, Rahaningsing N, Sriyulia E. 2020. Sistem pakar identifikasi hama pada tanaman cabai dalam upaya meningkatkan kualitas cabai. *Inf Syst J*. 3(1):39–30. doi:10.32627/internal.v3i1.99.
- Santoso AF, Dewi C, Santoso E. 2025. Deteksi dan klasifikasi hama pada tanaman menggunakan yolov8. *J Pengemb Teknol Inf dan Ilmu Komput*. 9(8):2548–964. <http://j-ptiik.ub.ac.id>.
- Sapkota R, Flores-Calero M, Qureshi R, Badgujar C, Nepal U, Poulouse A, Zeno P, Vaddevolu UBP, Khan S, Shoman M, et al. 2025. YOLO advances to its genesis: a decadal and comprehensive review of the You Only Look Once (YOLO) series. *Artif Intell Rev*. 58(9). doi:10.1007/s10462-025-11253-3.

- Setiaji P, Triyanto WA, Nurhaliza M. 2025. Real-time traffic density and anomaly monitoring using YOLOv8, openCV and pattern recognition for smart city applications in demak. *J Tek Inform.* 6(4):1769–1782. doi:10.52436/1.jutif.2025.6.4.4867.
- Singh V, Chug A, Singh AP. 2023. Classification of beans leaf diseases using fine tuned CNN model. *Procedia Comput Sci.* 218:348–356. doi:10.1016/j.procs.2023.01.017.
- Sirisha U, Praveen SP, Srinivasu PN, Barsocchi P, Bhoi AK. 2023. Statistical analysis of design aspects of various YOLO-based deep learning models for object detection. *Int J Comput Intell Syst.* 16(1):1–29. doi:10.1007/s44196-023-00302-w.
- Ultralytics. 2024. YOLOv11 models documentation. <https://docs.ultralytics.com/models/yolo11/>., siap terbit.
- Utomo PD, Sulistyono A, Widiwurjani. 2024. Pengaruh konsentrasi dan selang waktu pemberian pupuk AB mix terhadap hasil panen buah cabai merah (*Capsicum annuum* L.). *Plumula Berk Ilm Agroteknologi.* 12(1):25–31. doi:10.33005/plumula.v12i1.137.
- Yasen NM, Rifka S, Vitria R, Yulindon. 2023. Pemanfaatan yolo untuk deteksi hama dan penyakit pada daun cabai menggunakan metode deep learning. *Elektron J Ilm.* 15(2):63–71. doi:10.30630/eji.0.0.397.
- Zulkarnain IA, Kusri. 2025. Yolov11 optimization through hyperparameter tuning and data augmentation to improve vehicle detection accuracy at night. *MALCOM Indones J Mach Learn Comput Sci.* 5(4):1294–1303. doi:10.57152/malcom.v5i4.2250.