

## Evaluasi Kinerja YOLOv11 pada Deteksi Penyakit Tanaman Cabai: Studi Komparatif dengan YOLOv8, YOLOv5, dan SSD

Jelita Permatasari\*, Edmund Ucok Armin, Egi Sunardi, Maria Bestarina Laili,  
Salsanabila Mariestiara Putri

Fakultas Teknik, Program Studi Teknik Elektro, Universitas Singaperbangsa Karawang

\*E-mail: jelita.permatasari@ft.unsika.ac.id

### Abstract

#### Article history:

Received: 18-11-2025

Accepted: 08-12-2025

Published: 30-12-2025

#### Keywords:

anthracnose;  
chili disease detection;  
fruit fly;  
object detection;  
YOLOv11.

Early and accurate detection of chili plant diseases is essential to support precision agriculture and minimize crop losses. Conventional visual inspection performed by farmers is often subjective and inconsistent, particularly under varying lighting conditions and complex field environments. Recent developments in deep learning, especially object detection models, enable the automation of disease identification with higher reliability. This study evaluates the performance of the YOLOv11 architecture for detecting three classes related to chili plant conditions—anthracnose, fruit fly, and healthy fruit—using a primary dataset of 1,062 field images collected in Karawang, Indonesia. The model was trained using a standardized configuration and compared with three widely used object detection models: YOLOv8, YOLOv5, and SSD. The training process was conducted for 100 epochs, with evaluation metrics including precision, recall, mAP50, mAP50–95, and inference time. Experimental results show that YOLOv11 achieved the highest detection performance, with an mAP50 of 86.94%, outperforming YOLOv8 by 3.8%, YOLOv5 by 6.8%, and SSD by 12.7%. The model also demonstrated the fastest inference speed at 10.9 ms, making it suitable for real-time field applications. Training analysis indicated stable convergence at the 61st epoch, supported by balanced precision (0.82391) and recall (0.77967) values as well as consistent reductions in both training and validation losses. These findings demonstrate that YOLOv11 provides more accurate and efficient detection of chili plant diseases compared with previous YOLO variants and SSD, and it offers strong potential for implementation in practical agricultural environments.

### 1. Pendahuluan

Tanaman cabai (*Capsicum spp.*) merupakan komoditas hortikultura strategis yang memiliki nilai ekonomi tinggi dan dibudidayakan secara luas di berbagai wilayah Indonesia[1]. Namun, produktivitasnya sangat dipengaruhi oleh penyakit tanaman seperti antraknosa, bercak daun, *leaf curl*, dan serangan hama seperti lalat buah yang menyebabkan penurunan kualitas dan kuantitas hasil panen[2, 3]. Deteksi penyakit secara konvensional masih mengandalkan observasi visual oleh petani, yang bersifat subjektif, tidak konsisten, dan mudah dipengaruhi oleh kondisi lingkungan seperti intensitas cahaya, latar belakang daun, serta tingkat kematangan buah[4]. Hal ini mendorong perlunya sistem deteksi penyakit yang lebih akurat, stabil, dan dapat bekerja secara real-time.

Perkembangan visi komputer dan *deep learning* telah membuka peluang bagi otomatisasi deteksi penyakit tanaman melalui citra digital. Berbagai pendekatan CNN seperti *GSAAtt-CMNetV3*[5], *SE-CNN*[6], *EfficientNet*[7], *VGG16* dan *VGG-EffAttnNet*[8], serta *Pep-*

*VGGNet*[9] telah menunjukkan performa tinggi pada identifikasi penyakit daun cabai dan paprika. Meskipun model berbasis klasifikasi tersebut unggul dalam akurasi, metode tersebut tidak dapat memberikan informasi spasial mengenai lokasi penyakit, sehingga tidak dapat digunakan untuk pemetaan tingkat keparahan penyakit di lapangan.

Untuk mengatasi keterbatasan tersebut, algoritma deteksi objek satu tahap seperti YOLO (*You Only Look Once*) telah digunakan secara luas pada aplikasi pertanian presisi. YOLOv5 telah terbukti efektif dalam deteksi hama dan penyakit cabai di Indonesia[10], sedangkan YOLOv8 menunjukkan peningkatan signifikan dalam stabilitas dan akurasi deteksi penyakit cabai maupun paprika pada berbagai kondisi cahaya dan skala[11, 12]. Penelitian lain juga menunjukkan bahwa model YOLO mampu mengungguli pendekatan dua tahap dalam hal waktu inferensi, menjadikannya cocok untuk aplikasi lapangan di perangkat berdaya rendah[13, 14]. Selain YOLO, metode SSD (*Single Shot Multibox Detector*) juga digunakan karena kecepatannya, meskipun performanya pada

objek berukuran kecil seperti bercak penyakit cenderung lebih rendah[15].

Dalam beberapa tahun terakhir, penggunaan model-model *deep learning* untuk deteksi penyakit tanaman semakin meluas, termasuk pada tanaman kapas, gandum, anggur, kentang, tomat, dan padi dengan tingkat keberhasilan yang bervariasi[16-19]. Namun, mayoritas penelitian masih menggunakan dataset publik yang tidak selalu mencerminkan kondisi nyata di lapangan, sehingga akurasi menurun ketika diterapkan pada citra dengan pencahayaan natural atau latar belakang kompleks.

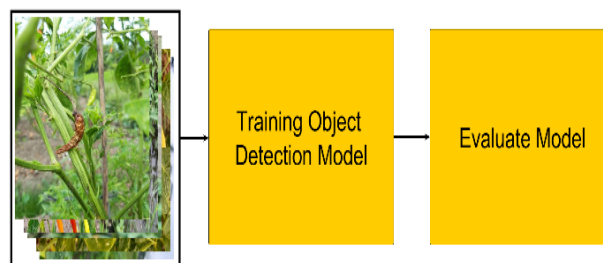
Sementara itu, YOLOv11 merupakan generasi terbaru keluarga YOLO, yang hadir dengan pembaruan pada arsitektur *backbone*, mekanisme *feature pyramid*, dan modul perhatian (*attention*) sehingga berpotensi menghasilkan deteksi lebih akurat pada objek kecil, termasuk gejala penyakit pada daun dan buah cabai. Hingga kini, belum ada penelitian yang mengevaluasi kinerja YOLOv11 dalam mendeteksi penyakit tanaman cabai menggunakan dataset primer dari kondisi lapangan Indonesia.

Oleh karena itu, penelitian ini bertujuan untuk mengevaluasi kinerja YOLOv11 pada deteksi penyakit tanaman cabai berdasarkan dataset primer dari Karawang, Indonesia, dan membandingkan performanya dengan YOLOv8, YOLOv5, dan SSD menggunakan *metrik precision, recall, mAP50, mAP50-95, dan inference time*.

Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem deteksi penyakit tanaman berbasis visi komputer yang lebih cepat, akurat, dan efektif untuk mendukung pertanian presisi di Indonesia.

## 2. Metode

Metode yang digunakan pada penelitian ini adalah penerapan arsitektur YOLOv11 sebagai model utama untuk melakukan deteksi penyakit pada tanaman cabai, dengan beberapa penyesuaian pada proses pelatihan untuk memperoleh akurasi yang optimal. Alur sistem secara keseluruhan ditunjukkan pada Gambar 1, dimulai dari pengumpulan dataset primer, tahap anotasi, proses pelatihan, hingga evaluasi performa model.

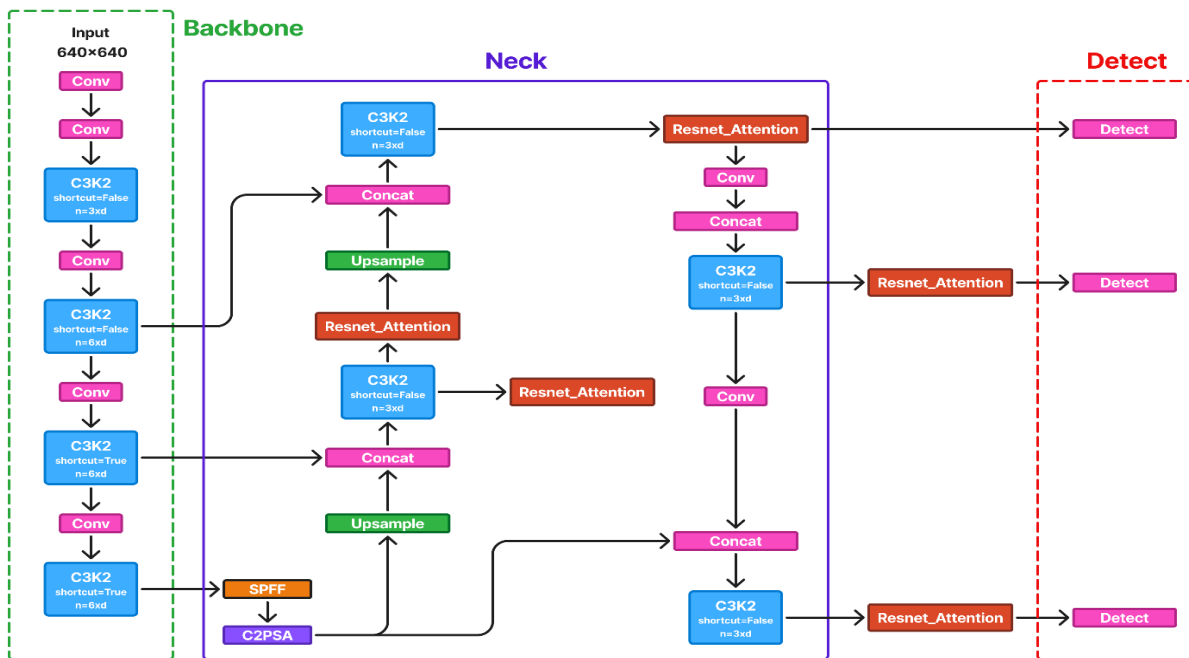


Gambar 1. Alur penelitian model deteksi penyakit tanaman cabai

Untuk memperoleh perbandingan yang komprehensif, YOLOv11 diuji bersama tiga arsitektur lain yang umum digunakan dalam penelitian deteksi objek, yaitu YOLOv8, YOLOv5, dan SSD. Ketiga model tersebut digunakan sebagai pembanding guna melihat sejauh mana peningkatan performa yang diberikan oleh YOLOv11 pada dataset yang digunakan. Dengan pendekatan ini, penelitian dapat menunjukkan kontribusi model yang diusulkan secara lebih objektif dalam konteks deteksi penyakit tanaman cabai.

### 2.1 Arsitektur YOLOv11

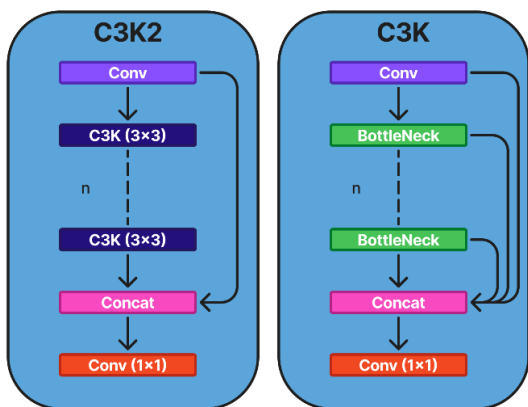
Arsitektur YOLOv11 ditunjukkan pada Gambar 2, yang menggambarkan alur pemrosesan mulai dari *backbone, neck*, hingga *head* prediksi. YOLOv11 merupakan pengembangan signifikan dari generasi sebelumnya, dirancang untuk meningkatkan akurasi deteksi sekaligus mempertahankan kecepatan inferensi *real-time*. Model ini mengintegrasikan tiga komponen utama, yaitu blok C3K2, modul SPFF, dan blok C2PSA, yang bersama-sama membentuk alur ekstraksi dan penguatan fitur yang lebih efisien. Pada bagian *head*, YOLOv11 tetap mempertahankan prediksi multi-skala dari *feature map* P3, P4, dan P5 sehingga mampu melakukan deteksi pada objek kecil hingga objek berukuran besar secara simultan. Kombinasi antara *backbone* yang ringan, *neck* yang diperkuat, dan mekanisme prediksi multi-skala inilah yang membuat YOLOv11 stabil, cepat, dan unggul dalam mendeteksi objek pada berbagai kondisi



Gambar 2. Arsitektur YOLOv11

**2.2 Blok C3K2**

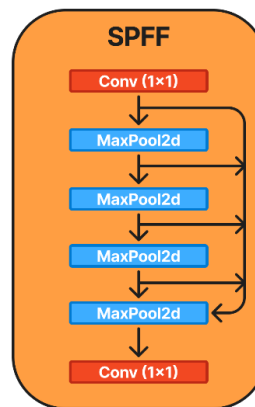
Struktur internal dari blok C3K2 ditunjukkan pada Gambar 3. Blok ini dirancang untuk menggantikan arsitektur CSP standar dengan pendekatan yang lebih efisien. C3K2 memanfaatkan konvolusi kernel 3x3 secara konsisten, memproses peta fitur melalui dua jalur terpisah, kemudian menyatukannya kembali pada tahap akhir untuk menghasilkan representasi fitur yang lebih kaya. Pembagian jalur ini memungkinkan pengurangan parameter komputasi tanpa mengorbankan kualitas ekstraksi fitur. Dengan desain tersebut, C3K2 memberikan keseimbangan yang baik antara kecepatan pemrosesan dan kedalaman informasi visual, menjadikannya fondasi penting dalam performa YOLOv11 pada tahap backbone.



Gambar 3. C3K2 blocks

**2.3 Modul Spatial Pyramid Pooling Fast (SPFF)**

Struktur *Spatial Pyramid Pooling Fast* (SPFF) dan *upsampling* diperlihatkan pada Gambar 4.



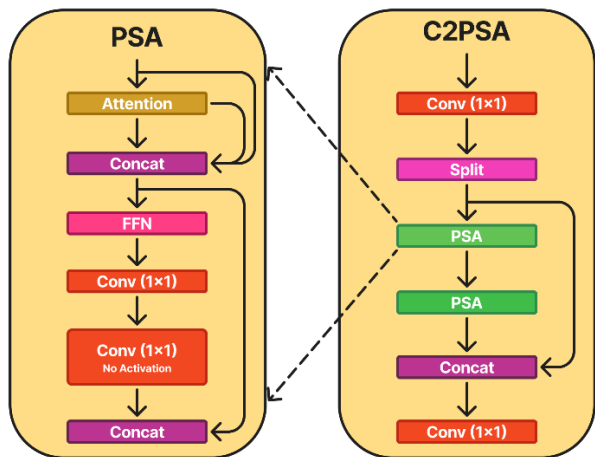
Gambar 4. *Spatial Pyramid Pooling Fast* (SPFF) and *upsampling*

Gambar 4 menunjukkan struktur modul SPFF yang berada pada bagian *neck* arsitektur YOLOv11. Modul ini bertugas mengumpulkan informasi kontekstual dari berbagai skala menggunakan mekanisme *multi-kernel pooling*. Dengan menggabungkan beberapa ukuran kernel dalam satu jalur ekstraksi, SPFF mampu menangkap pola visual dari objek kecil, objek yang samar, dan objek dengan variasi ukuran yang signifikan. Pendekatan ini memperkaya

fitur tanpa menambah beban komputasi yang besar, sehingga tetap mempertahankan kecepatan inferensi. SPFF menjadi salah satu komponen utama yang membuat YOLOv11 lebih akurat dalam kondisi gambar yang tidak ideal, seperti *occlusion* atau pencahayaan tidak merata.

### 2.4 Blok Cross Stage Partial Spatial Attention (C2PSA)

Struktur detail dari blok C2PSA ditampilkan pada Gambar 5.



Gambar 5. C2-Position Sensitive Attention Block (C2PSA)

Blok ini memperkenalkan mekanisme partial spatial attention, yaitu strategi untuk membuat model lebih fokus pada area penting di dalam gambar. Peta fitur dibagi menjadi dua cabang pemrosesan, masing-masing melewati modul PSA yang memperkuat informasi spasial terkait posisi, kontur, dan batas objek. Setelah diproses secara terpisah, kedua cabang ini digabungkan kembali untuk menghasilkan representasi fitur yang lebih selektif. Mekanisme ini memungkinkan YOLOv11 mendeteksi objek yang kecil, kompleks, atau sebagian tertutup dengan lebih akurat dibandingkan arsitektur YOLO generasi sebelumnya.

### 3. Hasil dan Pembahasan

Dalam penelitian ini, proses pelatihan seluruh model dilakukan menggunakan *framework PyTorch* pada bahasa pemrograman *python*. Seluruh eksperimen dijalankan pada sistem dengan spesifikasi Windows 11, NVIDIA RTX 3060 dengan prosesor *Intel Core i5* generasi ke-11, serta RAM 32 GB. Spesifikasi perangkat tersebut memastikan bahwa YOLOv11 sebagai model utama, maupun YOLOv8, YOLOv5, dan SSD sebagai model pembanding, dapat dilatih

menggunakan konfigurasi parameter yang seragam sehingga hasil evaluasi performa dapat diperoleh secara objektif dan setara.

### 3.1 Dataset

Penelitian ini menggunakan dataset primer yang dikumpulkan langsung dari kebun cabai di Desa Segaran, Kecamatan Batujaya, Karawang. Total terdapat 1.062 gambar yang merepresentasikan kondisi tanaman cabai di lapangan, termasuk variasi tingkat keparahan penyakit serta kondisi visual daun dan buah. Dataset ini terdiri dari tiga kelas, yaitu antraknosa dengan 934 anotasi, cabai normal dengan 3.353 anotasi, dan lalat buah dengan 388 anotasi yang dapat dilihat pada Gambar 6.

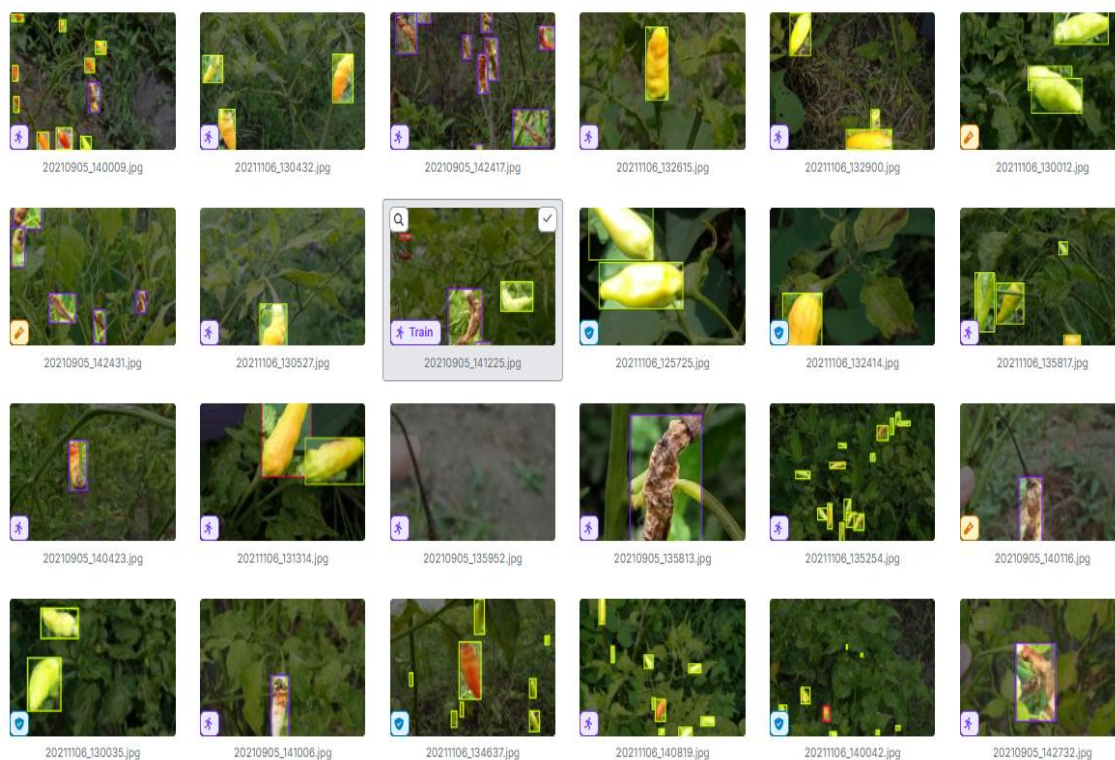
Seluruh data kemudian dibagi menjadi dua bagian, yaitu data pelatihan untuk proses training model dan data pengujian untuk mengevaluasi performa deteksi. Rincian jumlah data per kelas dapat dilihat pada Tabel 1.

Tabel 1. Distribusi kelas penyakit tanaman cabai

Class	train	valid	test
Normal	2370	643	340
Antraknosa	670	170	94
Lalat Buah	261	78	49

Tabel 1 menunjukkan distribusi jumlah sampel per kelas pada tiga subset dataset, yaitu data pelatihan (*train*), data validasi (*valid*), dan data pengujian (*test*). Terdapat tiga kelas yang digunakan dalam penelitian ini, yaitu normal, antraknosa, dan lalat buah. Kelas normal memiliki jumlah sampel terbesar dengan 2370 data train, 643 data valid, dan 340 data test, sedangkan kelas antraknosa memiliki 670 data train, 170 data valid, dan 94 data test. Kelas dengan jumlah sampel paling sedikit adalah lalat buah, yaitu 261 data train, 78 data valid, dan 49 data *test*.

Ketidakseimbangan jumlah sampel antar kelas mencerminkan kondisi nyata di lapangan, di mana cabai normal lebih sering ditemukan dibandingkan tanaman yang terserang penyakit. Perbedaan jumlah ini juga menimbulkan tantangan dalam proses pelatihan karena model berpotensi lebih mudah mengenali kelas mayoritas. Untuk mengurangi dampak ketidakseimbangan tersebut, penelitian ini menerapkan augmentasi data serta mekanisme loss adaptif selama proses pelatihan sehingga model tetap dapat belajar secara optimal pada setiap kelas.



Gambar 6. Dataset training penyakit cabe

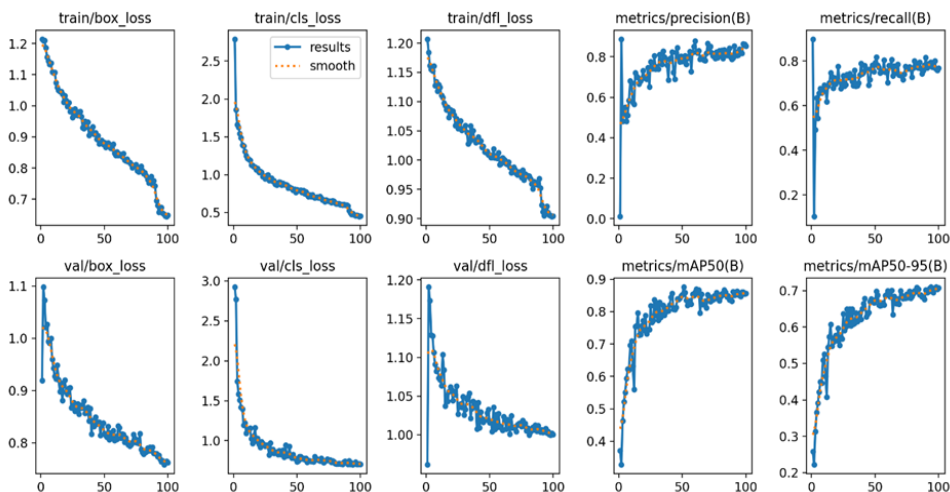
### 3.2 Pelatihan Model YOLOv11

Pada tahap pelatihan, seluruh citra pada dataset diproses menggunakan format dan resolusi standar yang sesuai dengan arsitektur model YOLO yaitu 640px, kemudian dibagi menjadi tiga subset dengan proporsi 70% untuk data pelatihan, 20% untuk data validasi dan 10% untuk data testing. Proses training dilakukan menggunakan framework *PyTorch*, dengan pengaturan umum berupa jumlah epoch sebanyak 100, learning rate awal sebesar 0.001, dan *batch size* sebanyak 16 citra per *inference*. pada Gambar 7.

Model YOLOv11 sebagai metode yang diusulkan menunjukkan peningkatan mAP dan penurunan loss yang cukup baik. Berdasarkan analisis menyeluruh terhadap seluruh parameter pelatihan, iterasi ke-61 teridentifikasi sebagai iterasi dengan performa paling seimbang dan stabil dibandingkan iterasi lainnya. Pada iterasi ini, model mencapai nilai

mAP50 sebesar 0.86938 dan mAP50-95 sebesar 0.69395, yang menunjukkan kemampuan deteksi yang kuat pada berbagai tingkat IoU. Nilai *precision* sebesar 0.82391 dan *recall* sebesar 0.77967 memperlihatkan bahwa model mampu menjaga keseimbangan antara ketepatan identifikasi objek dan sensitivitas dalam mendeteksi seluruh kejadian penyakit pada citra cabai.

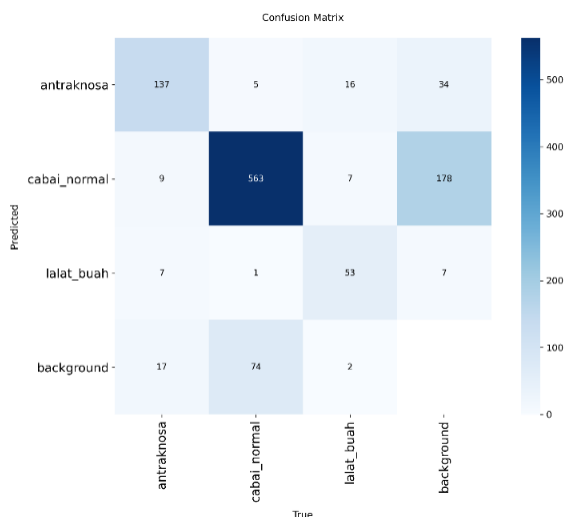
Selain itu, nilai *training loss* dan *validation loss* yang relatif berdekatan, dengan *train box loss* sebesar 0.84153, dan *train class loss* sebesar 0.71454 menunjukkan bahwa model tidak mengalami gejala *overfitting*, sehingga generalisasinya tetap terjaga. Secara keseluruhan, kombinasi nilai mAP yang tinggi, stabilitas metrik validasi, serta konsistensi penurunan loss menjadikan iterasi 61 sebagai titik optimal dari keseluruhan proses pelatihan YOLOv11 dalam mendeteksi penyakit tanaman cabai.



Gambar 7. Hasil training YOLOv11

### 3.3 Evaluasi Performa

Evaluasi performa dilakukan menggunakan data pengujian untuk menilai kemampuan masing-masing model dalam mengenali tiga kelas penyakit pada tanaman cabai. Dari hasil pengujian, YOLOv11 memperoleh akurasi tertinggi pada metrik mAP50, yakni mencapai 86,94%. *Confusion Matrix* dapat dilihat pada Gambar 8. Dan hasil visualisasi objek deteksi dapat dilihat pada Gambar 9.



Gambar 8. Confusion matrix

Gambar 8 menampilkan *confusion matrix* dari model YOLOv11 pada data pengujian, yang menggambarkan tingkat keberhasilan model dalam mengklasifikasikan masing-masing kelas: Normal, Antraknosa, dan Lalat Buah *confusion matrix* ini menunjukkan bahwa model berhasil mengenali kelas Normal dengan tingkat akurasi yang sangat baik, tercermin dari jumlah prediksi benar (*true positives*) yang tinggi. Pada kelas

Antraknosa, model juga menunjukkan performa kuat meskipun beberapa prediksi salah mengarah pada kelas Normal, mengingat kemiripan visual antara bercak penyakit dan pola alami permukaan cabai. Kelas Lalat Buah, yang memiliki jumlah sampel paling sedikit, juga dapat dikenali oleh model, meskipun tingkat kesalahannya sedikit lebih tinggi dibandingkan kelas lainnya.

Secara keseluruhan, Gambar 8 menggambarkan bahwa YOLOv11 memiliki kemampuan klasifikasi yang stabil pada ketiga kelas, dengan kesalahan prediksi yang relatif rendah, sejalan dengan nilai *precision* dan *recall* tinggi yang ditunjukkan pada Tabel 2.

Gambar 9 merupakan visualisasi hasil deteksi YOLOv11 terhadap objek penyakit cabai pada data uji. Gambar ini menunjukkan bagaimana model memberikan bounding box dan label prediksi pada setiap objek, termasuk identifikasi cabai normal, gejala antraknosa maupun keberadaan lalat buah.

Visualisasi ini membuktikan bahwa YOLOv11 mampu mendeteksi objek dalam kondisi lapangan yang kompleks, seperti variasi pencahayaan, latar belakang daun yang rapat, ukuran objek kecil seperti lalat buah, dan bercak penyakit yang tidak selalu memiliki pola seragam. YOLOv11 mampu memberikan prediksi yang presisi tinggi pada objek kecil dan area penyakit yang samar (*subtle lesions*). Hal ini sejalan dengan mekanisme perhatian (*attention*) dan pemrosesan multi-skala pada arsitektur YOLOv11.

Jika dibandingkan dengan YOLOv8, nilai ini lebih tinggi sekitar 3,8%, dan selisihnya meningkat menjadi 6,8% ketika dibandingkan

dengan YOLOv5. Kesenjangan terbesar terlihat pada perbandingan dengan SSD, di mana YOLOv11 unggul hingga 12,7%. Hasil analisa ini mengindikasikan bahwa YOLOv11 memberikan hasil prediksi yang lebih akurat dalam mendeteksi gejala penyakit cabai dibandingkan seluruh model pembanding dengan menggunakan dataset yang peneliti gunakan. Rangkuman lengkap dari seluruh parameter evaluasi untuk keempat model disajikan pada Tabel 2.

Tabel 2. Komparasi performa metode deteksi penyakit tanaman cabai

Model	Precision (%)	Recall (%)	mAP50 (%)	Speed (ms)
SSD	70,12	65,49	74,21	8
YOLOv5	76,85	70,21	80,12	13
YOLOv8	79,21	74,12	83,14	11,5
YOLOv11	82,39	77,97	86,94	10,9

Tabel 2 menyajikan perbandingan performa empat arsitektur deteksi objek yaitu SSD, YOLOv5, YOLOv8, dan YOLOv11 berdasarkan empat metrik evaluasi, yaitu

Model YOLOv11 menunjukkan performa terbaik pada seluruh metrik yang diukur.

Model SSD memperoleh nilai precision 70,12%, recall 65,49%, dan mAP50 74,21%, dengan kecepatan inferensi tercepat yaitu 8 ms. Meskipun SSD cepat, akurasi paling rendah di antara semua metode. YOLOv5 menunjukkan peningkatan dengan precision 76,85% dan mAP50 80,12%, tetapi kecepatan inferensinya menurun menjadi 13 ms. YOLOv8 memiliki performa yang lebih baik dibandingkan YOLOv5 dengan precision 79,21%, recall 74,12%, dan mAP50 83,14%, serta kecepatan inferensi 11,5 ms.

Model terbaik adalah YOLOv11, yang mencapai precision 82,39%, recall 77,97%, dan mAP50 86,94%. Selain itu, YOLOv11 juga mempertahankan kecepatan inferensi yang cepat, yaitu 10,9 ms, sehingga paling sesuai untuk implementasi *real-time*. Secara keseluruhan, Tabel 2 menegaskan bahwa YOLOv11 memberikan kinerja terbaik di antara semua model pembanding pada dataset penelitian ini.



precision, recall, mAP50, dan inference speed.

Gambar 9. Object detection implementation

#### 4. Kesimpulan

Penelitian ini mengusulkan penggunaan model YOLOv11 untuk meningkatkan akurasi deteksi penyakit tanaman cabai dibandingkan arsitektur deteksi objek yang telah banyak digunakan seperti YOLOv8, YOLOv5, dan SSD. Berdasarkan hasil pengujian, YOLOv11 menunjukkan performa terbaik dengan menggunakan dataset yang kami miliki, khususnya pada metrik mAP50 yang mencapai 86,94%, unggul 3,8% dari YOLOv8, 6,8% dari YOLOv5, dan 12,7% dari SSD. Selain akurasi yang lebih tinggi, YOLOv11 juga menghasilkan waktu inferensi tercepat, yaitu 10,9 ms, sehingga lebih efisien ketika diterapkan pada aplikasi deteksi penyakit secara real-time di lapangan. Selama proses pelatihan, YOLOv11 menunjukkan penurunan *loss* yang stabil hingga mendekati konvergensi, serta mempertahankan keseimbangan antara *precision* dan *recall*. Secara keseluruhan, hasil penelitian ini membuktikan bahwa YOLOv11 merupakan model yang lebih akurat, efisien, dan paling sesuai digunakan untuk mendeteksi gejala penyakit antraknosa dan lalat buah dibandingkan model pembanding lainnya dengan dataset yang peneliti miliki.

#### Daftar Pustaka

- [1] Hasbollah, A. F. K., Zin, Z. M., Ibrahim, N., & Suleiman, R. F. R., 2020. *Green chilli leaf disease detection using convolution neural networks*. J. Green Eng, Vol. 10, pp. 13005-13019.
- [2] Sari, Y., Baskara, A. R., & Wahyuni, R., Year. *Classification of chili leaf disease using the gray level co-occurrence matrix (glcm) and the support vector machine (svm) methods*. in *2021 Sixth International Conference on Informatics and Computing (ICIC)*: IEEE.
- [3] Lebrini, Y. & Ayerdi Gotor, A., 2024. *Crops disease detection, from leaves to field: What we can expect from artificial intelligence*. Agronomy, Vol. 14, No. 11, p. 2719.
- [4] Begum, S. S. A. & Syed, H., 2024. *Gsatt-cmnetv3: Pepper leaf disease classification using osprey optimization*. IEEE Access, Vol. 12, pp. 32493-32506.
- [5] Naik, B. N., Malmathanraj, R., & Palanisamy, P., 2022. *Detection and classification of chilli leaf disease using a squeeze-and-excitation-based cnn model*. Ecological Informatics, Vol. 69, p. 101663.
- [6] Rani, R., Bharany, S., Elkamchouchi, D. H., Ur Rehman, A., Singh, R., & Hussien, S., 2025. *Vgg-effattnnet: Hybrid deep learning model for automated chili plant disease classification using vgg16 and efficientnetb0 with attention mechanism*. Food Science & Nutrition, Vol. 13, No. 7, p. e70653.
- [7] Çetinkaya, S. & Tandirovic Gursel, A., 2025. *Pep-vggnet: A novel transfer learning method for pepper leaf disease diagnosis*. Applied Sciences, Vol. 15, No. 15, p. 8690.
- [8] Odounfa, M. G. F., Hounmenou, C. G., Salako, V. K., Affokpon, A., & Kakai, R. L. G., 2025. *Deep learning enables precision agriculture for sustainable chili pepper disease detection in benin*. Discover Artificial Intelligence, Vol. 5, No. 1, p. 315.
- [9] Agustian, I., Faurina, R., Ishak, S. I., Utama, F. P., Dinata, K., & Daratha, N., 2023. *Deep learning pest detection on indonesian red chili pepper plant based on fine-tuned yolov5*. International Journal of Advances in Intelligent Informatics, Vol. 9, No. 3, pp. 383-401.
- [10] Ma, N., Wu, Y., Bo, Y., & Yan, H., 2024. *Chili pepper object detection method based on improved yolov8n*. Plants, Vol. 13, No. 17, p. 2402.
- [11] Moya, V., Quito, A., Pilco, A., Vásconez, J. P., & Vargas, C., 2024. *Crop detection and maturity classification using a yolov5-based image analysis*. Emerg. Sci. J., Vol. 8, No. 2, pp. 496-512.
- [12] Anjanadevi, B., Charmila, I., Ns, A., & Anusha, R., 2020. *An improved deep learning model for plant disease detection*. International Journal of Recent Technology and Engineering, Vol. 8, No. 6, pp. 5389-5392.
- [13] Sadi, A. A., Hossain, Z., Ahmed, A. U., & Shad, M. T. M., Year. *A comparative study on plant diseases using object detection models*. in *Science and Information Conference*: Springer.
- [14] Zayani, H. M. et al., 2024. *Deep learning for tomato disease detection with yolov8*. Engineering, Technology & Applied Science Research, Vol. 14, No. 2, pp. 13584-13591.
- [15] Wagh, T. A., Samant, R., Gujarathi, S. V., & Gaikwad, S. B., 2019. *Grapes leaf disease detection using convolutional neural network*. Int. J. Comput. Appl, Vol. 178, No. 20, pp. 7-11.

- [16] Xu, L. *et al.*, 2023. *Wheat leaf disease identification based on deep learning algorithms*. *Physiological and Molecular Plant Pathology*, Vol. 123, p. 101940.
- [17] Muthulakshmi, M., Aishwarya, N., Vinesh Kumar, R. K., & Rakesh Thoppaen Suresh, B., 2024. *Potato leaf disease detection and classification with weighted ensembling of yolov8 variants*. *Journal of Phytopathology*, Vol. 172, No. 6, p. e13433.
- [18] Arya, S., Sandhu, K. S., Singh, J., & Kumar, S., 2022. *Deep learning: As the new frontier in high-throughput plant phenotyping*. *Euphytica*, Vol. 218, No. 4, p. 47.
- [19] Saharan, M.& Singh, G., Year. *Leaf disease detection using transfer learning*. in *International Conference on Artificial Intelligence: Towards Sustainable Intelligence*: Springer.