E-ISSN : 2988-1986
https://ejournal.cahayailmubangsa.institute/index.php/kohesi

Kohesi: Jurnal Multidisiplin Saintek
Volume 10 No 6 Tahun 2025

Implementasi Sistem Multimedia untuk Identifikasi Tanaman Nilam Berbasis Computer Vision Menggunakan YOLOv8

Rivaldo Paath[1], Daniel Azarya[2], Jacob Rahayaan[3], Andre Porayou[4], Ade Yusupa[5], Sherwin Reinaldo U Aldo Sompie[6], Wahyuni Fithratul Zalmi[7], Brave A Sugiarso[8]

[1,2,3,4,5,6,7,8]Teknik Informatika, Fakultas Teknik, Universitas Sam Ratulangi
[1]rivaldopaath026@student.unsrat.ac.id, [2]danielazarya026@student.unsrat.ac.id,
[3]jacorahayaan026@student.unsrat.ac.id, [4]andreporayou026@student.unsrat.ac.id, [5]ade@unsrat.ac.id,
[6]aldo@unsrat.ac.id, [7]wahyuni.fithratul.zalmi@unsrat.ac.id, [8]brave@unsrat.ac.id

Abstrak

Penelitian ini mengembangkan sistem multimedia berbasis computer vision untuk mengidentifikasi tanaman nilam (Pogostemon cablin). Sistem ini memanfaatkan teknologi pemrosesan citra digital, akuisisi visual, dan integrasi data multimedia untuk menyediakan metode pemantauan tanaman yang lebih akurat dibandingkan metode manual. Algoritma YOLOv8 (You Only Look Once versi 8) digunakan karena kemampuannya dalam mendeteksi objek secara cepat dan tepat pada beragam kondisi pencahayaan, sehingga sesuai dengan kebutuhan aplikasi multimedia yang bersifat interaktif dan real-time.

Tahapan penelitian meliputi pengumpulan dataset citra, praproses gambar, serta pelatihan model YOLOv8 yang dibandingkan dengan Convolutional Neural Network (CNN). Hasilnya menunjukkan bahwa YOLOv8 memiliki performa lebih unggul dalam hal akurasi dan kecepatan deteksi, sehingga sangat efektif untuk diterapkan dalam sistem multimedia cerdas yang mendukung pengambilan keputusan berbasis visual. Implementasi sistem ini diharapkan dapat membantu pengelolaan tanaman nilam secara lebih efisien, meningkatkan produktivitas, dan mendorong pemanfaatan teknologi AI di sektor pertanian.

Kata kunci: sistem multimedia; computer vision; tanaman nilam; deteksi real-time; YOLOv8.

Article History
Received: November 2025
Reviewed:November 2025
Published: November 2025

Plagiarism Checker No 235
Prefix DOI :
10.8734/Kohesi.v1i2.365
Copyright : Author
Publish by : Kohesi

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License

Abstract

This study develops a multimedia system based on computer vision to identify patchouli plants (Pogostemon cablin). The system utilizes digital image processing, visual acquisition, and multimedia data integration to provide a more accurate plant monitoring method compared to conventional manual techniques. The YOLOv8 (You Only Look Once version 8) algorithm is employed due to its ability to detect objects rapidly and accurately under various lighting conditions, making it suitable for interactive and real-time multimedia applications. The research stages include collecting image datasets, preprocessing images, and training the YOLOv8 model, which is compared with a Convolutional Neural Network (CNN). The results indicate that YOLOv8 outperforms CNN in both detection accuracy and processing speed, making it highly effective for intelligent multimedia systems that support visual-based decision-making. The implementation of this system is expected to enhance the efficiency of patchouli plant management, improve productivity, and promote the adoption of AI technology in the agricultural sector.

Keywords: *multimedia systems; computer vision; patchouli plant; real-time detection; YOLOv8.*

1. PENDAHULUAN

Tanaman nilam (Pogostemon cablin) merupakan tanaman penghasil minyak atsiri yang memiliki nilai ekonomi serta penting bagi berbagai sektor, termasuk industri parfum dan pengobatan tradisional. Perkembangan teknologi computer vision, khususnya dengan model YOLOv8, telah membawa kemajuan signifikan dalam mendeteksi penyakit tanaman secara cepat dan akurat, yang sangat dibutuhkan untuk mendukung pertanian presisi [1]. Selain itu, penerapan YOLO juga telah berhasil digunakan untuk mendeteksi dan mengklasifikasikan hama pada tanaman kentang, menunjukkan bahwa metode ini memiliki fleksibilitas tinggi dalam menangani berbagai permasalahan agronomi [2].

Dalam bidang klasifikasi penyakit tanaman, penggunaan model deep learning seperti *ResNet101* telah terbukti efektif dalam mengidentifikasi penyakit pada tanaman padi berdasarkan citra daun, yang membuka peluang besar dalam pengembangan sistem diagnosis otomatis [3]. Algoritma CNN Xception juga menjadi pilihan populer dalam klasifikasi penyakit pada tanaman bawang, berkat kemampuannya menangkap fitur-fitur halus pada citra daun yang terdampak penyakit [4]. Selain itu, metode Convolutional Neural Network (CNN) juga telah digunakan untuk mengidentifikasi berbagai jenis tanaman obat Indonesia berdasarkan citra daunnya, menunjukkan potensi besar kecerdasan buatan dalam konservasi dan pemanfaatan tanaman herbal [5].

Kondisi fisiologis tanaman nilam sangat dipengaruhi oleh berbagai faktor, baik internal maupun eksternal. Salah satu aspek penting adalah keberadaan bakteri rizosfer yang berpotensi menghasilkan senyawa antibakteri, yang dapat membantu meningkatkan kesehatan tanaman nilam serta melindunginya dari serangan patogen [6]. Dalam konteks pertanian modern, klasifikasi tingkat kematangan tanaman pakcoy hidroponik menggunakan metode Support Vector Machine (SVM) menjadi contoh bagaimana teknologi machine learning dapat meningkatkan efisiensi produksi tanaman melalui pemantauan yang lebih akurat [7].

Faktor lingkungan juga sangat menentukan pertumbuhan tanaman nilam, seperti dibuktikan oleh penelitian mengenai pengaruh konsentrasi dan lama perendaman kolkisin terhadap pertumbuhan varietas nilam MV1 dari Lhokseumawe, yang menjadi dasar dalam optimalisasi teknik budidaya [8]. Selain itu, kesuburan tanah juga berperan penting dalam mendukung pertumbuhan tanaman nilam, di mana analisis tingkat kesuburan tanah memberikan gambaran mengenai kondisi lahan yang mendukung atau menghambat perkembangan tanaman [9]. Penelitian lain yang sejalan menunjukkan bahwa perlakuan perendaman kolkisin memberikan pengaruh signifikan terhadap pertumbuhan tanaman nilam, menegaskan pentingnya pengelolaan teknik budidaya yang tepat [10].

Dalam upaya meningkatkan produktivitas, metode perbanyakan tanaman nilam menggunakan sistem fotoautotrof dengan variasi konsentrasi gula dan jumlah ventilasi telah diterapkan, sehingga memberikan alternatif teknologi budidaya yang lebih efisien dan ramah lingkungan [11]. Dengan berkembangnya teknologi pengolahan citra dan machine learning, diharapkan sistem identifikasi dan pemantauan tanaman nilam dapat diintegrasikan untuk mendukung pengambilan keputusan yang lebih cepat dan lebih tepat dalam budidaya tanaman.

2. METODOLOGI PENELITIAN

2.1 *Hardware Specification*

Spesifikasi perangkat keras mempengaruhi kinerja dan kemampuan suatu sistem komputer. Komponen seperti prosesor, RAM, dan penyimpanan berdampak pada kecepatan pemrosesan data, kapasitas untuk menjalankan aplikasi, serta kemampuan dalam menangani

tugas-tugas yang kompleks. Keberhasilan sebuah program atau aplikasi juga dapat dipengaruhi oleh kesesuaian antara spesifikasi perangkat keras yang digunakan dengan kebutuhan program yang dirancang. Adapun spesifikasi perangkat keras yang digunakan dalam penelitian ini adalah:

System Model	Lenovo Legion 5 16IRX9
Operating System	Windows 11 64-bit
BIOS	NMCN30WW
Processor	Intel Core i7-14650HX, 16C (8P + 8E) / 24T, P-core 2.2 / 5.2GHz, E-core 1.6 / 3.7GHz, 30MB
Memory	16GB SO-DIMM DDR5-5600 MT/s

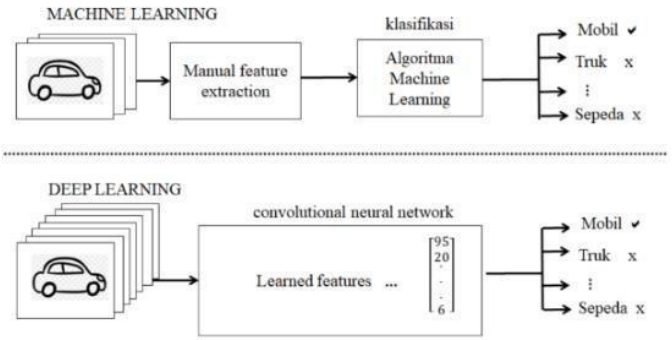
Tabel 1. Hardware Specifications

2.2 *Computer Vision*

Computer vision adalah bidang ilmu komputer yang berfokus pada pengembangan algoritma dan teknologi yang memungkinkan komputer memahami informasi visual dari dunia fisik. Dalam konteks computer vision, informasi visual dapat berasal dari berbagai sumber, seperti gambar, video, citra medis, data sensor optik, dan lainnya [12]. Computer vision bertujuan meniru cara kerja sistem visual manusia. Visual manusia sangat kompleks, di mana indera penglihatan digunakan untuk melihat objek, kemudian citra yang diterima dikirim ke otak untuk diinterpretasikan sehingga manusia dapat memahami apa yang dilihat.

2.3 *Deep Learning*

Deep Learning adalah salah satu sub bidang dalam Machine Learning yang berfokus pada penggunaan jaringan saraf yang dalam (deep neural networks) dalam memproses data [13]. Secara umum, deep learning memiliki struktur jaringan yang sama dengan neural network, yaitu terdiri dari input layer, hidden layer, dan output layer. Deep learning memiliki kaidah umum bahwa jumlah lapisannya lebih banyak dibandingkan neural network, sehingga disebut deep. Salah satu kemampuan deep learning adalah menggantikan fungsi manual dengan algoritma yang lebih efisien untuk Hierarchical Unsupervised Learning, Semi-Supervised Feature Learning, dan Hierarchical Feature Extraction [14]. Perbedaan antara Machine Learning dan Deep Learning dapat dilihat pada Gambar 1 berikut.

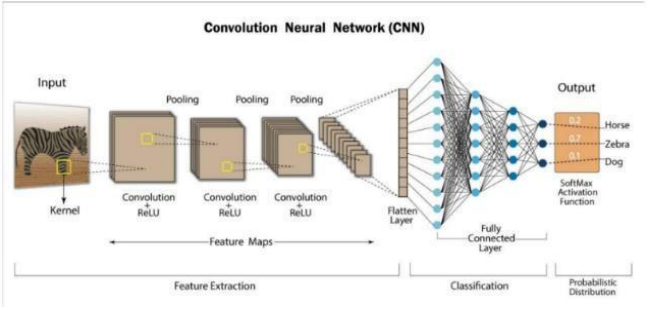


Gambar 1. Machine Learning vs Deep Learning

2.4 *Convolutional Neural Network*

CNN merupakan salah satu metode deep learning yang dikembangkan dari Multilayer Perceptron dan dirancang untuk memproses data dalam bentuk dua dimensi, seperti gambar, video, dan suara [15]. Teknik CNN terinspirasi dari cara manusia menghasilkan persepsi visual

dan umumnya digunakan pada data citra [16]. Secara prinsip, CNN merupakan pengembangan dari metode konvensional Artificial Neural Network (ANN). Pada ANN terdapat tiga hingga empat lapisan, sedangkan CNN dapat memiliki hingga ratusan lapisan. CNN memiliki kemampuan feature learning yang lebih baik dibandingkan metode lain. Pada Gambar 2 di bawah ini, dapat dilihat alur proses CNN dalam pengolahan citra.



Gambar 2. Arsitektur Model CNN

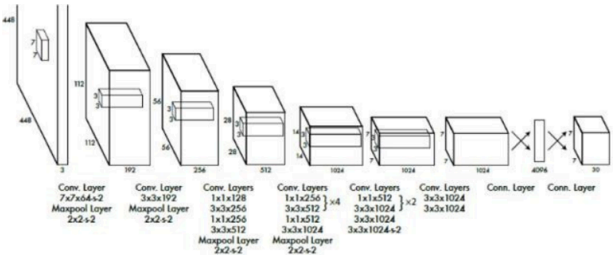
Menurut penelitian[17], terdapat beberapa parameter yang digunakan dalam algoritma CNN, di antaranya:

1. Epoch: jumlah iterasi proses pelatihan model pada data pelatihan. Semakin banyak epoch, semakin baik model mempelajari data dan menghasilkan prediksi yang lebih akurat.
2. Activation Function: fungsi yang mengubah output dari setiap neuron dalam jaringan saraf tiruan. Fungsi ini memungkinkan jaringan saraf mempelajari pola-pola kompleks pada data.
3. Loss Function: berfungsi untuk mengukur seberapa jauh prediksi model dari nilai sebenarnya. Tujuannya adalah meminimalkan besar kesalahan tersebut.
4. Optimizer: algoritma yang digunakan untuk menemukan nilai parameter model yang dapat meminimalkan loss function. Contoh optimizer meliputi SGD, Adam, dan RMSprop.

2.5 *You Only Look Once*

YOLO (You Only Look Once) adalah sebuah metode dalam bidang computer vision dan pengenalan objek yang digunakan untuk mendeteksi objek secara real-time pada gambar dan video [18]. Keunggulan utama YOLO adalah kemampuannya dalam mendeteksi objek dengan cepat dan akurat dalam satu kali proses. Pada model YOLO, sistem dapat memproses gambar hingga 45 FPS, bahkan dengan versi jaringan saraf yang lebih kecil, yaitu Fast YOLO, dapat memproses hingga 155 FPS dan menjadi salah satu algoritma tercepat dalam perbandingan algoritma deteksi real-time. Kecepatan ini jauh melampaui metode non-real-time seperti Fast R-CNN dan Faster R-CNN yang masing-masing hanya mampu menghasilkan sekitar 0,5 FPS dan 7 FPS [19].

YOLO memiliki struktur yang terdiri dari 24 lapisan konvolusi dan 4 lapisan max pooling, serta 2 lapisan fully connected. Pada beberapa lapisan konvolusi, YOLO menggunakan 1x1 reduction sebagai alternatif untuk mengurangi kedalaman feature map [18]. Arsitektur YOLO pertama kali diperkenalkan oleh Joseph Redmon, yang diilustrasikan pada Gambar 3.

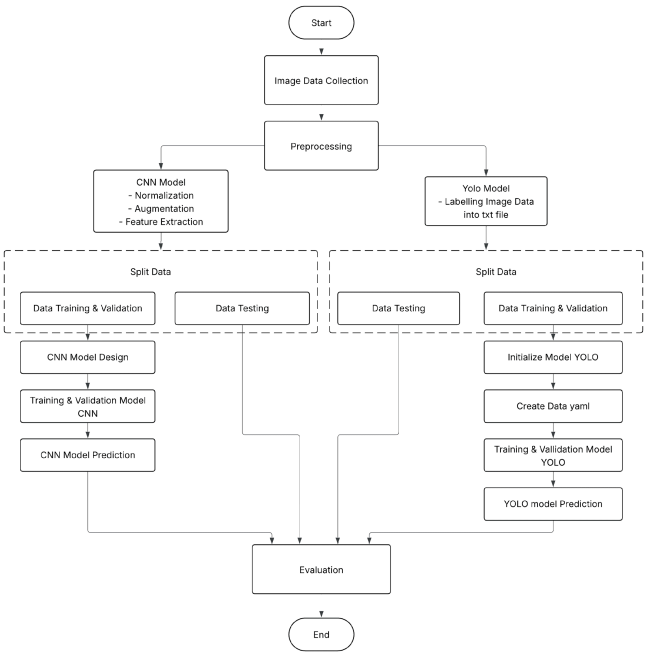


Gambar 3. Arsitektur Model Yolo

Menurut [20], algoritma YOLO v8 menggunakan beberapa parameter, yaitu:

1. Batch Size: jumlah data yang digunakan dalam satu iterasi pelatihan, sehingga model dapat diperbarui setelah melihat sebagian kecil data.
2. Filters: alat untuk memindai data dan mengekstraksi fitur tertentu. Semakin banyak filters, semakin banyak fitur yang dapat ditemukan, namun waktu pelatihan akan semakin lama.
3. Learning Rate: mengontrol besarnya perubahan bobot model pada setiap iterasi. Learning rate yang tinggi menyebabkan perubahan yang besar, sedangkan learning rate yang rendah menghasilkan perubahan kecil.
4. Max Batches: batas jumlah data yang digunakan untuk melatih model. Pelatihan akan berhenti ketika batas ini tercapai.
5. Confidence Threshold: menentukan tingkat keyakinan model terhadap prediksinya. Ambang batas yang tinggi menghasilkan prediksi yang lebih akurat, tetapi jumlah prediksi menjadi lebih sedikit.

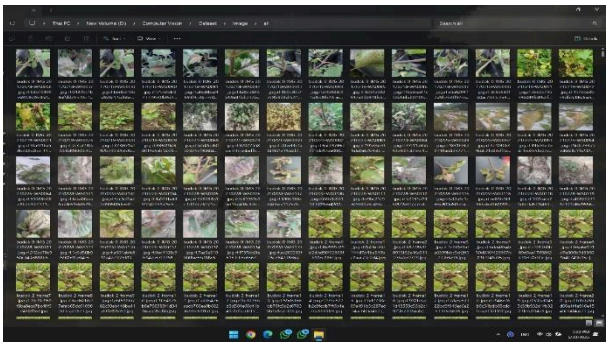
Metode penelitian yang digunakan adalah membandingkan dua algoritma dengan dataset yang sama. Alur metode tersebut dapat dilihat pada gambar berikut.



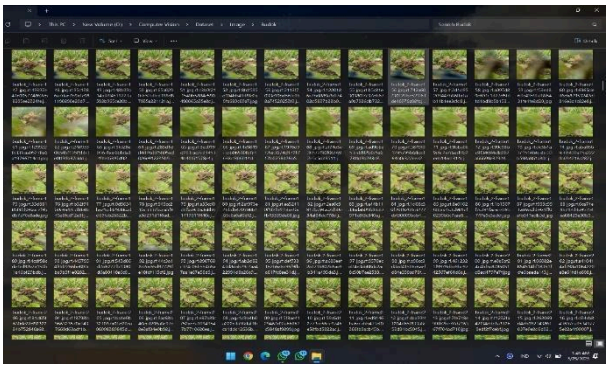
Gambar 4. Metode Penelitian

2.6 *Data Collection*

Pengumpulan dataset merupakan tahap awal dalam proses pengembangan model deteksi objek. Dataset tersebut diambil dari berbagai sumber, sudut pandang, dan dalam kondisi yang berbeda-beda. Gambar 5, 6, 7, dan 8 menunjukkan contoh pengumpulan dataset untuk setiap kategori.



Gambar 5. Dataset Budok



Gambar 6. Dataset Layu Bakteri

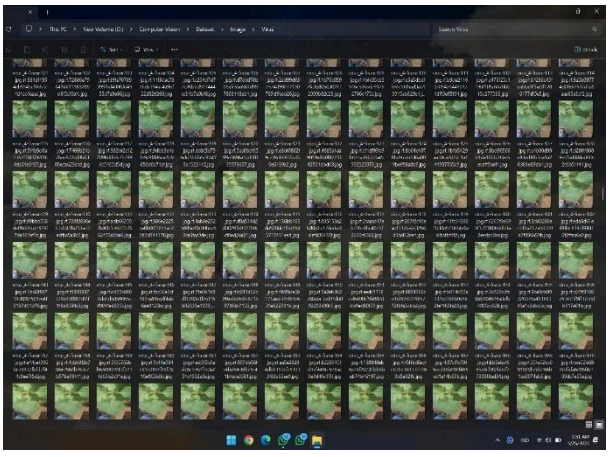
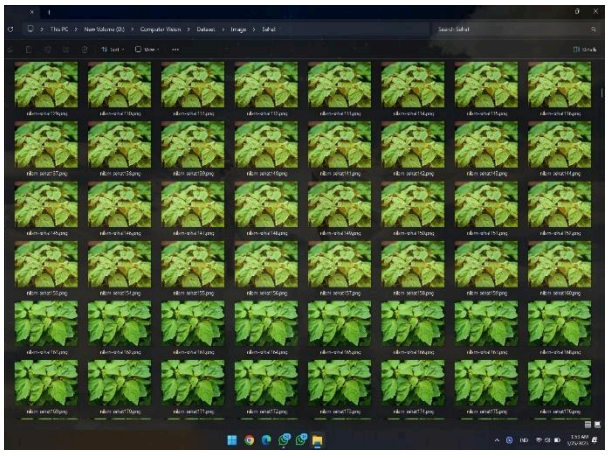


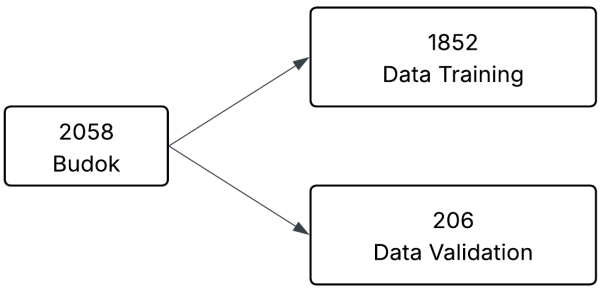
Figure 7. Dataset Virus



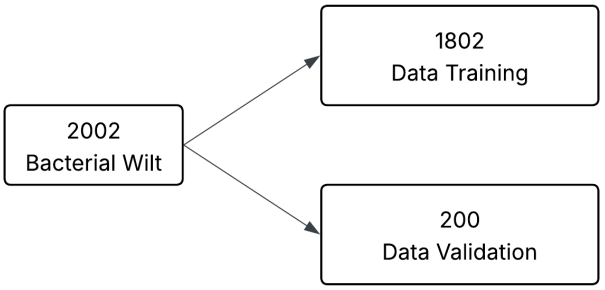
Gambar 8. Health Dataset

2.7 *Dataset division*

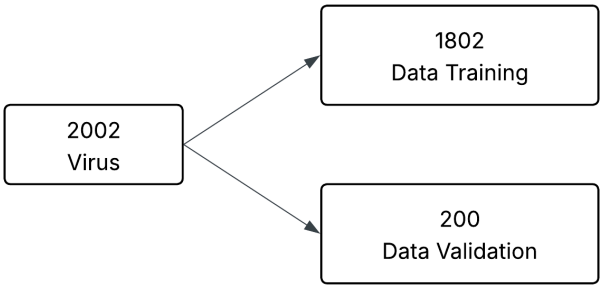
Pembagian dataset merupakan salah satu aspek penting untuk memastikan model dapat mempelajari data dengan baik saat diuji pada kondisi dunia nyata. Dataset dibagi menjadi tiga bagian, yaitu training, validation, dan testing. Dengan melakukan pembagian ini, model dapat mempelajari variasi data dan menghasilkan prediksi yang lebih akurat. Beberapa teknik pembagian dataset yang umum digunakan antara lain 90:10, 80:20, dan 70:20:10. Dalam penelitian ini, dataset dibagi menjadi 90% untuk training, 10% untuk validation, dan data untuk testing menggunakan data yang berada di luar data pada Gambar 9.



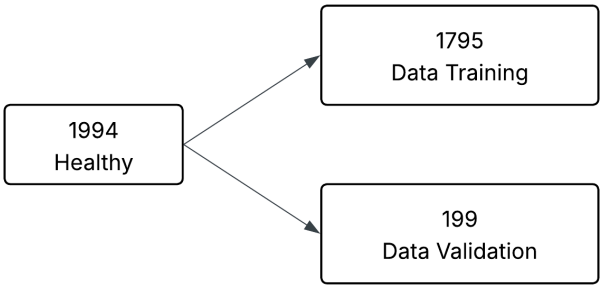
Gambar 9. Pembagian Dataset untuk Budok



Gambar 10. Pembagian Dataset untuk Bacterial Wilt



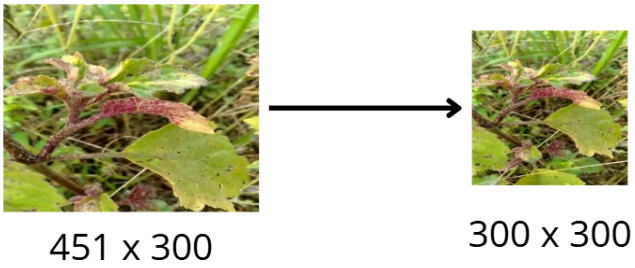
Gambar 11. Pembagian Dataset untuk Virus



Gambar 12. Pembagian Dataset untuk Healthy

2.8 *Implementasi Model CNN*

1. Resize Image
   Tahap ini bertujuan untuk melakukan standarisasi dengan mengubah skala piksel pada setiap gambar agar memiliki ukuran yang sama. Hal ini diperlukan karena gambar akan digunakan sebagai input untuk model yang akan dilatih. Model membutuhkan gambar input dengan ukuran tertentu, sehingga perlu ditetapkan ukuran gambar yang seragam untuk seluruh dataset seperti ditunjukkan pada gambar berikut:



Gambar 13. Penerapan Image Resize

2. Augmentation
   Augmentasi dataset dilakukan dengan mengubah data pelatihan yang sudah ada sebagai bagian dari proses perluasan dataset. Tujuan dari augmentasi ini adalah untuk mengurangi kemungkinan terjadinya overfitting, yaitu kondisi ketika model terlalu menghafal dataset. Beberapa teknik augmentasi yang digunakan meliputi rotation range, width shift range, height shift range, horizontal flip, vertical flip, shear range, dan zoom range.

```
# Membuat objek ImageDataGenerator untuk augmentasi data training
train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True
)

#Membuat objek ImageDataGenerator tanpa augmentasi untuk data validation
val_datagen = ImageDataGenerator(rescale=1./255)
```

Gambar 14. Penerapan Augmentasi

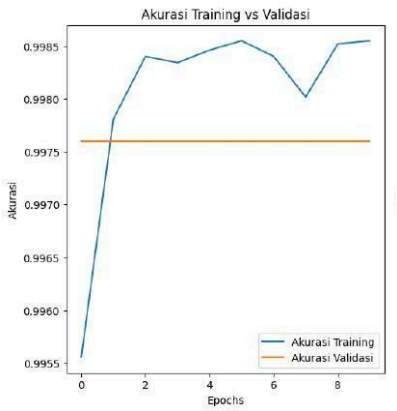
3. Feature Extraction
   Setelah melalui tahap praproses data, langkah selanjutnya adalah melakukan ekstraksi fitur pada gambar. Proses ini melibatkan dua tahap, di mana pertama, ekstraksi gambar dilakukan dengan memanfaatkan data augmentasi dari data pelatihan dan data validasi. Selanjutnya, ditentukan komposisi ukuran gambar yang akan diterapkan pada model CNN. Pada penelitian ini, ukuran gambar yang digunakan adalah 128 x 128. Implementasinya dapat dilihat pada Gambar 15.

```
[ ]  # # Menggunakan objek ImageDataGenerator untuk memuat data dari direktori
    train_generator = train_datagen.flow_from_directory(
        train_path,
        target_size=(128, 128),
        batch_size=32,
        class_mode='categorical'
    )

    val_generator = val_datagen.flow_from_directory(
        val_path,
        target_size=(128, 128),
        batch_size=32,
        class_mode='categorical'
    )
```

Gambar 15. Penerapan Ekstraksi Fitur

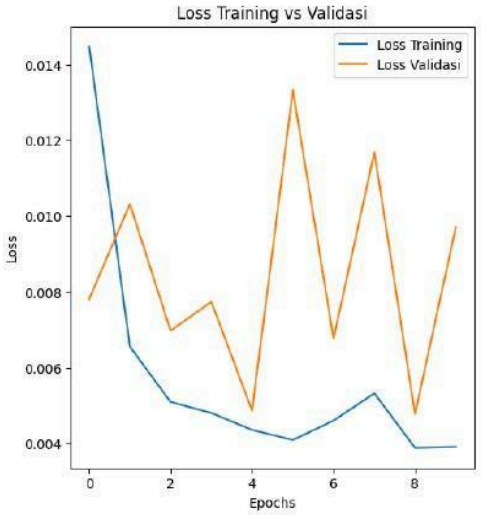
4. Model Evaluation
   Berikut adalah Gambar 13 yang menunjukkan akurasi pelatihan (training accuracy) dan akurasi validasi (validation accuracy) pada algoritma CNN.



Gambar 16. Akurasi Pelatihan dan Validasi

Grafik di atas menunjukkan akurasi pelatihan (biru) dan akurasi validasi (oranye) dari sebuah model Machine Learning selama beberapa epoch. Akurasi pelatihan meningkat tajam hingga mencapai sekitar 0,9995, sementara akurasi validasi tetap stabil di angka sekitar 0,9973. Perbedaan yang cukup besar antara keduanya mengindikasikan kemungkinan terjadinya overfitting, yaitu kondisi ketika model terlalu menyesuaikan diri dengan data pelatihan sehingga kurang mampu melakukan generalisasi pada data baru. Diperlukan evaluasi lebih lanjut pada data pengujian serta penyesuaian parameter atau arsitektur model untuk meningkatkan kemampuan generalisasi dan mengurangi risiko overfitting.

Berikut adalah Gambar 17 yang menunjukkan nilai loss pada pelatihan (training loss) dan validasi (validation loss) pada algoritma CNN.



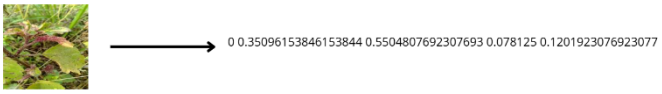
Gambar 17. Loss Pelatihan dan Validasi

Grafik di atas menunjukkan training loss (biru) dan validation loss (oranye) selama beberapa epoch. Training loss menurun secara konsisten, menandakan bahwa model sedang belajar dari data pelatihan. Namun, validation loss sangat fluktuatif dan tidak menunjukkan penurunan yang stabil, yang mengindikasikan kemungkinan terjadinya overfitting. Model tampaknya kesulitan melakukan generalisasi pada data validasi, sehingga diperlukan penerapan teknik seperti early stopping atau regularization untuk meningkatkan kinerja model.

2.9 *Implementasi Model YOLO*

1. Labeling

   Pada tahap ini, dilakukan pelabelan gambar JPG yang dikonversi ke format TXT, yaitu proses menandai atau memberi anotasi pada gambar untuk menunjukkan lokasi atau batas dari objek tertentu di dalamnya. Dalam konteks identifikasi objek, label ini biasanya mencakup informasi mengenai lokasi dan ukuran objek yang terdeteksi. Secara khusus, pada format TXT yang digunakan untuk pelabelan gambar, setiap baris dalam file TXT biasanya berisi informasi tentang satu objek, termasuk kelas objek, koordinat pusat, serta dimensi (lebar dan tinggi) objek tersebut. Format ini nantinya akan digunakan dalam algoritma YOLO (You Only Look Once). Contoh format label dalam file TXT dapat dilihat pada Gambar 18.



Gambar 18. Implementasi Pelabelan

2. Inisialisasi Model YOLO
   Tahap ini melibatkan pembuatan model deteksi objek YOLO dan inisialisasi berbagai parameter serta bobot model. Inisialisasi dilakukan saat membuat objek model dari pustaka atau framework yang menyediakan implementasi YOLO. Dalam penelitian ini, YOLO v8 digunakan sebagai model untuk identifikasi, di mana model tersebut diimpor dari Ultralytics, sebuah pustaka open-source yang menyediakan implementasi YOLO yang kuat dan mudah digunakan. Untuk mengimpor model YOLO v8, dapat digunakan skrip seperti yang ditunjukkan pada Tabel 2 berikut.

Script	Keterangan
!pip install ultralytics -q	Install the ultralytics library via the Python package manager "pip".
From ultralytics import YOLO	Import the YOLO model from the ultralytics library by creating an instance of the YOLO class.
Model = YOLO("yolov8m.pt")	Create a YOLO v8 model object that contains model parameters and architecture.

Tabel 2. Skrip Import Model YOLO v8

3. Membuat Data.yaml
   File data.yaml berfungsi sebagai file konfigurasi untuk melatih model YOLO menggunakan pustaka Ultralytics. File ini menyimpan informasi penting terkait dataset pelatihan dan konfigurasi pelatihan. Untuk membuat file data.yaml, dapat digunakan perintah shell !touch data.yaml. Adapun isi dari file data.yaml dapat dilihat pada Gambar 19.

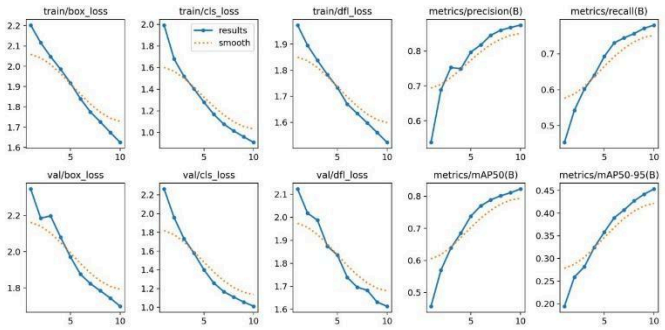
```
Created config file at /content/data.yaml

File contents:

path: /content/data
train: train/images
val: validation/images
nc: 4
names:
- sehat
- layu_bakteri
- virus
- budok
```

Gambar 19. Implementasi Pembuatan File data.yaml

4. Evaluasi Model
   Grafik pada Gambar 20 di bawah ini menunjukkan hasil eksperimen pelatihan kinerja model machine learning untuk tugas deteksi objek menggunakan algoritma YOLO.



Gambar 20. Hasil Eksperimen Model YOLO

Gambar tersebut menunjukkan grafik hasil pelatihan model deteksi objek, di mana sumbu X mewakili jumlah epoch dan sumbu Y menunjukkan nilai metrik kinerja model. Grafik ini dibagi menjadi dua bagian, yaitu hasil pada data pelatihan (train) dan data validasi (val). Terlihat bahwa nilai loss untuk box, klasifikasi (cls), dan distribusi (dfl) menurun pada kedua data tersebut, yang menunjukkan peningkatan kemampuan model dalam mempelajari fitur objek. Selain itu, nilai matrik evaluasi seperti precision, recall, mAP50, dan mAP50-95 menunjukkan tren kenaikan yang konsisten, menandakan bahwa model semakin baik dalam mendeteksi dan mengklasifikasikan objek dengan akurat seiring bertambahnya epoch. Hal ini mencerminkan proses pelatihan yang efektif serta kemampuan generalisasi model yang baik terhadap data yang sebelumnya belum pernah dilihat.

3. HASIL DAN PEMBAHASAN

Performa dan efektivitas algoritma pada sistem yang dikembangkan dievaluasi menggunakan matrik perbandingan. Studi ini menggunakan sejumlah matrik evaluasi, termasuk:

3.1 Training Time

Waktu pelatihan (training time) adalah total waktu yang diperlukan untuk melatih model menggunakan data pelatihan. Semakin cepat waktu pelatihan, semakin baik model dapat berkembang dan menyesuaikan diri. Waktu pelatihan dapat dipengaruhi oleh jumlah data pelatihan, spesifikasi sistem, dan kompleksitas model.

Epoch	Yolov5l (s)	Yolov8l (s)	Yolo11l (s)	MobileNet (s)
1	202	219	235	119
2	214	231	245	95
3	220	243	258	95
4	229	250	268	98
5	218	236	248	139
6	212	229	240	96
7	226	245	262	97
8	222	238	255	93
9	228	240	258	94
10	228	241	260	97

Total	2199	2372	2529	1023
Minute	36,65 m	39,53 m	42,15 m	17,05 m

Tabel 3. Waktu Pelatihan di Setiap Epoch

3.2 Detection Speed

Kecepatan deteksi model adalah ukuran seberapa cepat model dapat mengidentifikasi fitur, pola, atau objek dalam sebuah gambar. Pada metrik ini, diperlukan respons deteksi yang cepat. Parameter perangkat keras, ukuran citra masukan, dan kompleksitas model dapat mempengaruhi kecepatan deteksi.

No	Yolovl5 (s)	Yolov8l (s)	Yolo11l (s)	MobileNet
1	1	2	1	9
2	2	1	1	8
3	1	1	1	10
4	2	2	1	7
5	1	1	1	9
6	2	1	1	8
7	1	2	1	10
8	2	1	1	7
9	1	1	1	8
10	2	2	1	9
Total	15	14	10	85
Average	1,5	1,4	1	8,5

Tabel 4. Kecepatan Deteksi Setiap Pengujian Objek

3.3 Testing Accuracy

Akurasi prediksi dapat didefinisikan sebagai perbandingan antara prediksi yang benar dengan jumlah total sampel. Metrik ini memberikan gambaran tentang seberapa baik model dapat menghasilkan keluaran yang akurat. Meskipun akurasi sangat berguna dalam banyak situasi, metrik ini mungkin kurang efektif ketika terdapat distribusi kelas yang tidak seimbang.

No	Yolovl5	Yolov8l	Yolo11l	MobileNet
1	75%	92%	88%	95%
2	78%	89%	85%	93%
3	72%	94%	87%	96%
4	90%	91%	86%	97%
5	76%	73%	84%	94%
6	79%	95%	88%	98%
7	74%	91%	89%	92%
8	77%	88%	83%	95%
9	81%	76%	86%	91%
10	73%	93%	85%	96%
Total	775%	882%	861%	947%
Average	77.5%	88,2%	86,1%	94,7%

Tabel 5. Hasil Akurasi pada Setiap Objek Uji

4. SIMPULAN

Penelitian ini berhasil mengembangkan sistem yang mampu mengidentifikasi tanaman nilam menggunakan teknologi visi komputer berbasis AI dengan algoritma YOLO. Berdasarkan temuan penelitian, algoritma YOLO yang digunakan memiliki keunggulan dalam mendeteksi objek dan lebih sesuai untuk aplikasi real-time. Hal ini menjadikan YOLO sebagai pilihan yang paling efisien untuk sistem pemantauan tanaman nilam yang membutuhkan proses identifikasi yang cepat dan akurat. Implementasi sistem ini diharapkan dapat meningkatkan efektivitas pemantauan tanaman nilam. Selain itu, penelitian ini memberikan kontribusi terhadap pengembangan teknologi berbasis AI di sektor pertanian, khususnya dalam meningkatkan produktivitas dan kualitas tanaman secara konsisten.

Berdasarkan temuan penelitian, agar sistem identifikasi tanaman nilam dapat diterapkan secara optimal, diperlukan pengujian langsung di lapangan (real deployment) untuk mengevaluasi kinerja sistem dalam kondisi nyata. Selain itu, disarankan agar sistem identifikasi ini terus dikembangkan dengan memperluas dataset agar mencakup data yang lebih beragam, sehingga dapat meningkatkan akurasi sistem dalam mengidentifikasi berbagai variasi tanaman nilam.

DAFTAR PUSTAKA

[1] Y. Zhu, C. Wang, H. Zhang, and J. Liu, “SerpensGate-YOLOv8: A Novel Lightweight and Accurate Model for Plant Disease Detection,” Frontiers in Plant Science, vol. 15, p. 1514832, 2024.

[2] D. R. Syahputra, B. Priambodo, and M. A. Rahman, “Deteksi dan Klasifikasi Hama Potato Beetle pada Tanaman Kentang Menggunakan Yolov8,” J. Teknol. Inform. dan Ilmu Komput., vol. 11, no. 2, pp. 205–213, 2024.

[3] R. Handayani, M. A. Fithri, and D. R. Nasution, “Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih Resnet101,” J. RESTI, vol. 7, no. 1, pp. 95–102, 2023.

[4] R. Maulana and T. Hariyadi, “Klasifikasi Penyakit Daun Bawang Menggunakan Algoritma CNN Xception,” Pros. Inotek, vol. 8, no. 2, pp. 202–210, 2023.

[5] R. A. Irwanto and S. R. Dwi, “Identifikasi Tanaman Obat Indonesia Melalui Citra Daun Menggunakan Metode Convolutional Neural Network (CNN),” J. Teknol. Inform. dan Ilmu Komput., vol. 10, no. 1, pp. 93–100, 2023.

[6] I. N. Dewi, M. Sari, and D. S. Hasibuan, “Isolasi Bakteri Rhizosfer Tanaman Nilam (Pogostemon Cablin Benth.) Yang Berpotensi Sebagai Penghasil Senyawa Antibakteri Terhadap Bakteri Penyebab Infeksi Saluran Pencernaan,” Wahana Oeuvre Humaniora, vol. 7, no. 1, pp. 56–63, 2023.

[7] M. A. Azmi, D. S. Putri, and L. Irwanda, “Klasifikasi Kematangan Tanaman Hidroponik Pakcoy Menggunakan Metode SVM,” J. RESTI, vol. 6, no. 4, pp. 741–748, 2022.

[8] Z. Zuyasna, H. Hamdani, and L. Lestari, “Pertumbuhan tanaman nilam MV1 varietas Lhokseumawe akibat konsentrasi dan lama perendaman kolkisin,” Agro Bali: Agricultural Journal, vol. 4, no. 1, pp. 23–33, 2021.

[9] S. Saosang, N. Mambuhu, and H. A. Katili, “Analisis tingkat kesuburan tanah pada tanaman nilam (Pogostemon cablin) di Desa Balingara dan Desa Bella Kecamatan Nuhon,” Jurnal Ilmiah Mahasiswa Fakultas Pertanian, vol. 2, no. 1, pp. 155–161, 2022.

[10] Z. Zulfan, H. Fitri, and M. Mahdani, “Pertumbuhan Tanaman Nilam MV1 Varietas Lhokseumawe Akibat Konsentrasi dan Lama Perendaman Kolkisin,” Jurnal Agro, vol. 13, no. 2, pp. 87–95, 2021.

[11] A. Santoso and A. Nur, “Perbanyakan Tanaman Nilam (Pogostemon cablin Benth.) Menggunakan Sistem Fotoautotrofik dengan Berbagai Konsentrasi Gula dan Jumlah Ventilasi,” J. Agron. Indones., vol. 50, no. 3, pp. 275–284, 2022.

[12] A. Esteva et al., “Deep learning-enabled medical computer vision,” npj Digital Medicine, vol. 4, no. 1. Nature Research, Dec. 01, 2021. doi: 10.1038/s41746-020-00376-2.

[13] Y. Nurhadi, D. I. Mulyana, and Y. Akbar, "Klasifikasi rumput liar menggunakan algoritma deep learning dengan Dense Convolutional Neural Network," Progresif: Jurnal Ilmiah Komputer, vol. 19, no. 1, pp. 347–358, Feb. 2023.
[14] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," Electronic Markets, vol. 31, no. 3, pp. 685–695, 2021.
[15] A. W. Mulia, I. Ruslianto, and D. M. Midyanti, "Klasifikasi kerusakan jalan pada citra jalan raya Pontianak dan sekitarnya dengan menggunakan Convolutional Neural Network," Coding: Jurnal Komputer dan Aplikasi, vol. 11, no. 1, pp. 11–20, 2023.
[16] N. N. C. Sumartha, I. G. P. S. Wijaya, F. Bimantoro, and G. S. Nugraha, "Klasifikasi citra lubang pada permukaan jalan beraspal dengan metode Convolutional Neural Networks," J-COSINE: Journal of Computer Science and Informatics Engineering, vol. 8, no. 1, pp. 31–38, Jun. 2024.
[17] Zalmi, W. F., Sitompul, B. J. D., Nduru, S. W., & Lumbangaol, S. K. (2023). Welder Recruitment Decision Support System using the SMARTER method. SAGA Journal of Technology and Information System, 1(2), 44–49. https://doi.org/10.58905/saga.v1i2.98
[18] Solli, N. S., & Sompie, S. R. (2021). Aplikasi Web UPT Bimbingan dan Konseling Universitas Sam Ratulangi | Jurnal Teknik Informatika. ejournal.unsrat.ac.id. https://doi.org/10.35793/jti.v16i4.34145
[19] Yusupa, A., Tarigan, V., & Sengkey, D. F. (2025c). Development of Virtual Lab on Collision Dynamics Learning Object with Collision Algorithm Integration. Journal of Applied Informatics and Computing, 9(1), 44–54. https://doi.org/10.30871/jaic.v9i1.8765
[20] Zalmi, W. F. (2023). Sistem Aplikasi Surat Masuk dan Surat Keluar Berbasis Web pada Dinas Kehutanan Provinsi Sumatera Utara. JURNAL INFORMATIKA, 11(1), 12–25. https://doi.org/10.36987/informatika.v11i1.3844