

## **SISTEM DETEKSI HAMA PADA TANAMAN JAGUNG (ZEA MAYS) BERBASIS KECERDASAN BUATAN DAN INTERNET OF THINGS (IoT)**

Edwin Tenda<sup>1</sup>, Stephano C. W. Ngangi<sup>2</sup>, Christian A. J. Soewoeh<sup>3</sup>, Eliasta Ketaren<sup>4</sup>

Program Studi Sistem Informasi

<sup>123</sup>Universitas Sam Ratulangi

Email: [tenda.edwin@unsrat.ac.id](mailto:tenda.edwin@unsrat.ac.id)<sup>1</sup>, [stephano.ngangi@unsrat.ac.id](mailto:stephano.ngangi@unsrat.ac.id)<sup>2</sup>, [christia.suwuh@unsrat.ac.id](mailto:christia.suwuh@unsrat.ac.id)<sup>3</sup>,  
[eliastaketaren@unsrat.ac.id](mailto:eliastaketaren@unsrat.ac.id)<sup>4</sup>

### **Abstrak**

Produktivitas tanaman jagung (*Zea mays*) di Indonesia masih menghadapi tantangan serius akibat serangan hama yang sering terlambat terdeteksi. Metode konvensional berbasis pengamatan visual manual memiliki keterbatasan dari sisi waktu, ketelitian, dan ketergantungan pada tenaga ahli. Penelitian ini bertujuan mengembangkan sistem deteksi hama tanaman jagung berbasis kecerdasan buatan dan Internet of Things (IoT) yang mampu bekerja secara real-time pada perangkat edge. Sistem dikembangkan menggunakan model klasifikasi YOLOv8n-cls dengan pendekatan transfer learning dan diimplementasikan pada Raspberry Pi 4. Dataset yang digunakan terdiri dari citra ulat grayak, penggerek batang, dan daun sehat, dengan strategi augmentasi dinamis untuk mengatasi ketidakseimbangan data. Hasil pengujian menunjukkan akurasi validasi mencapai lebih dari 98% dengan waktu inferensi rata-rata 40–50 ms per frame. Implementasi sistem membuktikan bahwa integrasi AI dan IoT efektif untuk mendukung deteksi dini hama jagung secara cepat dan efisien, serta berpotensi meningkatkan produktivitas pertanian berbasis smart farming.

**Kata kunci:** Deteksi hama, jagung, kecerdasan buatan, YOLOv8, Internet of Things

### **1. Pendahuluan**

Jagung (*Zea mays* L.) merupakan salah satu komoditas pertanian strategis di Indonesia yang memiliki peran penting dalam ketahanan pangan nasional. Selain berfungsi sebagai sumber pangan setelah padi, jagung juga menjadi bahan baku utama industri pakan ternak, industri pangan olahan, serta bioenergi [1]. Seiring dengan pertumbuhan penduduk dan meningkatnya kebutuhan sektor peternakan, permintaan terhadap jagung nasional terus mengalami peningkatan dari tahun ke tahun. Namun demikian, peningkatan permintaan tersebut belum sepenuhnya diimbangi dengan peningkatan produktivitas yang stabil di tingkat petani.

Data statistik pertanian menunjukkan bahwa produksi jagung di Indonesia masih menghadapi berbagai kendala, baik yang bersifat teknis maupun non-teknis. Salah satu faktor dominan yang memengaruhi penurunan produktivitas jagung adalah serangan hama tanaman [1]. Hama seperti ulat grayak (*Spodoptera frugiperda*) dan penggerek batang jagung (*Ostrinia furnacalis*) dapat menyebabkan kerusakan serius pada daun dan batang tanaman, terutama pada fase vegetatif awal [2]. Kerusakan tersebut berdampak langsung pada penurunan kemampuan fotosintesis, pertumbuhan tanaman yang tidak optimal, hingga penurunan hasil panen secara signifikan apabila tidak dilakukan pengendalian secara tepat waktu.

Dalam praktiknya, deteksi dan identifikasi hama tanaman jagung di tingkat lapangan masih didominasi oleh metode konvensional berbasis pengamatan visual manual. Petani atau penyuluh pertanian biasanya mengandalkan pengalaman dan intuisi untuk mengenali gejala serangan hama pada tanaman. Pendekatan ini memiliki sejumlah keterbatasan, antara lain bersifat subjektif, membutuhkan waktu dan tenaga yang besar, serta rentan terhadap kesalahan identifikasi, terutama pada tahap awal serangan ketika gejala visual belum terlihat jelas [8]. Akibatnya, tindakan pengendalian sering kali dilakukan terlambat atau tidak tepat sasaran, sehingga meningkatkan risiko kerugian ekonomi bagi petani.

Perkembangan teknologi kecerdasan buatan (Artificial Intelligence/AI), khususnya di bidang computer vision, telah banyak digunakan untuk menganalisis citra tanaman dan mengidentifikasi berbagai kondisi tanaman secara otomatis. Model deep learning berbasis convolutional neural network (CNN) terbukti mampu mengekstraksi fitur visual kompleks dari citra tanaman dan melakukan klasifikasi dengan tingkat akurasi yang tinggi [3], [6]. Berbagai penelitian menunjukkan bahwa teknologi machine learning dan deep learning efektif digunakan untuk deteksi dini penyakit dan hama tanaman melalui analisis citra daun dan batang [6], [9].

Di sisi lain, teknologi Internet of Things (IoT) memungkinkan integrasi perangkat keras seperti kamera, sensor, dan unit komputasi dalam satu sistem yang saling terhubung. Dalam konteks pertanian, IoT memungkinkan proses pemantauan kondisi tanaman secara real-time dan berkelanjutan langsung di lapangan [10]. Integrasi AI dan IoT menjadi fondasi utama dalam pengembangan konsep pertanian cerdas (smart farming), di mana pengambilan keputusan dapat dilakukan secara lebih cepat dan berbasis data [7].

Meskipun demikian, penerapan AI dan IoT di sektor pertanian masih menghadapi sejumlah tantangan. Banyak sistem deteksi berbasis AI yang bergantung pada pemrosesan berbasis cloud atau server berdaya komputasi tinggi, sehingga kurang sesuai untuk diterapkan di lingkungan pertanian dengan keterbatasan infrastruktur jaringan [10]. Selain itu, model deep learning berukuran besar umumnya membutuhkan sumber daya komputasi yang tinggi, sehingga meningkatkan biaya implementasi dan membatasi penerapannya pada petani skala kecil.

Pendekatan edge computing menjadi solusi yang menjanjikan untuk mengatasi permasalahan tersebut. Dengan melakukan proses inferensi langsung pada perangkat di lapangan, ketergantungan terhadap koneksi internet dapat diminimalkan dan latensi sistem dapat ditekan [5], [7]. Penggunaan model AI yang ringan dan dioptimalkan untuk perangkat embedded memungkinkan sistem deteksi hama bekerja secara mandiri dan efisien.

Berdasarkan latar belakang tersebut, penelitian ini mengusulkan pengembangan sistem deteksi hama tanaman jagung berbasis kecerdasan buatan dan Internet of Things yang diimplementasikan pada perangkat edge menggunakan Raspberry Pi. Model klasifikasi citra dikembangkan menggunakan arsitektur YOLOv8n-cls yang memiliki ukuran model relatif kecil dan kecepatan inferensi tinggi [4]. Untuk meningkatkan efisiensi inferensi pada CPU perangkat embedded, model dikonversi ke format Open Neural Network Exchange (ONNX) dan dijalankan menggunakan ONNX Runtime [5].

Kontribusi utama penelitian ini terletak pada pengembangan sistem end-to-end yang mencakup proses akuisisi citra, pra-pemrosesan data, inferensi model AI, dan visualisasi hasil secara real-time di perangkat edge. Berbeda dengan penelitian sebelumnya yang umumnya berfokus pada klasifikasi citra secara offline atau berbasis cloud [6], [9], penelitian ini menekankan implementasi langsung di lapangan dengan mempertimbangkan keterbatasan sumber daya dan kebutuhan praktis petani. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi nyata dalam pengembangan teknologi pertanian cerdas yang aplikatif, efisien, dan berbiaya rendah.

Penelitian ini bertujuan untuk mengembangkan sistem deteksi hama tanaman jagung berbasis kecerdasan buatan dan IoT yang diimplementasikan pada perangkat edge. Sistem dirancang agar mampu bekerja secara mandiri tanpa ketergantungan pada koneksi internet, sehingga sesuai untuk kondisi pertanian di daerah dengan keterbatasan infrastruktur jaringan. Kontribusi utama penelitian ini terletak pada integrasi model YOLOv8n-cls yang ringan dengan perangkat Raspberry Pi sebagai solusi deteksi hama real-time yang efisien dan aplikatif.

## 2. Landasan Teori

### 2.1 Tanaman Jagung dan Permasalahan Hama

Jagung (*Zea mays* L.) merupakan tanaman pangan semusim yang memiliki nilai ekonomi tinggi dan peran strategis dalam sistem pertanian nasional. Tanaman ini relatif adaptif terhadap berbagai kondisi agroklimat, sehingga banyak dibudidayakan di berbagai wilayah Indonesia. Meskipun demikian, produktivitas tanaman jagung sangat dipengaruhi oleh gangguan organisme pengganggu tanaman (OPT), khususnya hama [1].

Hama utama pada tanaman jagung antara lain ulat grayak (*Spodoptera frugiperda*), penggerek batang (*Ostrinia furnacalis*), kutu daun (*Rhopalosiphum maidis*), dan belalang. Ulat grayak dikenal sebagai hama invasif yang menyerang daun jagung secara agresif dan dapat menyebabkan kehilangan hasil panen yang signifikan apabila tidak dikendalikan sejak dini [2]. Serangan hama umumnya sulit dideteksi pada tahap awal karena gejala visual yang masih terbatas dan sering kali menyerupai gangguan fisiologis tanaman. Oleh karena itu, deteksi dini hama menjadi faktor kunci dalam upaya pengendalian yang efektif dan berkelanjutan. Beberapa citra hama jagung dapat dilihat pada Tabel 1.

**Tabel 1.** Citra Hama Jagung

| No | Hama Jagung                                               | Gambar                                                                               |
|----|-----------------------------------------------------------|--------------------------------------------------------------------------------------|
| 1. | Ulat grayak<br>( <i>Spodoptera frugiperda</i> )           |  |
| 2. | penggerek batang jagung<br>( <i>Ostrinia furnacalis</i> ) |  |
| 3. | Kutu daun<br>( <i>Rhopalosiphum maidis</i> )              |  |
| 4. | Belalang<br>( <i>Caelifera</i> )                          |  |

### 2.2 Internet of Things dalam Pertanian

Internet of Things (IoT) merupakan konsep sistem yang menghubungkan berbagai perangkat fisik melalui jaringan komunikasi untuk saling bertukar data secara otomatis. IoT memungkinkan objek fisik seperti sensor, kamera, dan aktuator untuk mengumpulkan data dari lingkungan dan mengirimkannya ke sistem pemrosesan tanpa intervensi manusia secara langsung [10].

Dalam konteks pertanian, IoT berperan penting dalam mendukung penerapan pertanian presisi (precision agriculture). IoT memungkinkan pemantauan kondisi tanaman, lingkungan, dan serangan hama secara real-time dan berkelanjutan. Data yang diperoleh dari perangkat IoT dapat digunakan untuk mendukung pengambilan keputusan yang lebih cepat dan berbasis data, sehingga meningkatkan efisiensi pengelolaan lahan pertanian [7]. Integrasi IoT dengan sistem kecerdasan buatan memungkinkan terciptanya sistem pertanian cerdas (smart farming) yang adaptif dan responsif terhadap perubahan kondisi lapangan.

### **2.3 Kecerdasan Buatan dalam Pertanian**

Kecerdasan buatan (Artificial Intelligence/AI) telah menjadi salah satu teknologi utama dalam transformasi sektor pertanian modern. AI memungkinkan sistem komputer untuk melakukan analisis, pembelajaran, dan pengambilan keputusan berdasarkan data dalam jumlah besar. Berbagai pendekatan AI, seperti machine learning, deep learning, dan sistem pakar, telah diterapkan untuk menyelesaikan permasalahan pertanian, termasuk prediksi hasil panen, optimasi penggunaan sumber daya, serta deteksi hama dan penyakit tanaman [3].

Penelitian oleh Kamilaris dan Prenafeta-Boldú menunjukkan bahwa penerapan AI dalam pertanian mampu meningkatkan efisiensi dan produktivitas melalui analisis data lingkungan dan tanaman secara otomatis [3]. Pendekatan berbasis AI juga terbukti mampu mengurangi ketergantungan pada input kimia seperti pestisida dan pupuk dengan mendukung pengambilan keputusan yang lebih tepat sasaran.

### **2.4 Metode Kecerdasan Buatan untuk Deteksi Hama**

Berbagai metode kecerdasan buatan telah digunakan dalam sistem deteksi dan diagnosis hama tanaman. Metode machine learning klasik seperti Support Vector Machine (SVM) telah digunakan untuk mengklasifikasikan hama tanaman berdasarkan fitur citra dan menunjukkan tingkat akurasi yang cukup tinggi [8]. Selain itu, sistem pakar berbasis metode Certainty Factor dan Dempster-Shafer juga banyak diterapkan untuk mendiagnosis hama dan penyakit tanaman berdasarkan basis pengetahuan dan aturan yang telah ditentukan [11], [12].

Metode ensemble learning seperti Random Forest juga digunakan untuk mengidentifikasi tingkat keparahan penyakit tanaman melalui analisis citra digital. Pendekatan ini dinilai mampu menangani data dengan variasi tinggi dan mengurangi risiko overfitting dibandingkan metode klasifikasi tunggal [9]. Meskipun metode-metode tersebut menunjukkan hasil yang baik, sebagian besar penelitian masih berfokus pada sistem berbasis rule-based atau klasifikasi offline yang belum sepenuhnya mendukung implementasi real-time di lapangan.

### **2.5 Deep Learning dan Convolutional Neural Network**

Perkembangan deep learning membawa kemajuan signifikan dalam bidang pengolahan citra digital. Convolutional Neural Network (CNN) merupakan arsitektur deep learning yang dirancang khusus untuk menangani data visual dan telah terbukti unggul dalam tugas klasifikasi dan deteksi objek [5]. CNN mampu mengekstraksi fitur visual secara hierarkis, mulai dari fitur sederhana seperti tepi dan tekstur hingga fitur kompleks yang merepresentasikan bentuk objek.

Dalam konteks pertanian, CNN banyak digunakan untuk mendeteksi penyakit dan hama tanaman melalui analisis citra daun dan batang. Mohanty et al. menunjukkan bahwa pendekatan deep learning berbasis CNN mampu mengungguli metode machine learning tradisional dalam mendeteksi penyakit tanaman dengan tingkat akurasi yang tinggi [6]. Keunggulan CNN terletak pada kemampuannya mempelajari fitur secara otomatis tanpa memerlukan ekstraksi fitur manual.

### **2.6 YOLO dan Model Ringan untuk Edge Computing**

YOLO (You Only Look Once) merupakan salah satu arsitektur deep learning yang dirancang untuk kecepatan dan efisiensi tinggi. Berbeda dengan pendekatan deteksi konvensional yang memisahkan tahap ekstraksi fitur dan klasifikasi, YOLO melakukan proses deteksi dalam satu tahap (single-stage detection), sehingga sangat cocok untuk aplikasi real-time [4].

YOLOv8 merupakan pengembangan terbaru dari keluarga YOLO yang menawarkan peningkatan performa dan fleksibilitas. Varian YOLOv8n-cls dirancang sebagai model ringan untuk tugas klasifikasi dengan ukuran parameter yang kecil dan latensi rendah, sehingga sesuai untuk implementasi pada perangkat embedded dengan keterbatasan sumber daya komputasi. Penggunaan model ringan menjadi penting dalam konteks edge computing, di mana proses inferensi dilakukan langsung pada perangkat di lapangan tanpa bergantung pada server eksternal [7].

### **2.7 Edge Computing dan Format ONNX**

Edge computing merupakan paradigma komputasi yang memindahkan proses pemrosesan data dari pusat (cloud) ke perangkat tepi jaringan (edge). Pendekatan ini memungkinkan pengurangan latensi, peningkatan privasi data, dan pengurangan ketergantungan pada koneksi internet [7]. Dalam sistem pertanian berbasis IoT, edge computing memungkinkan sistem deteksi hama bekerja secara mandiri dan responsif terhadap kondisi lapangan. Open Neural Network Exchange (ONNX) merupakan format standar terbuka yang memungkinkan pertukaran model deep learning antar framework. Format ONNX bersifat platform-agnostik dan dioptimalkan untuk inferensi berkecepatan tinggi pada CPU, sehingga banyak digunakan dalam implementasi AI pada perangkat embedded.

Penggunaan ONNX Runtime memungkinkan eksekusi model yang lebih efisien dibandingkan runtime framework pelatihan, seperti PyTorch, khususnya pada perangkat dengan sumber daya terbatas [5].

### 3. Metode Penelitian

Penelitian ini menggunakan pendekatan eksperimental dengan tujuan mengembangkan dan mengevaluasi sistem deteksi hama tanaman jagung berbasis kecerdasan buatan dan Internet of Things (IoT) yang mampu bekerja secara real-time pada perangkat edge computing. Metode penelitian disusun dalam beberapa tahapan utama, yaitu pengumpulan data, pra-pemrosesan data, pengembangan dan pelatihan model AI, konversi model ke format ONNX, implementasi sistem pada perangkat IoT, serta evaluasi kinerja sistem.

#### 3.1 Arsitektur Umum Sistem

Arsitektur sistem yang dikembangkan terdiri atas tiga komponen utama, yaitu perangkat akuisisi data, unit pemrosesan edge, dan modul visualisasi hasil. Perangkat akuisisi data berupa kamera yang digunakan untuk menangkap citra tanaman jagung secara langsung di lapangan. Unit pemrosesan edge menggunakan Raspberry Pi 4 sebagai perangkat utama untuk menjalankan proses inferensi model AI. Modul visualisasi digunakan untuk menampilkan hasil deteksi hama secara real-time kepada pengguna.

Pendekatan edge computing dipilih untuk meminimalkan latensi sistem dan mengurangi ketergantungan pada koneksi internet, sehingga sistem tetap dapat beroperasi pada lingkungan pertanian dengan keterbatasan infrastruktur jaringan [7]. Seluruh proses, mulai dari pengambilan citra hingga penentuan kelas hama, dilakukan secara lokal pada perangkat Raspberry Pi.

#### 3.2 Dataset dan Akuisisi Data

Dataset yang digunakan dalam penelitian ini terdiri dari citra tanaman jagung yang mencakup tiga kelas utama, yaitu:

1. daun jagung sehat,
2. tanaman jagung yang terserang ulat grayak (*Spodoptera frugiperda*), dan
3. tanaman jagung yang terserang penggerek batang (*Ostrinia furnacalis*).

Citra diperoleh dari kombinasi dataset publik dan pengambilan data lapangan menggunakan kamera digital. Penggunaan dataset publik bertujuan untuk memperkaya variasi data dan meningkatkan kemampuan generalisasi model, sebagaimana direkomendasikan dalam penelitian sebelumnya terkait deteksi hama dan penyakit tanaman [6]. Seluruh citra dikurasi secara manual untuk memastikan kualitas data dan kesesuaian label.

Dataset dibagi menjadi dua bagian, yaitu data latih (80%) dan data validasi (20%). Pembagian ini dilakukan secara acak dengan tetap menjaga proporsi jumlah data pada setiap kelas agar tidak terjadi ketimpangan distribusi kelas yang signifikan.

#### 3.3 Pra-pemrosesan dan Augmentasi Data

Sebelum digunakan dalam proses pelatihan, seluruh citra melalui tahap pra-pemrosesan. Tahapan ini meliputi:

1. perubahan ukuran citra menjadi  $128 \times 128$  piksel,
2. normalisasi nilai piksel ke rentang 0–1, dan
3. penyesuaian format kanal warna sesuai kebutuhan model.

Untuk meningkatkan keragaman data dan mengurangi risiko overfitting, diterapkan teknik augmentasi data selama proses pelatihan. Teknik augmentasi yang digunakan meliputi rotasi citra, horizontal flipping, dan vertical flipping. Pendekatan ini terbukti efektif dalam meningkatkan performa model deep learning pada dataset berskala terbatas [6], [5].

#### 3.4 Pengembangan dan Pelatihan Model AI

Model kecerdasan buatan yang digunakan dalam penelitian ini adalah YOLOv8n-cls, yaitu varian ringan dari keluarga YOLOv8 yang dirancang khusus untuk tugas klasifikasi citra [4]. Model ini dipilih karena memiliki ukuran parameter yang relatif kecil dan kecepatan inferensi yang tinggi, sehingga sesuai untuk implementasi pada perangkat embedded.

Pelatihan model dilakukan menggunakan pendekatan transfer learning, di mana bobot awal model diinisialisasi dari model yang telah dilatih pada dataset ImageNet. Pendekatan ini memungkinkan model memperoleh representasi fitur awal yang kuat dan mempercepat proses konvergensi pelatihan [5]. Proses pelatihan dilakukan selama sejumlah epoch yang telah ditentukan dengan menggunakan optimizer AdamW dan loss function cross-entropy.

Evaluasi performa model selama pelatihan dilakukan menggunakan metrik akurasi pada data validasi. Model dengan performa terbaik disimpan sebagai best model dalam format PyTorch (.pt) untuk selanjutnya digunakan pada tahap implementasi.

#### 3.5 Konversi Model ke Format ONNX

Model best.pt yang dihasilkan dari proses pelatihan berbasis PyTorch tidak sepenuhnya optimal untuk inferensi pada CPU perangkat embedded. Oleh karena itu, model dikonversi ke format Open Neural Network Exchange

(ONNX). Format ONNX bersifat platform-agnostik dan mendukung optimisasi inferensi pada berbagai perangkat keras, termasuk CPU berbasis ARM seperti Raspberry Pi [5].

Proses konversi dilakukan menggunakan export tool yang disediakan oleh framework Ultralytics. Model hasil konversi kemudian dijalankan menggunakan ONNX Runtime, yang berperan sebagai inference engine utama dalam sistem. Penggunaan ONNX Runtime bertujuan untuk menurunkan latensi inferensi dan meningkatkan stabilitas sistem selama operasi real-time.

### 3.6 Implementasi Sistem Real-Time pada Raspberry Pi

Implementasi sistem dilakukan pada perangkat Raspberry Pi 4 yang berfungsi sebagai unit pemrosesan edge. Kamera dihubungkan melalui antarmuka CSI dan diaktifkan menggunakan utilitas konfigurasi sistem. Lingkungan pemrograman Python disiapkan dengan menginstal beberapa pustaka utama, yaitu opencv-python-headless untuk akuisisi citra, numpy untuk manipulasi data numerik, dan onnxruntime untuk menjalankan inferensi model.

Sebuah skrip Python dikembangkan untuk menjalankan proses deteksi secara real-time. Skrip ini melakukan pengambilan citra secara berkelanjutan menggunakan cv2.VideoCapture, diikuti dengan tahap pra-pemrosesan citra yang meliputi konversi warna, perubahan ukuran, normalisasi, dan penyesuaian dimensi input. Citra hasil pra-pemrosesan kemudian dimasukkan ke dalam sesi ONNX Runtime untuk memperoleh hasil prediksi kelas. Nilai probabilitas dihitung menggunakan fungsi softmax, dan hasil klasifikasi ditampilkan secara langsung pada layar menggunakan antarmuka OpenCV.

### 3.7 Evaluasi Kinerja Sistem

Evaluasi kinerja sistem dilakukan dari dua aspek utama, yaitu akurasi klasifikasi dan waktu inferensi. Akurasi klasifikasi diukur menggunakan data validasi untuk memastikan kemampuan model dalam mengenali kelas hama dengan tepat. Waktu inferensi diukur dengan menghitung durasi proses inferensi untuk setiap frame citra pada Raspberry Pi.

Hasil evaluasi menunjukkan bahwa sistem mampu mencapai waktu inferensi rata-rata sekitar 40–50 ms per frame, yang menunjukkan bahwa sistem layak digunakan untuk aplikasi real-time di lapangan. Evaluasi ini menjadi dasar untuk menilai kelayakan arsitektur YOLOv8n-cls dan format ONNX dalam mendukung sistem deteksi hama tanaman jagung berbasis edge computing.

## 4. Hasil Penelitian

### 4.1 Akuisisi data

Proses pelatihan model klasifikasi citra menggunakan arsitektur YOLOv8n-cls menghasilkan performa yang sangat baik pada data validasi. Model menunjukkan akurasi validasi lebih dari 98%, yang mengindikasikan bahwa model mampu mengenali pola visual pada citra tanaman jagung dengan tingkat ketepatan yang tinggi. Tingginya akurasi ini menunjukkan bahwa pendekatan transfer learning yang digunakan efektif dalam membantu model mempelajari fitur penting meskipun dataset yang digunakan relatif terbatas [5], [6]. Distribusi detail dari dataset yang telah diolah disajikan pada Tabel 2.

**Tabel 2.** Distribusi Dataset

| Kelas            | Jumlah Citra | Data Latih (80%) | Data Validasi (20%) |
|------------------|--------------|------------------|---------------------|
| Ulat Grayak      | 595          | 476              | 119                 |
| Penggerek Batang | 5            | 4                | 1                   |
| Daun Sehat       | 35           | 28               | 7                   |
| Total            | 635          | 508              | 127                 |

Selama proses pelatihan, kurva akurasi menunjukkan tren peningkatan yang stabil dan konvergen, sementara nilai loss mengalami penurunan secara konsisten. Hal ini menandakan bahwa model tidak mengalami overfitting yang signifikan dan mampu melakukan generalisasi dengan baik pada data validasi. Penerapan teknik augmentasi data seperti rotasi dan flipping turut berkontribusi dalam meningkatkan keragaman data latih dan mengurangi ketergantungan model pada pola visual tertentu [6]. Parameter Augmentasi yang digunakan dapat dilihat pada Tabel 3.

**Tabel 3.** Distribusi Dataset

| Kelas   | Nilai | Deskripsi                       |
|---------|-------|---------------------------------|
| Degrees | 20    | Rotasi acak ( $\pm 20^\circ$ )  |
| Fliplr  | 0.5   | 50% kemungkinan flip horizontal |
| Flipud  | 0.1   | 10% kemungkinan flip vertikal   |

### 4.2 Pengembangan Model Cerdas dengan YOLO

Arsitektur Model dalam penelitian ini menggunakan model Convolutional Neural Network (CNN) dari

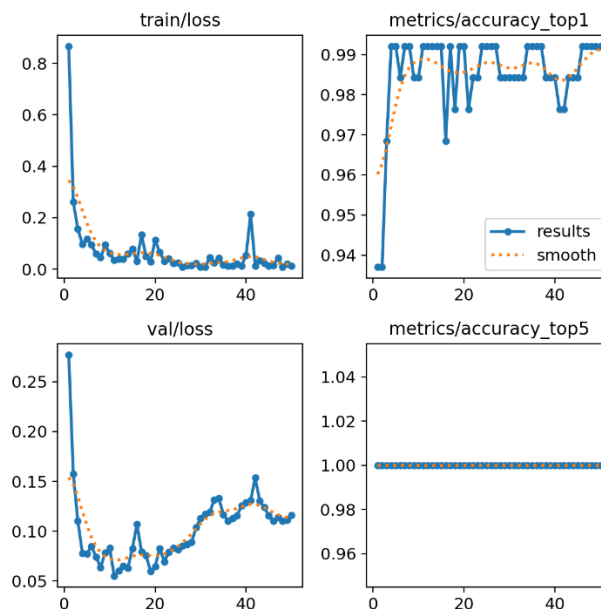
keluarga YOLO (You Only Look Once), yaitu YOLOv8n-cls. Arsitektur ini dipilih berdasarkan pertimbangan spesifik untuk tujuan implementasi pada perangkat edge. Varian 'n' (Nano): Ini adalah versi teringan dan tercepat dari keluarga YOLOv8. Model ini memiliki jumlah parameter yang jauh lebih sedikit dibandingkan versi lain (seperti 's', 'm', 'l', 'x'), sehingga ideal untuk inferensi berkecepatan tinggi pada perangkat dengan sumber daya komputasi terbatas seperti CPU Raspberry Pi 4. Modul 'cls' (Classification): Model ini difokuskan murni untuk tugas klasifikasi gambar. Tujuannya adalah untuk menetapkan satu label (misal, 'ulat grayak') untuk keseluruhan gambar yang di-input, bukan deteksi objek (yang akan menggambar kotak di sekitar hama).

Model ini dilatih menggunakan metode transfer learning, di mana bobot pre-trained dari dataset raksasa ImageNet digunakan sebagai titik awal. Bobot ini kemudian disesuaikan (fine-tuned) menggunakan dataset hama jagung yang telah disiapkan untuk mengenali 3 kelas target. Parameter Training dan Konfigurasi Proses training dilakukan di lingkungan Google Colab yang dilengkapi dengan GPU (Nvidia T4/P100) untuk mempercepat komputasi. Model dilatih selama 50 epochs, di mana satu epoch adalah satu siklus penuh model melihat keseluruhan data latih. penelitian ini menggunakan metode validasi Hold-Out, di mana data dibagi satu kali menjadi 80% data latih dan 20% data validasi. Parameter teknis, hyperparameter, dan konfigurasi augmentasi yang digunakan selama training dirangkum secara detail pada Tabel 4.

**Tabel 4.** Parameter Pelatihan Model YOLO

| Kategori       | Parameter        | Nilai                | Deskripsi                                           |
|----------------|------------------|----------------------|-----------------------------------------------------|
| Arsitektur     | Bobot Awal       | ImageNet             | Bobot pre-trained untuk transfer learning.          |
|                | Model            | YOLOv8n-cls          | Arsitektur model klasifikasi.                       |
|                | Epochs           | 50                   | Jumlah total iterasi training.                      |
|                | Image Size       | 128x128              | Resolusi gambar input (px).                         |
|                | Batch Size       | 32                   | Jumlah gambar per iterasi.                          |
| Lingkungan     | Platform         | Google Colab         | Lingkungan training berbasis cloud.                 |
|                | Akselerator      | GPU (Nvidia T4/P100) | Perangkat keras untuk mempercepat komputasi.        |
| Hyperparameter | Epochs           | 50                   | Jumlah total siklus training.                       |
|                | Batch Size       | 32                   | Jumlah gambar yang diproses per iterasi.            |
|                | Ukuran Gambar    | 128x128              | Resolusi gambar input (px).                         |
|                | Optimizer        | AdamW                | (Default YOLOv8) Optimizer untuk penyesuaian bobot. |
| Metodologi     | Validasi         | Hold-Out (80/20)     | Pembagian 80% latih dan 20% validasi.               |
|                | K-Fold           | Tidak Digunakan      | Validasi silang tidak diterapkan.                   |
| Augmentasi     | Rotasi (degrees) | 20                   | Rotasi acak gambar ( $\pm 20^\circ$ ).              |
|                | Flip Horizontal  | 0.5                  | 50% probabilitas membalik gambar secara horizontal. |
|                | Flip Vertikal    | 0.1                  | 10% probabilitas membalik gambar secara vertikal.   |

Dari hasil pelatihan model yang dilakukan di peroleh hasil sebagaimana tampak pada Gambar 1.

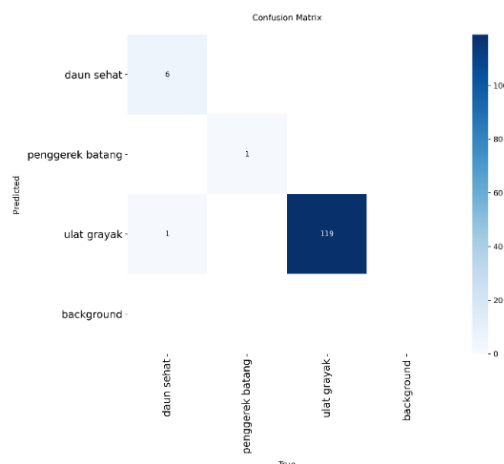


**Gambar 1.** Performa Model

Dari Gambar 5, dapat diobservasi beberapa temuan kunci:

- Akurasi Keseluruhan: Model mencapai performa yang sangat tinggi. Grafik metrics/accuracy\_top1 (b) menunjukkan akurasi validasi stabil di atas 98%, mendekati 99%. Grafik metrics/accuracy\_top5 (d) bahkan mencapai 100%, yang berarti prediksi yang benar selalu ada dalam 5 tebakan teratas model.
- Kurva Pembelajaran: Grafik train/loss (a) dan val/loss (c) menunjukkan model belajar sangat cepat, dengan loss menurun drastis pada 15 epoch pertama.
- Indikasi Overfitting: Setelah epoch ~20, kurva val/loss (c) menunjukkan tren kenaikan tipis. Ini adalah indikasi overfitting ringan, di mana model mulai menghafal data training dan performanya pada data validasi (data baru) mulai sedikit menurun. Namun, framework YOLOv8 secara otomatis menyimpan bobot model terbaik (best.pt) dari epoch dengan val/loss terendah, sehingga mitigasi overfitting telah terjadi secara efektif.
- Analisis Performa per Kelas (Confusion Matrix) Meskipun akurasi keseluruhan sekitar 99%, penting untuk menganalisis performa pada setiap kelas secara spesifik.

Untuk menganalisis performa pada setiap kelas secara spesifik maka perlu melakukan evaluasi melalui Confusion Matrix, sebagaimana tampak pada Gambar 2.



**Gambar 2.** Confusion Matrix Model pada Validasi

Dari Gambar 2 maka dapat di simpulkan bahwa :

- Performa ulat grayak (*Recall* 100%): Model ini sempurna dalam mendeteksi ulat grayak. Seluruh 119 sampel ulat grayak berhasil diprediksi dengan benar.
- Performa daun sehat (*Recall* 85.7%): Dari 7 sampel daun sehat, 6 diprediksi benar, namun 1 sampel salah diprediksi sebagai ulat grayak.
- Performa penggerek batang (Tidak Signifikan): Model memprediksi 1 sampel dengan benar. Dengan hanya satu data uji, performa 100% pada kelas ini tidak signifikan secara statistik.

4. Akurasi Miring (*Biased Accuracy*): Temuan ini mengkonfirmasi bahwa akurasi sekitar 99% yang tinggi pada Gambar 5 bersifat miring (biased). Akurasi ini didominasi oleh kemampuan superior model dalam memprediksi kelas ulat grayak, yang mencakup 93.7% dari seluruh data validasi.

#### 4.3 Implementasi Perangkat Raspberry Pi

Tahap akhir adalah mengimplementasikan model yang telah dilatih pada perangkat edge Raspberry Pi 4 untuk pengujian real-time. Konfigurasi Raspberry Pi dan kamera di tunjukan pada Gambar 3.



**Gambar 3.** Konfigurasi Raspberry Pi 4 dan Kamera Raspberry

Konversi Model ke ONNX Model best.pt yang dihasilkan (berbasis PyTorch) tidak ideal untuk inferensi CPU pada perangkat embedded. Oleh karena itu, model dikonversi ke format ONNX (Open Neural Network Exchange). Format ONNX bersifat platform-agnostik dan sangat dioptimalkan untuk inferensi berkecepatan tinggi pada CPU, sehingga sempurna untuk Raspberry Pi. Konfigurasi Sistem Real-Time Implementasi pada Raspberry Pi 4 melibatkan langkah-langkah berikut:

1. Aktivasi Kamera: Antarmuka kamera (CSI) diaktifkan melalui utilitas raspi-config.
2. Instalasi Library: Lingkungan Python disiapkan dengan menginstal library esensial: opencv-python headless (untuk akuisisi frame video), numpy (untuk manipulasi data), dan onnxruntime (sebagai engine inferensi).
3. Pengembangan Skrip Inferensi: Sebuah skrip Python dikembangkan untuk menjalankan loop deteksi real-time: cv2.VideoCapture(0) digunakan untuk membuka stream video dari kamera.
4. Setiap frame yang ditangkap diproses melalui fungsi preprocess (konversi BGR ke RGB, resize ke 128x128, normalisasi 0-1, dan transposisi ke format CHW).
5. Frame yang telah diproses dimasukkan ke session.run() milik ONNX Runtime.
6. Hasil prediksi (probabilitas) dihitung menggunakan softmax dan ditampilkan ke layar secara real-time menggunakan cv2.putText().

Implementasi sistem pada perangkat Raspberry Pi 4 menunjukkan bahwa model YOLOv8n-cls yang telah dikonversi ke format ONNX mampu berjalan secara stabil dan responsif. Pengujian dilakukan dengan menjalankan sistem secara berkelanjutan untuk menangkap dan memproses citra dari kamera secara real-time. Hasil pengujian menunjukkan bahwa waktu inferensi rata-rata berada pada kisaran 40–50 ms per frame, yang setara dengan sekitar 20–25 frame per detik. Waktu inferensi tersebut menunjukkan bahwa sistem memenuhi kriteria aplikasi real-time untuk pemantauan tanaman di lapangan. Dibandingkan dengan pendekatan berbasis cloud, sistem yang diimplementasikan secara edge computing memiliki keunggulan berupa latensi yang lebih rendah dan tidak bergantung pada kualitas koneksi internet [7]. Hal ini menjadi aspek penting dalam konteks pertanian di daerah pedesaan yang sering kali memiliki keterbatasan infrastruktur jaringan. Contoh deteksi dilihat pada beberapa Gambar 4.



**Gambar 4.** Hasil deteksi objek (Ulat Grayak)

## 5. Kesimpulan

- a. Penelitian ini berhasil mengembangkan sistem deteksi hama tanaman jagung berbasis kecerdasan buatan dan Internet of Things yang diimplementasikan pada perangkat edge menggunakan Raspberry Pi, sehingga sistem

- dapat bekerja secara mandiri tanpa ketergantungan pada koneksi internet.
- b. Model klasifikasi citra YOLOv8n-cls yang digunakan mampu mencapai akurasi validasi lebih dari 98% dalam membedakan tanaman jagung sehat dan tanaman yang terserang hama, yang menunjukkan kemampuan model dalam mengenali pola visual serangan hama secara andal.
  - c. Implementasi sistem secara real-time menghasilkan waktu inferensi rata-rata sekitar 40–50 ms per frame, yang memenuhi kriteria performa untuk aplikasi pemantauan lapangan dan membuktikan efisiensi arsitektur edge computing yang digunakan.
  - d. Konversi model ke format Open Neural Network Exchange (ONNX) terbukti meningkatkan efisiensi inferensi dan stabilitas sistem pada perangkat embedded dengan sumber daya terbatas.
  - e. Secara keseluruhan, integrasi kecerdasan buatan dan Internet of Things dalam sistem yang dikembangkan menunjukkan potensi besar sebagai solusi deteksi dini hama tanaman jagung yang aplikatif, efisien, dan mendukung penerapan pertanian cerdas secara berkelanjutan

### Ucapan Terima kasih

Penulis menyampaikan terima kasih kepada Universitas Sam Ratulangi atas dukungan terhadap pelaksanaan penelitian ini, yang pendanaannya bersumber dari PNBP-BLU Universitas Sam Ratulangi Tahun 2025, sehingga penelitian ini dapat dilaksanakan dan diselesaikan dengan baik.

### Daftar Pustaka

- [1] Badan Pusat Statistik, Statistik Tanaman Pangan Jagung Indonesia 2023, Jakarta: BPS Republik Indonesia, 2023.
- [2] J. Liu, X. Wang, and S. Li, “Automatic pest recognition in crop fields using deep convolutional neural networks,” *IEEE Access*, vol. 7, pp. 124280–124291, 2019, doi: 10.1109/ACCESS.2019.2937852.
- [3] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018, doi: 10.1016/j.compag.2018.02.016.
- [4] Ultralytics, “YOLOv8: A state-of-the-art real-time object detection system,” Ultralytics Inc., 2023. [Online]. Available: <https://docs.ultralytics.com>
- [5] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, “Image classification using deep learning: A survey,” *IEEE Access*, vol. 9, pp. 128345–128371, 2021, doi: 10.1109/ACCESS.2021.3113565.
- [6] M. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” *Frontiers in Plant Science*, vol. 7, Art. no. 1419, 2016, doi: 10.3389/fpls.2016.01419.
- [7] H. Zhang, M. Wang, and Y. Li, “Edge computing for intelligent agriculture: Opportunities and challenges,” *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14015–14029, 2021, doi: 10.1109/IIOT.2021.3069967.
- [8] R. Rosdiana, A. F. Hidayat, and M. Arifin, “Deteksi hama tanaman palawija menggunakan metode Support Vector Machine,” *Jurnal Informatika*, vol. 15, no. 2, pp. 85–94, 2021.
- [9] J. Prasetyo, A. Nugroho, and R. A. Nugraha, “Identifikasi tingkat keparahan penyakit tanaman menggunakan metode Random Forest berbasis citra digital,” *Jurnal Sains dan Seni ITS*, vol. 8, no. 2, pp. A233–A238, 2019.
- [10] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010, doi: 10.1016/j.comnet.2010.05.010.
- [11] D. Sari, A. P. Wibowo, and E. M. Putra, “Sistem pakar diagnosis hama dan penyakit tanaman cabai besar menggunakan metode Dempster–Shafer,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 4, pp. 403–410, 2019.
- [12] A. Wihartiko, A. R. Hakim, and Y. A. Pranoto, “Sistem pakar diagnosis penyakit dan hama tanaman teh menggunakan metode Certainty Factor,” *Jurnal Rekursif*, vol. 9, no. 1, pp. 45–54, 2021.
- [13] L. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, “Machine learning in agriculture: A review,” *Sensors*, vol. 18, no. 8, Art. no. 2674, 2018, doi: 10.3390/s18082674.
- [14] A. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, “Big data in smart farming – A review,” *Agricultural Systems*, vol. 153, pp. 69–80, 2017, doi: 10.1016/j.agsy.2017.01.023.