RANCANG BANGUN SISTEM SMART HIDROPONIK UNTUK MONITORING DETEKSI HAMA PADA TUMBUHAN BERBASIS INTERNET OF THINGS

Sub Judul :

PERANCANGAN DAN IMPLEMENTASI SISTEM DETEKSI SERANGAN HAMA KUTU KEBUL DAN KUTU DAUN BERSAYAP PADA TANAMAN HIDROPONIK BAYAM BERBASIS CITRA DENGAN METODE YOLOv8n

SKRIPSI

ALFARIZKI NURACHMAN - 2207421041

PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK NEGERI JAKARTA

2026

PERANCANGAN DAN IMPLEMENTASI SISTEM DETEKSI SERANGAN HAMA KUTU KEBUL DAN KUTU DAUN BERSAYAP PADA TANAMAN HIDROPONIK BAYAM BERBASIS CITRA DENGAN METODE YOLOv8n

SKRIPSI

Dibuat untuk Melengkapi Syarat-Syarat yang Diperlukan untuk Memperoleh Diploma Empat Politeknik

ALFARIZKI NURACHMAN
2207421041

PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK NEGERI JAKARTA
2026

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama	: Alfarizki Nurachman
NIM	: 2207421041
Jurusan/Program Studi	: Teknik Informatika dan Komputer / Teknik Multimedia dan Jaringan
Judul skripsi	: PERANCANGAN DAN IMPLEMENTASI SISTEM DETEKSI SERANGAN HAMA KUTU KEBUL DAN KUTU DAUN BERSAYAP PADA TANAMAN HIDROPONIK BAYAM BERBASIS CITRA DENGAN METODE YOLOv8n

Menyatakan dengan sebenarnya bahwa skripsi ini benar-benar merupakan hasil karya saya sendiri, bebas dari peniruan terhadap karya dari orang lain. Kutipan pendapat dan tulisan orang lain ditunjuk sesuai dengan cara-cara penulisan karya ilmiah yang berlaku.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa dalam skripsi ini terkandung cirri-ciri plagiat dan bentuk-bentuk peniruan lain yang dianggap melanggar peraturan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

POLITEKNIK NEGERI JAKARTA

Depok, 29 Juni 2026

Yang membuat pernyataan

Alfarizki Nurachman

NIM 2207421041

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

LEMBAR PENGESAHAN

Skripsi diajukan oleh:

Nama	: Alfarizki Nurachman
NIM	: 2207421056
Program Studi	: Teknik Multimedia dan Jaringan
Judul Skipsi	: PERANCANGAN DAN IMPLEMENTASI SISTEM DETEKSI SERANGAN HAMA KUTU KEBUL DAN KUTU DAUN BERSAYAP PADA TANAMAN HIDROPONIK BAYAM BERBASIS CITRA DENGAN METODE YOLOv8n

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari Rabu Tanggal 1, Bulan Juni Tahun 2026 dan dinyatakan **LULUS.**

Disahkan Oleh

Pembimbing I	: Maria Agustin, S.Kom., M.Kom.	( )
Penguji I	: Dr. Prihatin Oktivasari S.Si.,M.Si	( )
Penguji II	: Susana Dwi Yulianti M.Kom.	( )
Penguji III	: Chandra Wirawan, S.Kom, M.Kom.	( )

Mengetahui:

Jurusan Teknik Informatika dan Komputer

Ketua

Dr. Anita Hidayati, S.Kom., M.Kom.

NIP.197908032003122003

POLITEKNIK NEGERI JAKARTA

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan skripsi yang berjudul "Perancangan dan Implementasi Sistem Deteksi Serangan Hama Kutu Kebul dan Kutu Daun Bersayap Pada Tanaman Hidroponik Bayam Berbasis Citra dengan Metode YOLOv8n". Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk memperoleh gelar Diploma IV di Politeknik Negeri Jakarta. Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, penyusunan skripsi ini tidak akan dapat terselesaikan dengan baik. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa yang selalu memberikan hikmat dan rahmatnya.
2. Kedua orang tua, saudara, dan sahabat yang telah memberikan doa dan dukungan sehingga penulis dapat menyelesaikan perkuliahan.
3. Ibu Maria Agustin, S.Kom., M.Kom. selaku dosen pembimbing skripsi yang telah memberikan bimbingan dan saran-saran kepada penulis.
4. Seluruh dosen dan staf di Jurusan Teknik Multimedia dan Jaringan yang telah banyak memberikan ilmu dan dukungan baik secara langsung maupun tidak langsung selama proses perkuliahan.
5. Rekan-rekan Teknik Multimedia dan Jaringan 2022 Politeknik Negeri Jakarta.

Penulis berharap semoga skripsi ini dapat membawa manfaat dan menambah wawasan bagi pembaca. Penulis menyadari masih banyak kekurangan yang perlu diperbaiki dalam penyusunan skripsi ini, oleh karena itu penulis sangat mengharapkan kritik dan saran dari pembaca untuk menyempurnakan laporan ini.

Depok, Juni 2026

Alfarizki Nurachman

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Politeknik Negeri Jakarta, saya bertanda tangan dibawah ini:

Nama	: Alfarizki Nurachman
NIM	: 2207421041
Jurusan/Program	: Teknik Informatika dan Komputer / Teknik Studi Multimedia dan Jaringan

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Politeknik Negeri Jakarta Hak Bebas Royalti Non-Eksklusif atas karya ilmiah saya yang berjudul:

PERANCANGAN DAN IMPLEMENTASI SISTEM DETEKSI SERANGAN HAMA KUTU KEBUL DAN KUTU DAUN BERSAYAP PADA TANAMAN HIDROPONIK BAYAM BERBASIS CITRA DENGAN METODE YOLOv8n

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksklusif ini Politeknik Negeri Jakarta Berhak menyimpan, mengalihmediakan/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Depok, 29 Juni 2026

Yang Menyatakan

Alfarizki Nurachman

NIM 2207421056

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

PERANCANGAN DAN IMPLEMENTASI SISTEM DETEKSI SERANGAN HAMA KUTU KEBUL DAN KUTU DAUN BERSAYAP PADA TANAMAN HIDROPONIK BAYAM BERBASIS CITRA DENGAN METODE YOLOv8n

ABSTRAK

Serangan hama merupakan salah satu faktor yang dapat menurunkan produktivitas tanaman bayam hidroponik. Proses pemantauan hama yang masih dilakukan secara manual memiliki keterbatasan dalam mendeteksi hama berukuran kecil secara cepat dan konsisten. Penelitian ini bertujuan mengimplementasikan sistem deteksi multi-hama berbasis YOLOv8n pada Raspberry Pi 4 untuk mendeteksi whitefly (Bemisia tabaci) dan winged aphid (Aphididae) secara real-time. Sistem diintegrasikan dengan mekanisme gantry berbasis ESP8266 untuk melakukan akuisisi citra secara otomatis, sedangkan model dikonversi ke format NCNN agar proses inferensi lebih efisien pada prosesor ARM. Dataset yang digunakan terdiri dari 3.041 citra dengan pembagian 70% data pelatihan, 20% data validasi, dan 10% data pengujian, serta dilakukan augmentasi untuk meningkatkan kemampuan generalisasi model. Hasil pelatihan menunjukkan bahwa model YOLOv8n dengan resolusi masukan 640 × 640 piksel memperoleh nilai mAP@50 sebesar 0,967, sedangkan konfigurasi 320 × 320 piksel menghasilkan waktu inferensi GPU tercepat sebesar 0,8 ms dengan penurunan akurasi yang relatif kecil. Pada pengujian lapangan, model YOLOv8n resolusi 640 × 640 dan 416 × 416 memperoleh nilai Precision, Recall, dan F1-Score sebesar 100%, sedangkan konfigurasi 320 × 320 menghasilkan Precision 97,32%, Recall 100%, dan F1-Score 98,96%. Pengujian pada Raspberry Pi 4 menunjukkan bahwa konfigurasi YOLOv8n 320 × 320 mampu mencapai latensi rata-rata 0,200 detik, kecepatan 4,41 FPS, penggunaan CPU 75,44%, RAM 19,85%, dan suhu prosesor 47,01°C. Hasil tersebut menunjukkan bahwa optimasi menggunakan NCNN memungkinkan implementasi sistem deteksi hama secara efisien pada perangkat edge tanpa memerlukan akselerator eksternal.

Kata kunci: deteksi hama, hidroponik, raspberry pi, sistem gantry, whitefly, winged aphid, YOLOv8n

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

DAFTAR ISI

SURAT PERNYATAAN BEBAS PLAGIARISME 1
LEMBAR PENGESAHAN 3
KATA PENGANTAR 4
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS 5
ABSTRAK 6
DAFTAR ISI 1
DAFTAR GAMBAR 4
DAFTAR TABEL 5
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Perumusan Masalah 5
1.3 Batasan Masalah 5
1.4 Tujuan dan Manfaat 6
1.5 Sistematika Penulisan 6
BAB II TINJAUAN PUSTAKA 8
2.1 Hidroponik 8
2.1.1 Hama pada Tanaman Bayam 8
2.2 Open Source Computer Vision Library (OPEN CV) 10
2.3 Convolutional Neural Network (CNN) 11
2.4 NCNN 12
2.5 You Only Look Once (YOLO) 13
2.5.1 YOLOv8 14
2.6 Raspberry Pi 4 Model B 15
2.7 Raspberry Pi Camera Module 15
2.8 NodeMCU ESP8266 17
2.9 Stepper Motor Nema 17 18
2.10 Timing Belt GT2 20
2.11 Power Supply 12V 21
2.12 Driver Stepper Motor A4988 22
2.13 Limit Switch 24
2.14 Flask 25
2.15 Roboflow 26
2.16 Google Colab 28
2.17 Pengujian Sistem 29
2.18 Penelitian Terdahulu 32

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB III METODE PENELITIAN....36
3.1 Rancangan Penelitian....36
3.2 Tahapan Penelitian....37
3.3 Objek Penelitian....39
BAB IV HASIL DAN PEMBAHASAN....41
4.1 Analisis Kebutuhan....41
4.1.1 Kebutuhan Fungsional....41
4.1.2 Kebutuhan Non-Fungsional....42
4.2 Perancangan Sistem....43
4.2.1 Perancangan Perangkat Keras dan Perangkat Lunak....43
4.2.2 Diagram Blok....46
4.2.3 Flowchart Sistem....47
4.2.3.1 Flowchart Sistem Node Camera....47
4.2.3.2 Flowchart Sistem Node Gantry....48
4.2.4 Rancangan Perangkat Keras....50
4.2.4.1 Racangan Node Camera dan Pinout....51
4.2.4.2 Racangan Node Gantry dan Pinout....52
4.2.5 Rancangan Model....54
4.2.5.1 Labelling....54
4.2.5.2 Training....57
4.2.5.3 Perancangan Transmisi Data....59
4.3 Implementasi Sistem....59
4.3.1 Implementasi Perangkat Keras....59
4.3.2 Implementasi Perangkat Lunak....60
4.3.2.1 Implementasi Akuisisi Citra....61
4.3.2.2 Implementasi Inferensi Model YOLOv8n-NCNN....63
4.3.2.3 Implementasi Komunikasi Raspberry Pi dengan ESP8266....66
4.3.2.4 Implementasi Pengiriman Data....67
4.3.2.5 Implementasi Pergerakan Gantry Pada ESP8266....68
4.4 Pengujian....70
4.4.1 Deskripsi Pengujian....70
4.4.2 Prosedur Pengujian....70
4.4.2.1 Prosedur Analisis Hyperparameter Model....70
4.4.2.2 Prosedur Pengujian Performa Deteksi Model....71
4.4.2.3 Prosedur Pengujian Performa Perangkat....72
4.4.2.4 Prosedur Pengujian Fungsionalitas....73
4.4.2.5 Prosedur Pengujian Reliabilitas....74

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4.4.3 Data Hasil Pengujian.....75
4.4.3.1 Data Analisis Hyperparameter Model.....75
4.4.3.2 Data Hasil Pengujian Performa Deteksi Model.....76
4.4.3.3 Data Hasil Pengujian Performa Perangkat.....81
4.4.3.4 Data Hasil Pengujian Fungsionalitas.....84
4.4.3.5 Data Hasil Pengujian Reliabilitas.....85
4.4.4 Analisis Data.....86
4.4.4.1 Analisis Hyperparameter Model.....86
4.4.4.2 Analisis Data Hasil Pengujian Performa Deteksi Model.....88
4.4.4.3 Analisis Data Hasil Pengujian Performa Perangkat.....91
4.4.4.4 Analisis Data Hasil Pengujian Fungsionalitas.....92
4.4.4.5 Analisis Data Hasil Pengujian Reliabilitas.....93
BAB V PENUTUP.....95
5.1 Kesimpulan.....95
5.2 Saran.....96
DAFTAR PUSTAKA.....97
DAFTAR RIWAYAT HIDUP.....102
LAMPIRAN.....103

POLITEKNIK NEGERI JAKARTA

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

DAFTAR GAMBAR

Gambar 2.1 Kutu Kebul 9
Gambar 2.2 Aphids 10
Gambar 2.3 Arsitektur CNN 12
Gambar 2.4 NCNN 13
Gambar 2.5 Arsitektur YOLO 14
Gambar 2.6 Raspberry Pi 4 Model B 15
Gambar 2.7 Raspi Camera Module OV5647 17
Gambar 2.8 NodeMCU ESP8266 18
Gambar 2.9 Stepper Motor Nema 17 20
Gambar 2.10 Timing Belt 21
Gambar 2.10 PSU 12V 22
Gambar 2.11 Driver Stepper Motor A4988 24
Gambar 2.12 Limit Switch 25
Gambar 2.13 Gambar Aplikasi Roboflow 28
Gambar 3.1 Hama Pada Tanaman di Lokatani 40
Gambar 4.1 Diagram Blok Sistem 47
Gambar 4.2 Flowchart Sistem Node Camera 48
Gambar 4.3 Flowchart Sistem Node Camera 50
Gambar 4.4 Rancangan Perangkat Keras 51
Gambar 4.5 Rancangan Node Camera 52
Gambar 4.6 Rancangan Node Gantry 54
Gambar 4.7 Proses Labeling Data 55
Gambar 4.8 Total Dataset Training, Validation dan Testing 57
Gambar 4.10 Implementasi Perangkat Keras 60
Gambar 4.11 Potongan Kode Implementasi Akuisisi Citra 62
Gambar 4.12 Potongan Kode Implementasi Inferensi Model 64
Gambar 4.13 Potongan Kode Implementasi Inferensi Model 65
Gambar 4.14 Potongan Kode Implementasi Inferensi Model 65
Gambar 4.15 Potongan Kode Implementasi Komunikasi Raspberry Pi dengan ESP8266 66
Gambar 4.16 Potongan Kode Implementasi Pengiriman Data 68
Gambar 4.17 Potongan Kode Implementasi Pergerakan Gantry 69
Gambar 4.18 Potongan Kode Implementasi Pergerakan Gantry 69
Gambar 4.19 Potongan Kode Implementasi Pergerakan Gantry 70
Gambar 4.20 Hasil Pengujian Model YOLOv8n 640 78
Gambar 4.21 Hasil Pengujian Model YOLOv8n 416 78
Gambar 4.22 Hasil Pengujian Model YOLOv8n 320 79
Gambar 4.23 Hasil Pengujian Model YOLOv5s 320 80
Gambar 4.24 Hasil Pengujian Model YOLO26n 320 80

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu....32
Tabel 4.1 Kebutuhan Penelitian.... 41
Tabel 4.2 Kebutuhan Penelitian.... 42
Tabel 4.3 Kebutuhan Penelitian.... 44
Tabel 4.4 Pinout Node Gantry....53
Tabel 4.5 Data Analisis Hyperparameter Model....76
Tabel 4.6 Ringkasan Hasil Pengujian Model YOLOv8n 640.... 77
Tabel 4.7 Ringkasan Hasil Pengujian Model YOLOv8n 416.... 78
Tabel 4.8 Ringkasan Hasil Pengujian Model YOLOv8n 320.... 78
Tabel 4.9 Ringkasan Hasil Pengujian Model YOLOv5s 320....79
Tabel 4.10 Ringkasan Hasil Pengujian Model YOLO26n 320.... 80
Tabel 4.11 Ringkasan Hasil Pengujian Performa Perangkat dengan Model YOLOv8n 640....81
Tabel 4.12 Ringkasan Hasil Pengujian Performa Perangkat dengan Model YOLOv8n 416....82
Tabel 4.13 Ringkasan Hasil Pengujian Performa Perangkat dengan Model YOLOv8n 320....82
Tabel 4.14 Ringkasan Hasil Pengujian Performa Perangkat dengan Model YOLOv5s 320.... 83
Tabel 4.15 Ringkasan Hasil Pengujian Performa Perangkat dengan Model YOLO26n 320....83
Tabel 4.16 Data Hasil Pengujian Fungsionalitas....84
Tabel 4.17 Ringkasan Hasil Pengujian Reliabilitas.... 86
Tabel 4.18 Status Operasional Selama Pengujian Reliabilitas....86
Tabel 4.17 Hasil Perhitungan Precision, Recall, dan F1-Score Setiap Model..... 89

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pertanian hidroponik merupakan salah satu metode budidaya tanaman yang semakin banyak diterapkan di Indonesia, terutama sebagai alternatif dalam mengatasi keterbatasan lahan pertanian. Sistem ini memanfaatkan larutan nutrisi sebagai media tumbuh sehingga penggunaan air dan unsur hara dapat dilakukan secara lebih efisien dibandingkan dengan budidaya konvensional. Meskipun demikian, keberhasilan budidaya hidroponik masih dipengaruhi oleh berbagai faktor, salah satunya adalah serangan organisme pengganggu tanaman yang dapat menurunkan kualitas maupun kuantitas hasil panen. Menurut Kardinan (2011), serangan organisme pengganggu tanaman dapat menyebabkan kehilangan hasil panen hingga 20–30% pada berbagai komoditas pertanian.

Tanaman bayam (*Amaranthus spp.*) merupakan salah satu komoditas yang populer dibudidayakan secara hidroponik karena memiliki siklus panen yang relatif singkat, yaitu berkisar antara 25–30 hari setelah tanam, serta menunjukkan adaptabilitas pertumbuhan yang baik pada sistem *Nutrient Film Technique* (NFT) maupun *Deep Water Culture* (DWC) (Mubarok et al., 2021; Adinda et al., 2024). Secara fisiologis, bayam mengandung beragam nutrisi penting seperti vitamin A, C, K, serta mineral esensial berupa zat besi dan kalsium yang menjadikannya sebagai sayuran fungsional bernilai gizi tinggi bagi pemenuhan kebutuhan harian masyarakat (Hidayanti & Kartika, 2019). Dari prospek ekonomi, bayam hidroponik memiliki nilai komersial yang menjanjikan seiring dengan meningkatnya permintaan pasar urban terhadap sayuran higienis, sehingga memicu tren peralihan ke sistem kultivasi modern berbasis *smart greenhouse* dan *Internet of Things* (IoT) demi menjaga kontinuitas serta kualitas produksi (Kartika et al., 2025).

Meskipun memiliki produktivitas yang tinggi, sistem budidaya hidroponik memiliki kerentanan tersendiri terhadap aspek proteksi tanaman. Ketersediaan nitrogen yang tinggi dalam larutan nutrisi hidroponik dapat memicu peningkatan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

akumulasi asam amino bebas pada jaringan daun, sehingga mengubah tanaman menjadi inang yang sangat atraktif bagi serangga herbivor (Xie et al., 2022; ul Ain, 2025). Dampaknya, tanaman ini rentan terserang berbagai jenis hama seperti kutu kebul (*whitefly*) dan kutu daun (*aphids*) yang dapat menyebabkan kerusakan signifikan pada komoditas bayam (Mureithi et al., 2017; Seni, 2018). Kondisi tersebut diperparah oleh pola penanaman hidroponik yang cenderung rapat, sehingga menciptakan kondisi yang mendukung perkembangan maupun penyebaran populasi hama. Oleh karena itu, keberadaan hama-hama tersebut memerlukan sistem monitoring yang efektif agar serangan dapat terdeteksi sejak dini sebelum menimbulkan kerugian ekonomi yang lebih besar.

Deteksi terhadap serangan hama sangat penting untuk meminimalkan kerusakan tanaman. Metode konvensional yang mengandalkan pengamatan manual oleh petani memiliki keterbatasan, terutama dalam hal akurasi, konsistensi, dan efisiensi waktu. Petani seringkali terlambat mendeteksi keberadaan hama hingga populasinya sudah mencapai tingkat yang membahayakan tanaman (Kardinan, 2011).

Berdasarkan hasil observasi awal dan wawancara dengan pengelola greenhouse hidroponik Lokatani, serangan hama masih menjadi salah satu permasalahan yang sering dijumpai selama proses budidaya. Hama yang paling banyak ditemukan adalah kutu kebul (*whitefly*) dan kutu daun (*aphids*), yang menyerang bagian daun sehingga berpotensi mengganggu pertumbuhan tanaman serta menurunkan kualitas hasil panen. Pada lokasi penelitian, kedua jenis hama tersebut kerap ditemukan pada tanaman bayam yang menjadi salah satu komoditas utama budidaya hidroponik. Kondisi tersebut menunjukkan bahwa keberadaan hama masih menjadi tantangan yang memerlukan penanganan secara cepat dan tepat agar kerugian yang ditimbulkan dapat diminimalkan.

Proses pemantauan hama yang diterapkan di greenhouse masih dilakukan melalui pengamatan visual oleh pekerja secara berkala. Meskipun metode ini mudah diterapkan, efektivitasnya sangat dipengaruhi oleh ketelitian pengamat dan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

frekuensi pemeriksaan. Hama yang berukuran kecil seperti kutu kebul dan kutu daun sering kali sulit dikenali pada tahap awal serangan sehingga keberadaannya baru disadari setelah populasi meningkat dan gejala kerusakan mulai tampak pada tanaman. Keterlambatan dalam mendeteksi keberadaan hama dapat menyebabkan tindakan pengendalian menjadi kurang optimal serta meningkatkan risiko penurunan produktivitas tanaman.

Perkembangan teknologi kecerdasan buatan, khususnya dalam bidang computer vision, telah membuka peluang baru dalam sistem deteksi hama secara otomatis. Algoritma *You Only Look Once (YOLO)* merupakan salah satu metode object detection yang memiliki keunggulan dalam hal kecepatan dan akurasi. Penelitian sebelumnya oleh Dinata (2023) telah mengimplementasikan YOLOv5 untuk deteksi hama belalang pada tanaman selada hidroponik dengan tingkat akurasi mencapai 90%.

Implementasi sistem deteksi hama pada embedded device seperti Raspberry Pi 4 memberikan beberapa keuntungan berupa portabilitas, efisiensi energi, dan biaya yang lebih terjangkau dibandingkan menggunakan komputer dengan spesifikasi tinggi. Wahab et al. (2021) telah membuktikan bahwa Raspberry Pi 4 mampu menjalankan model *deep learning* dengan performa yang baik melalui optimasi model untuk perangkat *embedded*, menjadi landasan bahwa pendekatan *on-device inference* pada Raspberry Pi 4 merupakan solusi yang layak untuk sistem deteksi hama di lapangan.

Penelitian mengenai integrasi *Artificial Intelligence* dan *Internet of Things* (AIoT) untuk deteksi hama telah berkembang pesat melalui berbagai pendekatan arsitektur dan algoritma. Chen et al. (2020) mengintegrasikan sensor lingkungan dengan algoritma LSTM untuk memprediksi kemunculan hama, serta menggunakan *Unmanned Aerial Vehicle* (UAV) berbasis YOLOv3 untuk pengumpulan citra dengan tingkat akurasi mencapai 90%. Di sisi lain, peningkatan performa deteksi pada level algoritma dilakukan oleh Dai et al. (2023) melalui modifikasi arsitektur YOLOv5m menggunakan *SwinTransformer*

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

pada bagian *neck* (SWinTR) dan metode *WConcat*, yang terbukti meningkatkan *feature fusion* serta memberikan hasil yang lebih *robust* dibandingkan model baseline. Tren komputasi kini juga bergeser ke arah *edge deployment* guna memangkas *latency* dan ketergantungan pada *cloud*. Nugroho et al. (2025) berhasil mengimplementasikan YOLOv8-nano (YOLOv8n) pada Raspberry Pi 4 untuk deteksi real-time, dibantu oleh Google Coral USB Accelerator guna menjaga efisiensi daya. Sementara itu, untuk objek berukuran mikro, Hakim et al. (2025) merancang *YOLO-pest* berbasis arsitektur YOLOv8x yang dioptimalkan untuk mendeteksi hama serangga kecil seperti kutu kebul (*whitefly*) dengan rata-rata akurasi mencapai 94%.

Namun, penelitian-penelitian tersebut memiliki beberapa keterbatasan yang perlu dikembangkan lebih lanjut. Pertama, sistem pada sebagian besar studi literatur terdahulu, seperti pada penelitian Dinata (2023), Chen et al. (2020), dan Nugroho et al. (2025), masih bersifat *single-class detection* yang hanya mampu mengenali satu jenis objek (masing-masing belalang, *Tessaratoma papillosa*, dan burung pipit). Padahal dalam praktiknya, komoditas tanaman sering kali diserang oleh berbagai jenis hama secara simultan. Kedua, terdapat *trade-off* yang signifikan antara kompleksitas model, kebutuhan perangkat keras, dan lokasi inferensi. Model *YOLO-pest* oleh Hakim et al. (2025) menggunakan varian YOLOv8x yang memiliki parameter besar sehingga proses inferensi harus dijalankan di sisi server, sedangkan riset Nugroho et al. (2025) membutuhkan perangkat akselerator eksternal tambahan berupa Google Coral USB TPU untuk menjalankan varian *nano*. Ketiga, belum ada optimasi khusus seperti kuantisasi atau kompilasi *framework* inferensi ringan untuk arsitektur *edge* murni tanpa akselerator hardware pada studi kasus hama mikro tanaman hidroponik spesifik.

Penelitian ini mengusulkan pengembangan sistem deteksi multi-hama secara *on-device inference* langsung pada Raspberry Pi 4 tanpa akselerator eksternal. Sistem ini difokuskan untuk mendeteksi hama kutu kebul (*whitefly*) dan kutu daun bersayap (*winged aphid*) pada tanaman bayam (*Amaranthus sp.*) hidroponik. Untuk mengatasi keterbatasan komputasi *hardware embedded*, digunakan model

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

YOLOv8-nano (YOLOv8n) yang dioptimalkan menggunakan *framework* NCNN. Pendekatan ini diharapkan mampu memangkas ukuran model dan mempercepat *inference time* secara signifikan melalui optimasi graf komputasi dan kuantisasi, sehingga deteksi hama mikro secara real-time dapat berjalan optimal dan efisien langsung di lapangan.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dapat dirumuskan beberapa permasalahan sebagai berikut:

1. Bagaimana merancang sistem deteksi dan klasifikasi multi-hama pada tanaman hidroponik berbasis computer vision menggunakan perangkat Raspberry Pi?
2. Bagaimana mengimplementasikan model YOLOv8 pada Raspberry Pi untuk melakukan deteksi hama secara real-time pada lingkungan hidroponik?
3. Bagaimana perbandingan performa YOLOv8, YOLOv5 dan YOLO26 dalam mendeteksi berbagai jenis hama berdasarkan precision, recall, F1-score, dan mAP@50?
4. Bagaimana perbandingan performa komputasi YOLOv8, YOLOv5 dan YOLO26 pada Raspberry Pi berdasarkan kecepatan inferensi, penggunaan CPU, penggunaan memori, dan kemampuan real-time sistem?

1.3 Batasan Masalah

Untuk membatasi rumusan masalah yang jelas mengenai perancangan sistem pendeteksi hama pada tanaman hidroponik bayam. Agar penelitian ini lebih fokus dan terarah, maka ditetapkan batasan masalah sebagai berikut:

1. Penelitian difokuskan pada deteksi dan klasifikasi dua jenis hama, yaitu kutu kebul (*whitefly*) dan kutu daun bersayap (*winged aphid*), pada tanaman bayam (*Amaranthus sp.*) hidroponik berskala kecil menggunakan citra digital.
2. Model yang digunakan dalam penelitian dibatasi pada YOLOv8n sebagai model utama, sedangkan YOLOv5s dan YOLO26n digunakan sebagai

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

model pembanding. Seluruh model dilatih menggunakan dataset, konfigurasi pelatihan, dan lingkungan komputasi yang sama.

3. Perbandingan performa deteksi dibatasi pada tiga model, yaitu YOLOv8n, YOLOv5s, dan YOLO26n, dengan parameter evaluasi mAP@50, Precision, Recall, dan F1-Score.
4. Perbandingan performa komputasi dibatasi pada parameter inference latency, frame per second (FPS), penggunaan CPU, penggunaan RAM, dan suhu prosesor selama proses inferensi pada Raspberry Pi 4.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Tujuan dari penelitian ini merupakan:

1. Merancang dan mengimplementasikan sistem deteksi serangan hama pada tanaman hidroponik berbasis citra menggunakan algoritma YOLOv8 yang dijalankan pada Raspberry Pi 4.
2. Mengukur kinerja sistem deteksi yang dikembangkan berdasarkan tingkat keberhasilan deteksi dan waktu inferensi pada Raspberry Pi 4

1.4.2 Manfaat

Manfaat yang diharapkan dari penelitian ini merupakan:

1. Membantu proses pemantauan kondisi daun tanaman bayam secara otomatis melalui sistem deteksi yang dijalankan pada perangkat embedded.
2. Menjadi referensi penerapan algoritma YOLOv8n pada Raspberry Pi untuk sistem pertanian cerdas berbasis embedded.

1.5 Sistematika Penulisan

Sistematika penulisan laporan penelitian ini disusun sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang penelitian yang menjelaskan pengembangan sistem deteksi hama menggunakan YOLOv8 pada Raspberry Pi 4, perumusan masalah yang akan diselesaikan, batasan masalah, serta tujuan dan manfaat penelitian.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB II TINJAUAN PUSTAKA

Bab ini membahas teori-teori yang menjadi landasan penelitian, meliputi: penelitian-penelitian terkait yang telah dilakukan sebelumnya, hidroponik, jenis-jenis hama pada tanaman bayam, computer vision dan deep learning, arsitektur YOLO, serta Raspberry Pi 4.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan metode dan tahapan penelitian yang dilakukan, meliputi: desain penelitian, objek penelitian, dan tahapan penelitian.

BAB IV HASIL DAN PEMBAHASAN

pengumpulan dan preparasi dataset, proses labeling dan augmentasi data, arsitektur model YOLOv8 yang digunakan, proses training dan hyperparameter tuning, optimasi model untuk deployment, implementasi pada Raspberry Pi 4, serta metode pengujian dan evaluasi performa sistem.

Bab ini menyajikan hasil analisis kebutuhan sistem, perancangan perangkat keras dan perangkat lunak, implementasi sistem deteksi hama berbasis YOLOv8n NCNN pada Raspberry Pi 4, serta data hasil pengujian sistem yang mencakup pengujian performa deteksi model, pengujian performa perangkat, dan pengujian fungsionalitas sistem. Selain itu, bab ini juga memuat analisis dan pembahasan terhadap seluruh hasil pengujian yang diperoleh.

BAB V PENUTUP

Bab ini berisi kesimpulan yang menjawab tujuan penelitian berdasarkan hasil pengujian dan analisis yang telah dilakukan, serta saran untuk pengembangan sistem lebih lanjut di masa mendatang.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB II
TINJAUAN PUSTAKA

2.1 Hidroponik

Hidroponik merupakan metode budidaya tanaman tanpa menggunakan media tanah, melainkan menggunakan air yang diperkaya dengan larutan nutrisi. Sistem hidroponik memungkinkan pertumbuhan tanaman yang lebih cepat dan hasil panen yang lebih tinggi dibandingkan metode konvensional karena nutrisi dapat diserap langsung oleh akar tanaman (Cantica et al., 2022).

Terdapat berbagai jenis sistem hidroponik yang umum digunakan, diantaranya merupakan *Nutrient Film Technique* (NFT), *Deep Flow Technique* (DFT), sistem sumbu (*wick system*), dan *Deep Water Culture* (DWC). Penelitian ini Menggunakan sistem NFT, yaitu teknik hidroponik dimana larutan nutrisi dialirkan secara kontinu dan tipis di sepanjang jalur pipa sehingga akar tanaman hanya tersentuh lapisan aliran nutrisi tanpa terendam sepenuhnya. Kualitas larutan nutrisi pada sistem NFT harus dikontrol secara berkala melalui pengukuran parameter *Parts Per Million* (PPM) sebagai indikator konsentrasi nutrisi dan nilai pH sebagai indikator keasaman larutan, guna memastikan tanaman memperoleh asupan nutrisi yang optimal sepanjang siklus pertumbuhan (Cantica et al., 2022).

2.1.1 Hama pada Tanaman Bayam

Tanaman bayam rentan terhadap serangan berbagai jenis hama, khususnya dari kelompok serangga penghisap cairan tanaman (Mureithi, Fiaboe, Ekesi & Meyhöfer, 2017). Dua jenis hama yang umum ditemukan pada pertanaman bayam merupakan kutu kebul (*whitefly*) dan kutu daun (*aphids*).

1. Kutu Kebul: Hama kutu kebul (*whitefly*) merupakan serangga mikro yang menyerang tanaman bayam dengan cara menusuk jaringan epidermis dan menghisap cairan nutrisi pada sel-sel daun. Hilangnya cairan sel ini, mengakibatkan tanaman mengalami gejala klorosis (menguning), tanaman menjadi kerdil karena kehilangan suplai fotosintat, dan dalam jangka panjang dapat menurunkan kualitas serta bobot kering hasil panen. Selain kerusakan langsung, kutu kebul juga berperan sebagai vektor utama

penyakit virus tanaman yang dapat memperburuk kondisi pertanaman secara keseluruhan (Mureithi et al., 2017).

Gambar 2.1 Kutu Kebul

2. Winged Aphids: *Winged Aphids* (atau kutu daun bersayap) merupakan hama mikro bertubuh lunak yang merusak tanaman dengan cara mengkolonisasi permukaan bawah daun atau pucuk muda tanaman bayam guna menghisap cairan floem. Aktivitas hama ini menyebabkan daun bayam mengalami distorsi pertumbuhan berupa daun menggulung, keriput, hingga mengalami nekrosis (kematian jaringan sel). Ekskresi dari aktivitas hisap aphids menghasilkan embun madu (*honeydew*) yang melapisi permukaan daun dan memicu pertumbuhan jamur jelaga (*sooty mold*), yang selanjutnya menghalangi masuknya cahaya matahari sehingga mengganggu proses fotosintesis tanaman (Seni, 2018; Mureithi et al., 2017).

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Gambar 2.2 Aphids

Mengingat ukuran tubuh kedua hama tersebut yang sangat kecil, deteksi dini secara visual oleh petani seringkali terlambat dilakukan hingga populasi hama sudah mencapai tingkat yang membahayakan tanaman. Oleh karena itu, diperlukan sistem deteksi otomatis berbasis citra yang mampu mengidentifikasi keberadaan hama sejak tahap awal serangan.

2.2 Open Source Computer Vision Library (OPEN CV)

OpenCV (Open Source Computer Vision Library) merupakan pustaka open source yang digunakan untuk pengolahan citra dan computer vision, yang awalnya dikembangkan oleh Intel dan dirancang untuk mendukung aplikasi dengan kebutuhan komputasi yang efisien dan bersifat real-time (Florentinus Budi Setiawan et al., 2021). OpenCV mendukung berbagai bahasa pemrograman seperti C, C++, Java, dan Python, serta dapat dijalankan pada berbagai sistem operasi, antara lain Windows, Linux, macOS, iOS, dan Android.

OpenCV menyediakan berbagai fungsi dan algoritma dalam pengolahan citra digital, meliputi akuisisi citra, pra-pemrosesan citra, deteksi objek, segmentasi citra, ekstraksi fitur, serta analisis visual lainnya. Dengan karakteristik performa yang tinggi dan dukungan lintas platform, OpenCV banyak digunakan dalam

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

pengembangan sistem visi komputer berbasis embedded device, termasuk pada implementasi sistem deteksi objek secara real-time (Florentinus Budi Setiawan et al., 2021).

Dalam penelitian ini, OpenCV diimplementasikan pada Raspberry Pi 4 sebagai komponen inti pipeline pengolahan citra. Fungsi utama OpenCV pada sistem ini meliputi akuisisi frame secara kontinu dari modul Raspberry Pi Camera OV5647, proses *preprocessing* berupa *resize* dimensi frame menjadi format 640×640 piksel sesuai standar input model YOLOv8n, serta visualisasi hasil inferensi berupa penggambaran *bounding box* dan label kelas hama beserta nilai *confidence score* di atas frame secara *real-time*. Frame hasil anotasi tersebut kemudian diteruskan ke Flask web server untuk disajikan melalui antarmuka *live stream* yang dapat diakses pengguna melalui *web browser*.

2.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salah satu arsitektur Neural Network yang dirancang khusus untuk menangani data berbasis citra, seperti gambar dan video. CNN mampu melakukan pembelajaran fitur secara otomatis melalui proses pelatihan menggunakan metode optimasi, seperti penurunan gradien, sehingga dapat mengekstraksi pola visual yang kompleks dari citra input (Sakib et al., 2018).

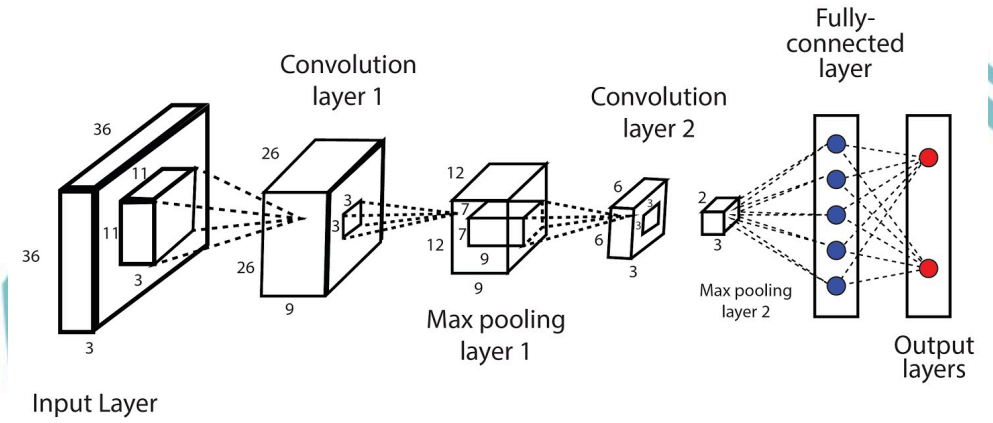
Arsitektur CNN umumnya tersusun atas beberapa lapisan utama, yaitu lapisan input, lapisan konvolusi (convolutional layer) yang berfungsi mengekstraksi fitur, lapisan pooling untuk mereduksi dimensi data, serta lapisan fully connected yang digunakan dalam proses klasifikasi atau prediksi akhir. Struktur berlapis ini menjadikan CNN sangat efektif dalam aplikasi computer vision, termasuk deteksi dan pengenalan objek (Sakib et al., 2018).

Dalam penelitian ini, CNN berperan sebagai pondasi utama arsitektur model YOLOv8n yang digunakan untuk mendeteksi hama pada tanaman bayam hidroponik. Model YOLOv8n memanfaatkan arsitektur CNN sebagai *backbone*

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

untuk mengekstraksi hierarki fitur visual dari citra input, mulai dari fitur tingkat rendah seperti tepi dan tekstur pada lapisan konvolusi awal, hingga fitur tingkat tinggi yang merepresentasikan karakteristik visual spesifik hama *whitefly* dan *winged aphid* pada lapisan konvolusi yang lebih dalam. Kemampuan CNN dalam mengekstraksi fitur secara otomatis inilah yang memungkinkan model YOLOv8n mengenali pola visual hama mikro milimeter secara akurat tanpa memerlukan rekayasa fitur manual.



Gambar 2.3 Arsitektur CNN
(sumber: www.analyticssteps.com)

2.4 NCNN

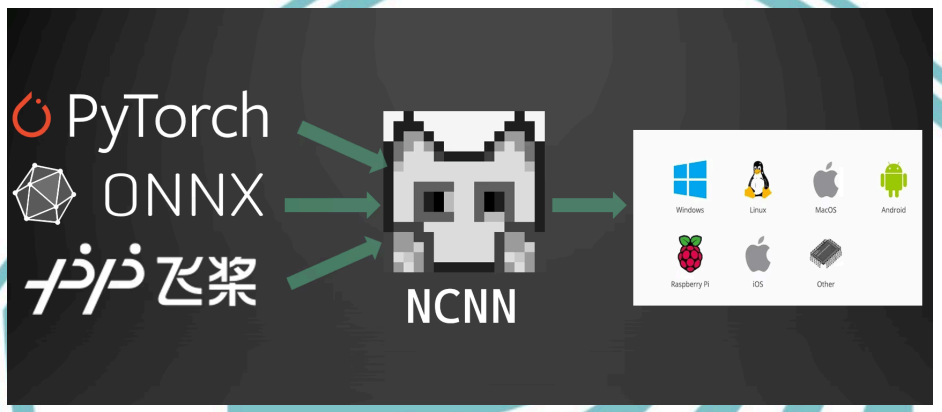
NCNN merupakan framework inferensi jaringan syaraf tiruan dengan performa tinggi yang dikembangkan oleh Tencent dan dirilis secara open source sejak tahun 2017. Framework ini dirancang khusus untuk deployment pada perangkat mobile dan embedded dengan sumber daya komputasi terbatas, tanpa bergantung pada library pihak ketiga (Nihui, 2017).

NCNN diimplementasikan menggunakan bahasa C++ murni dan dioptimalkan secara mendalam untuk arsitektur prosesor ARM melalui instruksi NEON SIMD, sehingga menghasilkan kecepatan inferensi yang lebih tinggi dibandingkan framework inferensi open source lainnya pada platform CPU berbasis ARM. Selain itu, NCNN mendukung akselerasi GPU melalui Vulkan API serta komputasi parallel multi-core, yang menjadikannya cocok untuk perangkat seperti Raspberry Pi yang menggunakan prosesor ARM Cortex-A72 (Nihui, 2017).

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Dalam penelitian ini, format NCNN dipilih karena mampu meningkatkan efisiensi inferensi yang lebih baik dibandingkan dengan model yang dijalankan dalam format PyTorch native yang membutuhkan dependensi besar dan alokasi memori yang lebih tinggi pada perangkat ARM. Karakteristik ini menjadikan NCNN sebagai pilihan optimal untuk mendukung proses deteksi hama secara *on-device* dengan latensi yang terukur dan stabil.



Gambar 2.4 NCNN

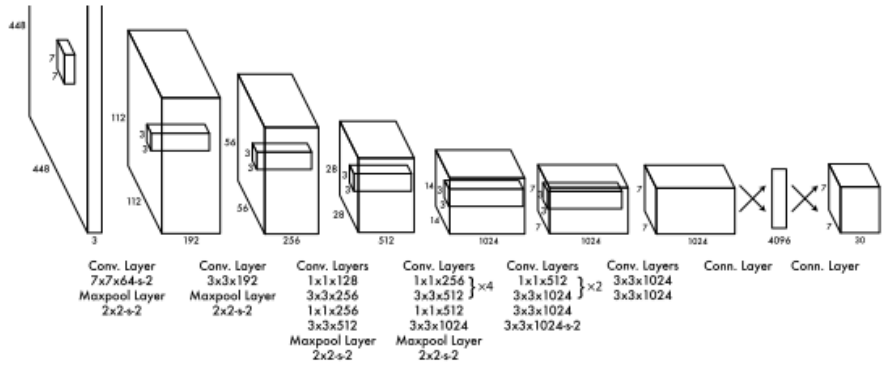
(Sumber : docs.ultralytics.com)

2.5 You Only Look Once (YOLO)

You Only Look Once (YOLO) merupakan algoritma deteksi objek yang menggunakan pendekatan one-stage detector, dimana deteksi objek dilakukan dalam satu forward pass melalui jaringan neural network. YOLO membagi gambar menjadi grid dan memprediksi bounding box serta probabilitas kelas untuk setiap grid cell secara simultan (Liu et al., 2018).

Arsitektur YOLO memanfaatkan Convolutional Neural Network (CNN) sebagai backbone untuk mengekstraksi fitur dari gambar input. Untuk setiap sel grid, YOLO memprediksi beberapa bounding box dan probabilitas kelas yang sesuai. Keunggulan utama YOLO merupakan kecepatan deteksi yang tinggi karena proses deteksi dilakukan dalam satu tahap, membuatnya cocok untuk aplikasi real-time (Guo et al., 2022). Karakteristik inilah yang menjadi dasar dalam pemilihan algoritma YOLO dalam penelitian ini, khususnya algoritma YOLOv8

varian nano yang dioptimalkan untuk perangkat *embedded* dengan sumber daya yang terbatas.



Gambar 2.5 Arsitektur YOLO

(Sumber : www.researchgate.net)

2.5.1 YOLOv8

YOLOv8 merupakan versi terbaru dari keluarga algoritma YOLO yang dikembangkan oleh Ultralytics. YOLOv8 menawarkan peningkatan signifikan dalam hal akurasi dan kecepatan dibandingkan versi-versi sebelumnya. YOLOv8 tersedia dalam beberapa varian berdasarkan ukuran model:

1. YOLOv8n (nano): Model terkecil dan tercepat, cocok untuk perangkat embedded dengan sumber daya terbatas
2. YOLOv8s (small): Keseimbangan antara ukuran dan akurasi
3. YOLOv8m (medium): Akurasi lebih tinggi dengan ukuran model yang lebih besar
4. YOLOv8l (large): Model besar dengan akurasi tinggi
5. YOLOv8x (extra large): Model terbesar dengan akurasi tertinggi

Untuk implementasi pada perangkat embedded seperti Raspberry Pi, YOLOv8n merupakan pilihan yang paling sesuai karena memiliki ukuran model yang kecil dan kecepatan inferensi yang tinggi dengan tetap mempertahankan akurasi yang memadai.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

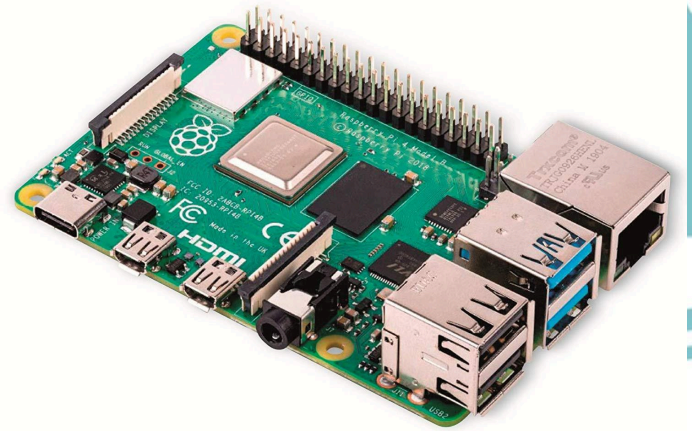
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.6 Raspberry Pi 4 Model B

Raspberry Pi 4 Model B merupakan single-board computer (SBC) yang dikembangkan oleh Raspberry Pi Foundation. Perangkat ini memiliki spesifikasi yang cukup untuk menjalankan aplikasi computer vision dan deep learning inference (Florentinus Budi Setiawan et al., 2021).

Spesifikasi Raspberry Pi 4 Model B:

1. Processor: Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
2. RAM: 4GB
3. Konektivitas:
    - Dual-band 802.11ac wireless (2.4 GHz dan 5 GHz)
    - Bluetooth 5.0
    - Gigabit Ethernet
    - 2 × USB 3.0 ports
    - 2 × USB 2.0 ports
4. GPIO: 40-pin GPIO header
5. Display: 2 × micro-HDMI ports (mendukung hingga 4K resolution)
6. Power: USB-C power input (5V/3A)



Gambar 2.6 Raspberry Pi 4 Model B
(Sumber : www.raspberrypi.com)

2.7 Raspberry Pi Camera Module

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Raspberry Pi Camera Module merupakan modul kamera resmi yang dikembangkan oleh Raspberry Pi Foundation dan dirancang untuk terhubung langsung dengan perangkat Raspberry Pi melalui antarmuka CSI (*Camera Serial Interface*). Modul kamera ini menggunakan sensor gambar OV5647 yang diproduksi oleh OmniVision Technologies, yaitu sensor CMOS berwarna dengan resolusi 5 megapiksel yang mampu menghasilkan citra dengan resolusi maksimal 2592×1944 piksel (OmniVision Technologies, 2012).

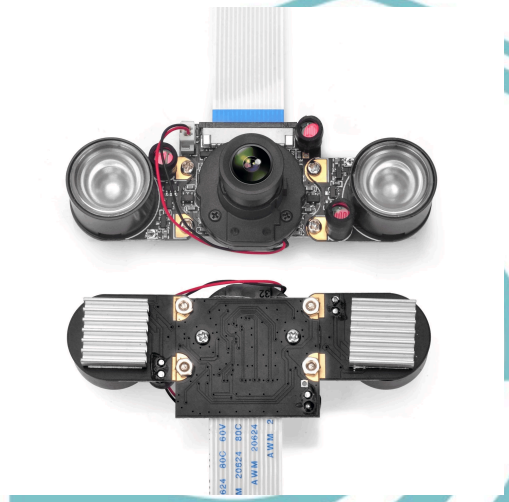
Sensor OV5647 mendukung berbagai mode output citra, termasuk format RAW, YUV, dan RGB, serta mampu merekam video pada resolusi 1080p dengan kecepatan 30 *frames per second* (fps) maupun 720p pada 60 fps. Sensor ini menggunakan teknologi *fixed focus lens* yang berarti jarak fokus kamera bersifat tetap dan tidak dapat disesuaikan secara otomatis. Karakteristik *fixed focus* ini menjadikan kamera sensitif terhadap perubahan jarak antara lensa dan objek, sehingga objek yang berada di luar rentang fokus optimal akan menghasilkan citra yang buram (*blur*) (OmniVision Technologies, 2012).

Dalam konteks penelitian ini, karakteristik *fixed focus* pada OV5647 memiliki implikasi langsung terhadap kualitas akuisisi citra hama mikro. Hama *whitefly* dan *winged aphid* yang berukuran 1–3 milimeter membutuhkan jarak fokus yang sangat dekat untuk menghasilkan detail citra yang tajam. Oleh karena itu, modul kamera OV5647 pada penelitian ini dikombinasikan dengan lensa makro eksternal yang dipasangkan pada kamera smartphone sebagai sumber data tambahan, guna menghasilkan citra *close-up* beresolusi tinggi yang diperlukan untuk pelatihan model YOLOv8n. Selain itu, sifat *fixed focus* pada OV5647 juga menjadi salah satu faktor penyebab munculnya *False Positive* pada hasil pengujian, dimana pergeseran jarak kamera akibat getaran menyebabkan efek *blur* yang membuat tekstur daun diinterpretasikan secara keliru sebagai koloni hama oleh model (Raspberry Pi Ltd, 2024).

Berikut merupakan spesifikasi lengkap Raspberry Pi Camera Module OV5647t:

1. Sensor: OmniVision OV5647
2. Resolusi: 5 megapiksel (2592×1944 piksel)

3. Format video: 1080p30, 720p60, 640×480p60/90
4. Antarmuka: CSI (*Camera Serial Interface*)
5. Dimensi: 25mm × 24mm × 9mm
6. Tipe fokus: *Fixed focus*
7. Sudut pandang: 54° × 41°



Gambar 2.7 Raspi Camera Module OV5647
(Sumber : osoyoo.store)

2.8 NodeMCU ESP8266

NodeMCU merupakan platform IoT yang bersifat open source. Platform ini terdiri dari hardware dan juga software. Perangkat keras NodeMCU menggunakan System on Chip (SoC) ESP8266 dari Espressif Systems. SoC ini memiliki fitur-fitur yang lengkap untuk IoT, termasuk mikrokontroler, kemampuan WiFi, dan chip komunikasi USB to serial.

ESP8266 NodeMCU memiliki kemampuan pemrosesan on-board yang kuat dan penyimpanan yang memadai yang memungkinkannya untuk diintegrasikan dengan pengembangan minimal di awal dan pemuatan minimal selama runtime melalui GPIO (General Purpose input/output) dengan sensor perangkat tertentu (Parihar 2019). Selain itu, ESP8266 memiliki biaya yang rendah dan fitur yang tinggi sehingga membuatnya menjadi modul yang ideal dalam proyek ini.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Software dari NodeMCU menggunakan bahasa pemrograman Lua. Bahasa Lua sendiri juga memiliki komunitas pengguna yang besar, sehingga developer dan user dapat dengan mudah menemukan bantuan jika ada terdapat masalah. Adapun spesifikasi NodeMCU ESP8266 V3 yang digunakan dalam proyek ini:

1. Ukuran Board: 57 mm x 30 mm
2. Tegangan Input: 3.3 ~ 5V
3. GPIO: 13 PIN
4. Kanal PWM: 10 Kanal
5. 10 bit ADC Pin: 1 Pin
6. Flash Memory: 4 MB
7. Clock Speed: 40/26/24 MHz
8. WiFi: IEEE 802.11 b/g/n
9. Frekuensi: 2.4 GHz
10. USB Port: Micro USB
11. USB to Serial Converter: CH340G

Gambar 2.8 NodeMCU ESP8266
(Sumber : cityos-air.readme.io)

2.9 Stepper Motor Nema 17

Stepper motor merupakan jenis motor listrik yang mengubah sinyal pulsa digital menjadi perpindahan sudut diskrit yang presisi. Berbeda dengan motor DC konvensional yang berputar secara kontinu, stepper motor bergerak dalam

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

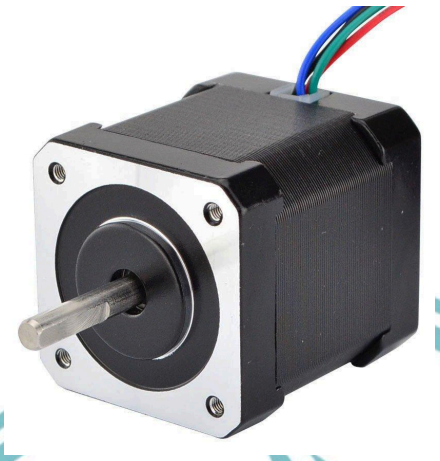
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

langkah-langkah tetap sehingga memungkinkan kontrol posisi yang akurat tanpa memerlukan sensor umpan balik tambahan (Laktionov & Yanko, 2025).

Nema 17 merupakan salah satu varian stepper motor yang paling banyak digunakan dalam sistem robotik dan otomasi skala kecil hingga menengah. Penamaan "Nema 17" mengacu pada standar dimensi faceplate motor sebesar 1,7 inci × 1,7 inci (42,3 mm × 42,3 mm) yang ditetapkan oleh National Electrical Manufacturers Association (NEMA). Motor ini beroperasi dengan prinsip bipolar dua fase, dimana setiap langkah penuh setara dengan perpindahan sudut sebesar 1,8° sehingga dibutuhkan 200 langkah untuk menyelesaikan satu putaran penuh (360°) (Laktionov & Yanko, 2025).

Dalam penelitian ini, Nema 17 digunakan sebagai aktuator mekanis utama pada sistem gantry untuk menggerakkan pelat kamera sepanjang jalur pipa instalasi hidroponik. Motor ini dikendalikan oleh mikrokontroler ESP8266 melalui driver motor stepper A4988. Driver A4988 berfungsi sebagai pengkondisi sinyal daya yang mengubah perintah pulsa logika digital bertegangan rendah dari ESP8266 menjadi arus fasa bertegangan tinggi yang dibutuhkan untuk memutar lilitan internal motor stepper secara presisi. Driver ini mendukung mode microstepping hingga 1/16 langkah, yang memungkinkan pergerakan gantry yang lebih halus dan terkontrol.

Karakteristik termal Nema 17 perlu diperhatikan dalam implementasi sistem yang beroperasi secara kontinu. Laktionov & Yanko (2025) menunjukkan bahwa suhu operasional motor dipengaruhi oleh kondisi lingkungan eksternal, dimana peningkatan beban kerja secara langsung berkontribusi terhadap kenaikan suhu inti motor. Pada penelitian ini, motor dioperasikan secara intermiten mengikuti siklus pemindaian gantry sehingga suhu operasional tetap berada dalam batas aman.



Gambar 2.9 Stepper Motor Nema 17
(Sumber : www.amazon.com.mx)

2.10 Timing Belt GT2

Timing belt merupakan komponen transmisi mekanis bergerigi yang berfungsi mengubah gerak rotasi motor menjadi gerak translasi linear pada sistem gantry. Berbeda dengan belt konvensional, timing belt menggunakan profil gigi yang berkaitan langsung dengan puli bergigi sehingga tidak terjadi slip antara belt dan puli, menghasilkan perpindahan posisi yang presisi dan dapat diprediksi secara matematis (Patterson, 2018). Jenis yang umum digunakan pada sistem gantry berbasis stepper motor adalah GT2 (*Gates Tooth 2mm pitch*), dengan jarak antargigi sebesar 2 mm yang memberikan resolusi pergerakan linear sebesar 2 mm per gigi puli per putaran motor.

Dalam penelitian ini, timing belt GT2 digunakan sebagai komponen transmisi gerak pada sistem gantry untuk memobilisasi pelat kamera sepanjang rel instalasi hidroponik. Rotasi dari motor stepper Nema 17 ditransmisikan melalui puli bergigi yang terpasang pada poros motor, menggerakkan belt secara linear sehingga pelat kamera bergeser secara horizontal mengikuti jalur pipa tray tanaman bayam.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Gambar 2.10 Timing Belt
(Sumber : robocraze.com)

2.11 Power Supply 12V

Power Supply Unit (PSU) 12V merupakan komponen catu daya yang berfungsi mengkonversi tegangan listrik AC dari jala-jala PLN menjadi tegangan DC stabil sebesar 12V yang dibutuhkan oleh komponen aktuator pada sistem. PSU yang digunakan pada penelitian ini merupakan jenis *Switching Mode Power Supply* (SMPS), yaitu catu daya yang memanfaatkan teknik pensaklaran semikonduktor frekuensi tinggi untuk menghasilkan regulasi tegangan yang efisien dengan disipasi panas yang lebih rendah dibandingkan catu daya linier konvensional (Laktionov & Yanko, 2025). Motor stepper Nema 17 membutuhkan tegangan operasional sebesar 12V dengan kebutuhan arus minimal 2A untuk dapat menghasilkan torsi puncak secara optimal pada sistem gantry.

Dalam penelitian ini, PSU 12V digunakan sebagai sumber daya utama untuk menyuplai tegangan VMOT pada driver motor A4988 guna menggerakkan motor stepper Nema 17. Pemisahan jalur catu daya antara PSU 12V untuk motor dan catu daya 5V untuk Raspberry Pi dilakukan untuk menghindari interferensi noise elektrik dari beban induktif motor stepper terhadap stabilitas komputasi perangkat edge.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Gambar 2.10 PSU 12V
(Sumber : www.circuitspecialists.com)

2.12 Driver Stepper Motor A4988

Driver stepper motor merupakan komponen elektronik yang berfungsi sebagai pengkondisi sinyal daya antara mikrokontroler dan motor stepper. Mikrokontroler seperti ESP8266 hanya mampu menghasilkan sinyal logika digital bertegangan rendah, sehingga tidak dapat secara langsung menggerakkan lilitan internal motor stepper yang membutuhkan arus tinggi. Driver motor berperan dalam mengkonversi sinyal logika tersebut menjadi arus fasa bertegangan tinggi yang dibutuhkan untuk memutar rotor motor secara presisi (Fecko, Vince & Darák, 2020).

A4988 merupakan salah satu driver stepper motor yang paling banyak digunakan dalam proyek robotik dan otomasi berbasis embedded system. Driver ini memiliki total 16 pin dan kompatibel dengan berbagai jenis motor stepper bipolar termasuk Nema 17 (Fecko, Vince & Darák, 2020). Konfigurasi tegangan pada A4988 terbagi menjadi dua jalur terpisah, yaitu VDD yang berfungsi sebagai sumber daya logika internal dengan rentang tegangan 3V hingga 5.5V, serta VMOT yang merupakan sumber daya untuk menggerakkan motor dengan rentang tegangan 8V hingga 35V.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Salah satu keunggulan utama A4988 merupakan kemampuan microstepping, yaitu kemampuan untuk menghasilkan langkah-langkah perpindahan yang lebih kecil dari langkah penuh motor. Fitur ini dicapai dengan menyuplai lilitan motor menggunakan level arus menengah. Konfigurasi microstepping diatur melalui kombinasi logika pada pin MS1, MS2, dan MS3, yang memungkinkan resolusi mulai dari full-step hingga 1/16 step. Semakin kecil ukuran langkah yang digunakan, semakin halus dan presisi pergerakan motor yang dihasilkan (Fecko, Vince & Darák, 2020).

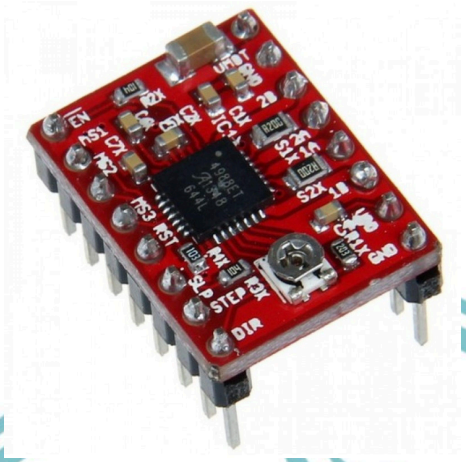
Kontrol arah dan kecepatan rotasi motor dilakukan melalui dua pin utama, yaitu pin STEP dan pin DIR. Pin STEP menerima sinyal pulsa dari mikrokontroler dimana setiap pulsa HIGH menggerakkan motor satu langkah sesuai dengan konfigurasi microstepping yang aktif, sehingga semakin cepat frekuensi pulsa yang dikirimkan maka semakin cepat putaran motor. Pin DIR menentukan arah rotasi, dimana nilai HIGH memutar motor searah jarum jam dan nilai LOW memutar motor berlawanan arah jarum jam (Fecko, Vince & Darák, 2020).

Dari sisi kapasitas arus, A4988 mampu menyuplai hingga 2A per coil, namun tanpa pendingin tambahan chip ini hanya mampu beroperasi secara stabil pada 1A per coil. Untuk mencegah kerusakan akibat panas berlebih pada operasi arus tinggi, diperlukan pemasangan heatsink pada permukaan chip. Selain itu, driver ini dilengkapi pin SLP yang berfungsi mengaktifkan mode tidur (*sleep mode*) untuk meminimalkan konsumsi daya saat motor tidak beroperasi (Fecko, Vince & Darák, 2020).

Dalam penelitian ini, A4988 digunakan sebagai driver penggerak motor stepper Nema 17 pada sistem gantry yang dikendalikan oleh mikrokontroler ESP8266. Sinyal pulsa dari ESP8266 dikirimkan ke pin STEP untuk mengatur pergerakan gantry plate, sementara pin DIR mengatur arah pergerakannya sepanjang jalur pipa instalasi hidroponik.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



Gambar 2.11 Driver Stepper Motor A4988
(Sumber : www.majju.pk)

2.13 Limit Switch

Limit switch merupakan komponen sensor yang berfungsi mendeteksi keberadaan atau posisi suatu objek berdasarkan kontak fisik langsung. Sensor ini bekerja dengan prinsip sederhana yaitu ketika sebuah objek bergerak menyentuh aktuator mekanis pada limit switch, kontak internal switch akan berubah kondisinya dari *normally open* (NO) menjadi tertutup, atau dari *normally closed* (NC) menjadi terbuka, sehingga menghasilkan sinyal digital yang dapat dibaca oleh mikrokontroler. Limit switch banyak digunakan dalam sistem otomasi industri maupun skala kecil karena karakteristiknya yang sederhana, andal, dan hemat biaya (Fecko, Vince & Darák, 2020).

Dalam sistem otomasi berbasis motor stepper, limit switch berperan sebagai perangkat keselamatan mekanis (*hardware interlock*) yang membatasi jangkauan pergerakan aktuator. Motor stepper yang dikendalikan secara *open-loop* tidak memiliki mekanisme umpan balik posisi secara mandiri, sehingga tanpa adanya sensor pembatas, motor dapat terus berputar melampaui batas fisik sistem dan menyebabkan kerusakan struktural. Dengan memasang limit switch pada titik batas pergerakan, sistem dapat secara otomatis menghentikan suplai pulsa ke driver motor pada saat aktuator mencapai titik akhir jalurnya (Fecko, Vince & Darák, 2020; Laktionov & Yanko, 2025).

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pada penelitian ini, limit switch dipasang pada kedua ujung rel gantry sebagai sensor batas pergerakan pelat kamera. Ketika pelat kamera yang digerakkan oleh motor stepper Nema 17 menyentuh limit switch di salah satu ujung rel, sinyal interupsi dikirimkan ke mikrokontroler ESP8266 untuk menghentikan suplai pulsa ke driver A4988 secara instan. Mekanisme ini memastikan pergerakan gantry berhenti tepat di titik akhir rel gantry dengan aman tanpa adanya resiko benturan fisik yang dapat merusak komponen sistem.

Gambar 2.12 Limit Switch
(Sumber : store.mainsci.com)

2.14 Flask

Flask merupakan web framework berbasis Python yang bersifat *lightweight* dan dikategorikan sebagai *microframework* karena tidak mengharuskan penggunaan library atau tools tertentu dalam pengembangannya. Flask pertama kali dikembangkan oleh Armin Ronacher dari organisasi Pallets Projects dan dirilis pada tahun 2010. Sejak saat itu, Flask berkembang menjadi salah satu web framework Python yang paling banyak digunakan, khususnya untuk pengembangan aplikasi web berskala kecil hingga menengah yang membutuhkan fleksibilitas tinggi (Pallets Projects, 2024).

Flask dibangun di atas dua komponen utama, yaitu Werkzeug sebagai toolkit WSGI (*Web Server Gateway Interface*) dan Jinja2 sebagai *template engine*.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Sebagai microframework, Flask tidak menyediakan lapisan abstraksi basis data, validasi formulir, maupun komponen bawaan lainnya secara default. Namun demikian, Flask mendukung mekanisme ekstensi yang memungkinkan penambahan fungsionalitas sesuai kebutuhan pengembang (Pallets Projects, 2024).

Dalam konteks sistem embedded dan IoT, Flask banyak dimanfaatkan sebagai web server lokal yang ringan pada perangkat seperti Raspberry Pi. Karakteristik Flask yang tidak membutuhkan dependensi besar menjadikannya cocok untuk dijalankan pada perangkat dengan sumber daya komputasi terbatas. Melalui Flask, sebuah perangkat embedded dapat menyajikan antarmuka berbasis HTTP yang dapat diakses oleh perangkat lain dalam satu jaringan lokal (*Local Area Network*) maupun melalui internet.

Pada penelitian ini, Flask diimplementasikan pada Raspberry Pi 4 sebagai web server utama yang menangani dua fungsi sekaligus secara asinkronus, yaitu menyajikan *live video stream* hasil inferensi model YOLOv8n NCNN kepada pengguna melalui *web browser*, serta mengelola transmisi paket data hasil deteksi hama menuju *cloud server*. Pemisahan jalur eksekusi (*decoupling*) antara proses inferensi dan proses transmisi data ini dimungkinkan oleh arsitektur Flask yang mendukung pemrosesan *request* secara non-blocking, sehingga aktivitas pengiriman data ke cloud tidak mengganggu kelancaran pipeline deteksi utama.

2.15 Roboflow

Roboflow merupakan platform berbasis *cloud* yang dirancang khusus untuk membantu pengembang dan peneliti dalam membangun dataset *computer vision* secara efisien. Platform ini menyediakan layanan terintegrasi yang mencakup anotasi citra, *preprocessing*, augmentasi data, manajemen dataset, hingga ekspor dataset dalam berbagai format yang kompatibel dengan framework *deep learning* populer seperti YOLOv8, TensorFlow, dan PyTorch (Kukartsev et al., 2024).

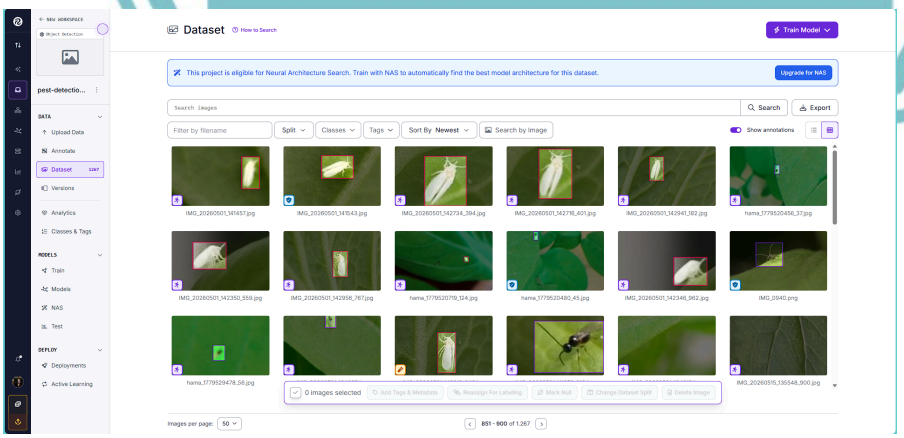
© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Salah satu fitur utama Roboflow merupakan kemampuan anotasi *bounding box* yang memungkinkan pengguna menandai objek target pada citra secara visual melalui antarmuka berbasis *web*. Setiap anotasi menyimpan informasi kelas objek beserta koordinat posisi *bounding box* dalam format yang secara langsung kompatibel dengan arsitektur input YOLOv8. Ultralytics selaku pengembang YOLOv8 secara resmi merekomendasikan Roboflow sebagai *tool* pelabelan yang kompatibel dengan ekosistem YOLOv8 karena kemudahan proses ekspor dataset ke format YOLO tanpa konversi manual (Kukartsev et al., 2024).

Selain anotasi, Roboflow menyediakan pipeline *preprocessing* dan augmentasi data yang dapat dikonfigurasi sesuai kebutuhan penelitian. Fitur *preprocessing* mencakup *auto-orient* untuk mengoreksi orientasi spasial gambar dan *resize* untuk menyeragamkan dimensi citra sesuai dengan standar input model. Sementara itu, fitur augmentasi memungkinkan penggambaran variasi visual baru dari setiap sampel citra asli melalui transformasi seperti *flip*, penyesuaian *brightness*, dan efek *blur*, sehingga meningkatkan variasi dataset dan kemampuan generalisasi model terhadap kondisi visual yang beragam di lingkungan nyata (Kukartsev et al., 2024).

Dalam penelitian ini, Roboflow digunakan sebagai platform utama untuk proses pelabelan dataset citra hama *whitefly* dan *winged aphid* pada daun tanaman bayam hidroponik, serta untuk menerapkan pipeline *preprocessing* dan augmentasi data sebelum dataset diekspor ke format YOLOv8 untuk proses pelatihan model.



Gambar 2.13 Gambar Aplikasi Roboflow

(Sumber : app.roboflow.com)

2.16 Google Colab

Google Colab merupakan layanan *cloud computing* berbasis *Jupyter Notebook* yang dikembangkan oleh Google dan dapat diakses secara gratis melalui *web browser*. Platform ini dirancang khusus untuk mendukung kegiatan riset dan pengembangan di bidang *machine learning* dan *deep learning* yang menyediakan akses terhadap sumber daya komputasi bertenaga tinggi tanpa memerlukan konfigurasi perangkat keras lokal (Carneiro et al., 2018).

Keunggulan utama Google Colab terletak pada ketersediaan akselerasi GPU secara gratis yang dapat dimanfaatkan untuk mempercepat proses pelatihan model *deep learning*. Google Colab menyediakan GPU NVIDIA Tesla T4 dengan arsitektur Turing yang memiliki 2.560 CUDA *cores*, memori GPU sebesar 16 GB GDDR6, dan performa puncak sebesar 8,1 TFLOPS, serta RAM sistem sebesar 12,7 GB (Carneiro et al., 2018). Spesifikasi ini jauh melampaui kemampuan komputasi perangkat *embedded* seperti Raspberry Pi 4, sehingga menjadikan Google Colab sebagai solusi yang tepat untuk proses pelatihan model yang membutuhkan komputasi intensif.

Selain akselerasi GPU, Google Colab mendukung berbagai *library* dan framework *deep learning* populer yang dapat langsung digunakan tanpa instalasi manual, termasuk PyTorch, TensorFlow, dan Ultralytics YOLOv8. Lingkungan eksekusi Colab yang berbasis Python memungkinkan integrasi langsung dengan platform seperti Roboflow untuk mengimpor dataset dan menjalankan pipeline pelatihan model secara end-to-end dalam satu *notebook* (Carneiro et al., 2018).

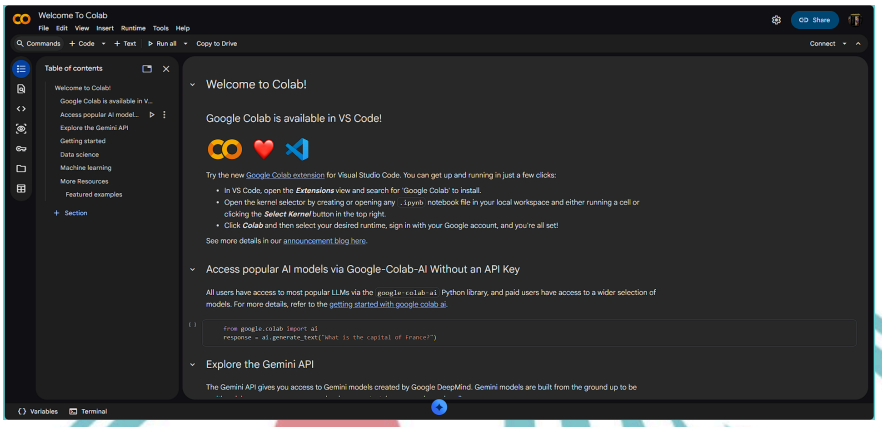
Dalam penelitian ini, Google Colab digunakan sebagai platform pelatihan model YOLOv8n dengan memanfaatkan akselerasi GPU Tesla T4. Pemilihan Google Colab didasarkan pada keterbatasan sumber daya komputasi perangkat lokal untuk menjalankan proses pelatihan model *deep learning* dengan dataset berskala

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

ribuan citra, sementara proses inferensi sepenuhnya dijalankan secara *on-device* pada Raspberry Pi 4 setelah model dioptimasi ke format NCNN.



Gambar 2.13 Gambar Google Colab
(Sumber : colab.research.google.com)

2.17 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa setiap perangkat yang telah dibuat dapat bekerja sesuai harapan. Terdapat beberapa pengujian yang dilakukan terhadap sistem dan perangkat yang dibangun dalam penelitian, yaitu:

1. Analisis Hyperparameter Model

   Sebelum diimplementasikan pada perangkat *edge*, model hasil pelatihan harus divalidasi terlebih dahulu untuk memastikan kapabilitas generalisasinya terhadap objek baru. Evaluasi komparatif dengan membandingkan nilai akurasi *Mean Average Precision* (mAP) dan kecepatan komputasi (*inference speed*) merupakan metodologi standar yang digunakan secara luas untuk mengukur efisiensi arsitektur *object detection* (Terven et al., 2023). Pengukuran timbal-balik (*trade-off*) antara akurasi dan kecepatan ini krusial dilakukan, terutama untuk memastikan model mampu berjalan secara optimal pada perangkat komputasi bergerak atau sistem tertanam dengan sumber daya terbatas (Lazarevich et al., 2023).

   Parameter yang digunakan sebagai indikator performa dalam validasi model ini meliputi:

1. Mean Average Precision (mAP): Metrik standar dalam *object detection* untuk mengukur akurasi model dalam memprediksi kelas sekaligus melokalisasi koordinat objek. Nilai mAP dihitung dengan mengintegrasikan kurva *Precision-Recall*. Nilai mAP50 merepresentasikan rerata presisi ketika ambang batas persentase irisan kotak (*Intersection over Union*) berada pada nilai IoU ≥ 0.5 (Jocher et al., 2023).
2. Inference Latency (GPU Server): Durasi waktu komputasi yang dibutuhkan oleh unit pemroses grafis untuk menyelesaikan satu siklus *forward pass* dari suatu citra masukan (diukur dalam satuan milidetik / ms).

2. Pengujian Performa Deteksi Model

Evaluasi performa deteksi model tidak didasarkan pada klasifikasi biner konvensional, melainkan mengukur tingkat ketepatan lokasi *bounding box* dan klasifikasi objek berdasarkan nilai *Intersection over Union* (IoU) dengan *Ground Truth* (label asli). Klasifikasi objek dibagi menjadi tiga komponen:

1. True Positive (TP): Hama berhasil dideteksi oleh model dengan klasifikasi kelas yang benar dan nilai IoU ≥ 0.5.
2. False Positive (FP): Model mendeteksi adanya hama pada area yang salah, atau klasifikasi kelasnya keliru.
3. False Negative (FN): Hama yang tertera pada *ground truth* gagal dideteksi oleh model.

Berdasarkan komponen di atas, parameter tingkat ketepatan model diukur menggunakan tiga metrik utama standar Ultralytics YOLOv8:

1. Precision (Presisi): Mengukur sejauh mana tingkat kepercayaan model terhadap objek yang berhasil dideteksi.

   $$\text{Precision} = \frac{TP}{TP+FP} \times 100\%$$

2. Recall (Sensitivitas): Mengukur keberhasilan model dalam mendeteksi seluruh hama yang sebenarnya ada pada objek nyata.

   $$\text{Recall} = \frac{TP}{TP+FN} \times 100\%$$

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

3. F1-Score: Nilai rata-rata harmonis (*harmonic mean*) yang merepresentasikan keseimbangan performa antara metrik *Precision* dan *Recall*.

   F1-Score= $2x\frac{Precision\ x\ Recall}{Precision + Recall}$

Precision mewakili kemampuan model untuk mengenali objek yang relevan sebagai persentase dari prediksi yang benar, sementara Recall mewakili kemampuan model untuk menemukan semua objek yang relevan sebagai persentase dari True Positive yang dapat terdeteksi dari semua data referensi (Ground Truth). Sedangkan skor F1 merupakan rata-rata Harmonic dari Precision dan Recall (Fandisyah et al., 2021).

3. Pengujian Performa Perangkat

   Pengujian performa ditujukan untuk menganalisis efisiensi eksekusi program dan beban kerja perangkat keras (*hardware resource utilization*) dari perangkat *edge computing* Raspberry Pi 4 saat menjalankan model YOLOv8n teroptimasi NCNN. Parameter yang diukur dalam pengujian performa ini merujuk pada standar evaluasi sistem tertanam (Varghese *et al.*, 2019) yang meliputi:

   1. *Inference Latency*: Waktu yang dibutuhkan oleh inti prosesor untuk mengeksekusi satu bingkai citra gambar dari tahap *preprocessing*, *inference*, hingga *postprocessing* (diukur dalam satuan milidetik / ms).
   2. *Memory and CPU Usage*: Persentase alokasi beban kerja unit pemroses sentral (CPU) serta kapasitas memori akses acak (RAM) yang dikonsumsi selama *pipeline* deteksi berjalan secara kontinu.
   3. Suhu Operasional: Pengukuran fluktuasi suhu pada *System on Chip* (SoC) Broadcom BCM2711 menggunakan sensor termal internal berbasis berkas perintah /sys/class/thermal/. Parameter ini dipantau secara ketat untuk mengevaluasi efektivitas pendingin mekanis (*heatsink/fan*) dan menghindari terjadinya *thermal throttling* (>80℃) yang dapat mendegradasi frekuensi *clock* CPU secara drastis (Murshed *et al.*, 2019) .

4. Pengujian Fungsionalitas

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pengujian fungsionalitas dilakukan untuk memvalidasi kesesuaian antara fungsi mekanis-elektronis sistem yang dibangun dengan logika program yang telah dirancang pada tahap analisis kebutuhan. Pengujian dilakukan menggunakan metode *Black-Box Testing* yang berfokus penuh pada fungsionalitas input dan output tanpa harus memeriksa struktur kode internal perangkat lunak (Myers *et al.*, 2011).

5. Pengujian Reliabilitas Sistem

   Pengujian reliabilitas dilakukan untuk mengukur tingkat ketahanan, stabilitas, dan konsistensi sistem dalam beroperasi secara terus-menerus (*continuous monitoring*) dalam jangka waktu tertentu tanpa mengalami kegagalan fungsi (*system crash*). Berdasarkan standar rekayasa perangkat lunak, pengujian ini menerapkan metode *Stress Testing* dan *Durability Testing* (Weyuker dan Avritzer, 2002).

2.18 Penelitian Terdahulu

Berikut merupakan beberapa penelitian terdahulu yang relevan dengan topik deteksi hama pada tanaman hidroponik yang menjadi referensi dalam pembuatan sistem deteksi hama menggunakan metode YOLOv8n:

Tabel 2.1 Penelitian Terdahulu

No	Nama Peneliti	Judul	Hasil Analisis	Catatan
1	(Dinata, 2023)	Rancang Bangun Sistem Penyiraman Pestisida dan Deteksi Hama Menggunakan Metode YOLOv5	- Akurasi >88% - implementasi pada Raspberry Pi 4 - Real-time detection dengan webcam - Sistem penyiraman	- Single-class detection (hanya belalang) - Menggunakan YOLOv5 yang sudah bukan versi terbaru - Belum ada optimasi model (TFLite, quantization) -

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

			otomatis berdasarkan intensitas serangan	Inference time belum optimal
2	(Chen et al., 2020)	*An AIoT Based Smart Agricultural System for Pests Detection*	- Pest identification accuracy: 90% - mAP: 0.92 - Menggunakan UAV untuk image collection - Terintegrasi dengan 6 environmental sensors - LSTM untuk prediksi pest occurrence	- Single-class detection (hanya Tessaratoma papillosa) - Implementasi pada drone, bukan embedded device statis - Dataset terbatas (687 citra asli → 7.000 setelah augmentasi) - Belum ada optimasi untuk edge deployment dan masih menggunakan YOLOv3
3	(Dai et al., 2023)	A New Pest Detection Method Based on Improved YOLOv5m	Mengusulkan modifikasi YOLOv5m dengan SwinTransforme r pada neck (SWinTR) dan WConcat untuk meningkatkan feature fusion. Dibandingkan	- Menggunakan YOLOv5m, bukan YOLOv8 - Dataset umum hama tanaman, tidak spesifik aphid/whitefly pada bayam - Tidak ada implementasi pada edge device/embedded

			YOLOv3, YOLOv4, dan YOLOv5m standar pada dataset hama tanaman publik. Model yang diusulkan mencapai akurasi lebih tinggi dan lebih robust dibanding baseline.	
4	(Nugroho et al., 2025)	Implementation of Sparrow Pest Detection Using YOLOv8 Method on Raspberry Pi and Google Coral USB Accelerator	- Implementasi YOLOv8n pada Raspberry Pi 4 dan Google Coral USB TPU - Deteksi hama burung secara real-time - Efisiensi daya tinggi pada embedded device	- Single-class detection (hanya burung pipit) - Membutuhkan akselerator TPU tambahan (Google Coral) - Tidak spesifik tanaman hidroponik - Bukan hama mikro
5	(Hakim et al., 2025)	Yolo-pest: an optimized YoloV8x for detection of small insect	- Akurasi deteksi rata-rata 94% - Deteksi hama mikro termasuk whitefly	- Menggunakan YOLOv8x, bukan varian nano - Inferensi dijalankan di server, bukan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

		pests using smart traps	menggunakan IoT Smart Trap berbasis kamera - Validasi pada citra grayscale dan berwarna	on-device embedded - Berbasis perangkap pheromone, bukan sistem gantry kamera - Dataset tidak spesifik tanaman bayam hidroponik

Berdasarkan daftar penelitian terdahulu pada Tabel 2.7, belum ditemukan penelitian yang berfokus pada deteksi hama mikro (whitefly dan winged *aphid*) secara lokal (*on-device*) dengan mengkombinasikan optimasi model YOLOv8n ke format NCNN dan sistem mekanis *gantry* sekuensial. Maka dari itu, penelitian ini mengusulkan sebuah sistem *Edge IoT* berbasis Raspberry Pi 4, kamera makro, dan algoritma YOLOv8n yang teroptimasi NCNN untuk mengisolasi objek hama mikro pada tanaman hidroponik secara asinkronus, guna menekan *latency* komputasi lokal serta meminimalkan kehilangan detail fitur piksel akibat proses *downsampling*.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB III
METODE PENELITIAN

3.1 Rancangan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode *prototyping* yang berfokus pada perancangan, implementasi, dan evaluasi sistem deteksi hama tanaman bayam hidroponik. Sistem ini dikembangkan berbasis citra digital menggunakan algoritma YOLOv8n yang dijalankan secara *on-device* pada perangkat Raspberry Pi 4. Metode *prototyping* dipilih karena memungkinkan pengembang untuk membangun, menguji, dan menyempurnakan sistem secara iteratif berdasarkan hasil evaluasi nyata di lapangan, sehingga setiap komponen perangkat keras maupun perangkat lunak dapat dioptimalkan sebelum sistem final ditetapkan (Pressman & Maxim, 2014).

Metode *prototyping* dalam penelitian ini diterapkan melalui siklus evolusioner untuk mengatasi ketidakpastian karakteristik objek di lapangan, seperti variasi intensitas cahaya alami *greenhouse* dan ukuran piksel hama mikro (*whitefly* dan *winged aphid*). Menurut Pressman dan Maxim (2014), *prototyping* yaitu menghasilkan visualisasi atau model kerja awal sistem sebagai alat bantu komunikasi dan evaluasi taktis sebelum sistem di-*deploy* secara permanen. Pengembang dapat mengidentifikasi kelemahan model komputasi atau kegagalan mekanis gantry sejak dini, kemudian melakukan perbaikan arsitektur sistem atau penyesuaian parameter model pada iterasi berikutnya. Pendekatan ini meminimalkan risiko kegagalan sistemik saat model dijalankan secara mandiri (*standalone edge deployment*) pada perangkat dengan spesifikasi komputasi terbatas seperti Raspberry Pi 4.

Pendekatan kuantitatif diterapkan dalam tahap evaluasi sistem karena kinerja komponen diukur menggunakan data numerik yang bersifat objektif, deterministik, dan terukur. Parameter pengukuran fungsional meliputi akurasi deteksi (Precision, Recall, F1-Score) serta metrik performa komputasi *hardware* yang mencakup waktu inferensi (*inference latency*), utilisasi CPU, konsumsi RAM, dan suhu operasional prosesor pada Raspberry Pi 4. Hasil pengukuran

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

numerik tersebut kemudian dianalisis secara statistik untuk menilai efektivitas serta efisiensi dari optimasi *framework* NCNN pada sistem deteksi hama mikro yang dikembangkan.

3.2 Tahapan Penelitian

Tahapan penelitian ini disusun berdasarkan siklus metode *prototyping* yang bersifat iteratif dan dikombinasikan dengan pendekatan kuantitatif. Penelitian kuantitatif menekankan pada proses pengukuran variabel secara objektif serta analisis data numerik untuk mengevaluasi kinerja suatu sistem (Sarwono & Handayani, 2021). Dalam konteks ini, data numerik hasil pengujian kinerja sistem bertindak sebagai indikator validasi kelayakan *prototype*. Apabila hasil evaluasi pada suatu iterasi tidak memenuhi batas ambang (*threshold*) parameter performa yang ditetapkan, proses pengembangan akan dikembalikan ke tahapan perencanaan atau pemodelan untuk diperbaiki kembali secara iteratif (Pressman & Maxim, 2014).

Tahapan penelitian yang dilakukan meliputi beberapa langkah fungsional sebagai berikut:

1. *Communication*

   Penelitian diawali dengan observasi langsung pada instalasi hidroponik serta wawancara dengan praktisi budidaya untuk mengidentifikasi permasalahan yang dihadapi dalam proses pemantauan hama. Pada tahap ini dilakukan identifikasi jenis hama yang paling sering menyerang tanaman bayam(*Amaranthus sp.*) hidroponik, gejala serangan pada tanaman, serta kebutuhan pengguna terhadap sistem deteksi hama secara otomatis sebagai dasar pengembangan prototype.

2. *Quick Plan*

   Berdasarkan hasil analisis kebutuhan di lapangan, disusun rencana kerja cepat yang mencakup:

   1. Penentuan perangkat keras berupa Raspberry Pi 4 dan modul kamera yang digunakan sebagai platform edge computing.
   2. Penentuan arsitektur *deep learning* ringan yang kompatibel untuk *edge device* (YOLOv8-nano).

3. Penentuan metode optimasi graf komputasi dan kuantisasi data via *framework* Tencent NCNN.
4. Perancangan jadwal pengumpulan dataset citra mandiri (*custom dataset*) di lapangan.

3. *Modeling Quick Design*

Tahap ini berfokus pada pembuatan cetak biru (*blueprint*) sistem yang mencakup dua domain utama:

1. Perangkat Keras: Perancangan konfigurasi perangkat keras yang terdiri atas Raspberry Pi 4, sensor kamera (CSI/USB), serta modul catu daya penunjang gantry.
2. Perangkat Lunak: Pemodelan diagram alur (*flowchart*) sistem, mulai dari *image acquisition*, preprocessing citra, inferensi model NCNN, hingga kalkulasi *bounding box* koordinat objek.

4. *Construction of Prototype*

Realisasi rancangan ke dalam bentuk fisik dan kode program dijalankan pada tahap ini melalui proses:

1. Pengumpulan Dataset: Pengumpulan dataset citra tanaman bayam hidroponik dilakukan secara mandiri pada lokasi penelitian dengan variasi sudut pengambilan gambar. Selanjutnya dilakukan proses anotasi atau pelabelan objek menggunakan *bounding box*, pembagian dataset ke dalam data pelatihan, validasi, dan pengujian (*train/validation/test split*), serta augmentasi citra untuk meningkatkan variasi data dan kemampuan generalisasi model.
2. Training dan Optimisasi Model: Model YOLOv8n dilatih menggunakan framework PyTorch pada komputer yang dilengkapi GPU. Kemudian dikonversi ke format NCNN (*.param* dan *.bin*) sebagai framework inferensi ringan pada Raspberry Pi 4. Optimasi dilakukan melalui proses konversi model dan pemanfaatan instruksi ARM NEON agar proses inferensi pada perangkat *edge* menjadi lebih efisien.
3. Implementasi Perangkat: Raspberry Pi 4, modul kamera, catu daya, serta komponen pendukung lainnya dirakit menjadi satu kesatuan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

sistem. Selanjutnya dilakukan integrasi antara perangkat keras dengan model YOLOv8n yang telah dioptimasi menggunakan NCNN sehingga prototype mampu melakukan proses deteksi hama secara *on-device*.

5. *Deployment, Delivery & Feedback*

   Prototype sistem dioperasikan secara penuh di area instalasi hidroponik untuk melakukan pengujian inferensi *on-device* secara langsung. Pada tahap ini, pengujian kuantitatif dilakukan dengan mengumpulkan data numerik performa model dan efisiensi daya *hardware*. Hasil pengujian dianalisis secara kritis:

   1. Jika nilai akurasi atau kecepatan inferensi (*latency*) belum memenuhi batas kelayakan akibat kendala *noise* visual atau *bottleneck* komputasi CPU, data kegagalan tersebut dijadikan umpan balik (*feedback*) untuk kembali ke tahap Komunikasi Awal atau Perencanaan Cepat guna mengevaluasi dataset atau tingkat kuantisasi data.
   2. Jika seluruh parameter numerik telah memenuhi target, *prototype* dinyatakan sukses dan ditetapkan sebagai sistem final.

3.3 Objek Penelitian

Penelitian ini akan mengambil fokus pada penerapan teknologi YOLOv8 sebagai pendeteksi hama pada pertanaman bayam di Lokatani. Pemilihan jenis hama yang diteliti, yaitu kutu kebul (*whitefly*) dan kutu daun (*aphids*), didasarkan pada literatur yang mengidentifikasi kedua hama tersebut sebagai organisme pengganggu utama pada tanaman bayam (Mureithi, Fiaboe, Ekesi & Meyhöfer, 2017; Seni, 2018). Proses penelitian dilakukan dalam rentang durasi yang telah ditentukan, dan kemudian hasil penelitian ini akan dianalisa berdasarkan performa dan kinerja sistemnya.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Gambar 3.1 Hama Pada Tanaman di Lokatani

Kedua jenis hama ini termasuk dalam kategori hama dengan ukuran tubuh yang sangat kecil, berkisar antara 1 hingga 3 milimeter. Karakteristik fisik yang kecil ini menjadikan proses deteksi berbasis citra menjadi tantangan tersendiri karena hama sulit terlihat pada resolusi gambar biasa dan rentan hilang akibat efek *downsampling* pada proses pengolahan citra. Oleh karena itu, penggunaan kamera makro dalam sistem ini menjadi krusial untuk memastikan detail fitur visual hama dapat ditangkap dengan cukup tajam agar lapisan konvolusi model YOLOv8n dapat mempelajari karakteristik hama secara optimal.

Proses penelitian dilakukan dalam rentang durasi yang telah ditentukan, dan hasil penelitian dianalisis berdasarkan performa dan kinerja sistem yang dikembangkan.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB IV
HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengidentifikasi seluruh komponen pembangun sistem, baik dari segi data, perangkat lunak (*software*), maupun perangkat keras (*hardware*). Hal ini krusial agar model *deep learning* YOLOv8n dapat berjalan secara optimal dan *real-time* pada perangkat *edge computing* Raspberry Pi 4. Berdasarkan pendekatan tersebut, analisis kebutuhan sistem diklasifikasikan menjadi dua aspek utama, yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

4.1.1 Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan layanan, fitur, atau fungsi spesifik yang wajib disediakan oleh sistem saat dioperasikan di lapangan. Pada sistem deteksi hama ini, kebutuhan fungsional meliputi:

Tabel 4.1 Kebutuhan Penelitian

ID	Fungsionalitas	Deskripsi
F-01	Akuisisi Citra Digital	Sistem harus mampu menangkap aliran citra (*video stream*) secara *real-time* dari area kanopi tanaman bayam hidroponik menggunakan modul kamera yang terintegrasi.
F-02	Preprocessing Citra	Sistem harus mampu melakukan *resizing* dimensi citra secara otomatis, serta melakukan normalisasi warna sebelum citra diumpankan ke model inferensi.
F-03	Inferensi Lokal	Sistem harus mampu memuat graf memori model YOLOv8n-NCNN (.param dan .bin) ke dalam RAM lokal Raspberry Pi 4 dan mengeksekusi proses

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

		deteksi tanpa ketergantungan pada jaringan internet atau *cloud server*.
F-04	Klasifikasi dan Lokalisasi Multi-Hama	Sistem harus mampu mendeteksi dan membedakan dua kelas objek hama mikro secara simultan, yaitu kutu kebul (*whitefly*) dan kutu daun bersayap (winged aphid), dengan menandai koordinat objek menggunakan *bounding box*.
F-05	Ekstraksi Metrik Performa	Sistem harus mampu mencatat dan menampilkan koordinat piksel deteksi, label kelas hama, nilai *confidence score* hama yang terdeteksi pada setiap *frame*.

4.1.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional mendefinisikan batasan properti, kualitas teknis, kriteria performa, dan atribut perilaku yang harus dipenuhi agar sistem dapat bekerja secara efisien pada lingkungan *embedded*. Kebutuhan non-fungsional dalam penelitian ini meliputi:

Tabel 4.2 Kebutuhan Penelitian

ID	Fungsionalitas	Deskripsi
NF-01	Performa Komputasi	Proses inferensi lokal pada CPU Raspberry Pi 4 menggunakan *framework* NCNN ditargetkan memiliki *latency* yang rendah guna menghindari penumpukan antrian memori (*memory bottleneck*).

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

NF-02	Akurasi Model	Model YOLOv8n setelah proses kuantisasi harus tetap mempertahankan nilai *Mean Average Precision* (mAP_{50}) yang tinggi untuk objek berukuran mikro
NF-03	Efisiensi Sumber Daya	Alokasi penggunaan memori RAM selama skrip deteksi berjalan tidak boleh melebihi kapasitas memori fisik Raspberry Pi 4 (maksimal 70% dari kapasitas RAM tersedia) agar sistem operasi tetap stabil.
NF-04	Stabilitas Termal	Kompilasi instruksi ARM NEON pada NCNN harus diimbangi dengan manajemen suhu (menggunakan *heatsink* atau *cooling fan*) agar suhu operasional prosesor tidak menyentuh batas *thermal throttling* (>80) yang dapat mendegradasi kecepatan inferensi.
NF-05	Portabilitas dan Keandalan	Perangkat keras harus mampu beroperasi secara terus-menerus (*continuous monitoring*) di area instalasi hidroponik terbuka dengan toleransi terhadap fluktuasi intensitas cahaya alami.

4.2 Perancangan Sistem

4.2.1 Perancangan Perangkat Keras dan Perangkat Lunak

Perancangan sistem pada penelitian ini mencakup penentuan kebutuhan perangkat keras (*hardware*) dan perangkat lunak (*software*) yang digunakan untuk mendukung proses pengembangan, implementasi, serta pengujian sistem deteksi hama berbasis YOLOv8n-NCNN pada Raspberry Pi 4. Seluruh kebutuhan perangkat yang digunakan dalam penelitian ini disajikan pada Tabel 4.3.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 4.3 Kebutuhan Penelitian

Perangkat Keras	
Raspberry Pi 4 Model B	Bertindak sebagai unit pemroses utama untuk menjalankan skrip Python Flask web server, mengeksekusi inferensi model YOLOv8n format Tencent NCNN secara lokal, serta mengelola aliran data ke cloud.
Raspberry Pi Camera	Berfungsi sebagai sensor akuisisi citra optik untuk menangkap gambar permukaan daun bayam yang digunakan untuk mendeteksi hama pada permukaan daun tanaman bayam..
ESP8266	Mikrokontroler pengontrol gerakan motor yang menerima instruksi dari Raspberry Pi sebelum beroperasi.
Driver Stepper Motor A4988	Berfungsi sebagai pengkondisi sinyal daya yang mengubah perintah pulsa logika digital rendah dari ESP8266 menjadi arus fasa tinggi yang dibutuhkan untuk memutar lilitan internal motor stepper secara presisi.
Limit Switch	Berfungsi sebagai sensor keselamatan untuk mendeteksi batas ujung rel *gantry*, guna memicu interupsi program agar motor berhenti dan mencegah pelat kamera menabrak struktur fisik instalasi.
Nema 17 Stepper Motor	Bertindak sebagai aktuator mekanis utama untuk menggerakkan sabuk transmisi (*timing belt*) yang memobilisasi pelat kamera di sepanjang rel gantry.
Perangkat Lunak	

Visual Studio Code	Digunakan sebagai lingkungan pengembangan (IDE) utama untuk penulisan, pengeditan, dan debugging seluruh kode program sistem, baik skrip Python pada Raspberry Pi maupun kode Arduino pada ESP8266.
Python	Bahasa pemrograman utama yang digunakan untuk mengembangkan pipeline inferensi YOLOv8n NCNN, Flask web server, serta modul transmisi data ke cloud pada Raspberry Pi 4.
OpenCV	Sebagai pustaka pengolahan citra untuk akuisisi frame dari kamera, preprocessing berupa resize ke format 640×640 piksel, serta visualisasi hasil inferensi berupa bounding box dan label kelas secara real-time.
Ultralytics YOLOv8	Framework deep learning yang digunakan untuk proses pelatihan model deteksi hama pada Google Colab, serta konversi model ke format NCNN untuk deployment pada Raspberry Pi 4.
NCNN	Framework inferensi jaringan saraf tiruan dari Tencent yang digunakan untuk mengeksekusi model YOLOv8n secara efisien pada prosesor ARM Raspberry Pi 4 tanpa ketergantungan library eksternal.
Flask	Web framework Python yang digunakan sebagai server utama untuk menyajikan live video stream hasil inferensi kepada pengguna melalui web browser serta mengelola transmisi data ke cloud secara asinkronus.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Roboflow	Platform yang digunakan untuk proses pelabelan dataset citra hama, preprocessing, augmentasi data, serta ekspor dataset ke format YOLOv8 untuk pelatihan model.
Google Colab	Platform yang digunakan untuk proses pelatihan model YOLOv8n dengan akselerasi GPU Tesla T4, serta konversi model ke format NCNN yang digunakan untuk deployment pada Raspberry Pi 4., mengingat keterbatasan sumber daya komputasi perangkat lokal.

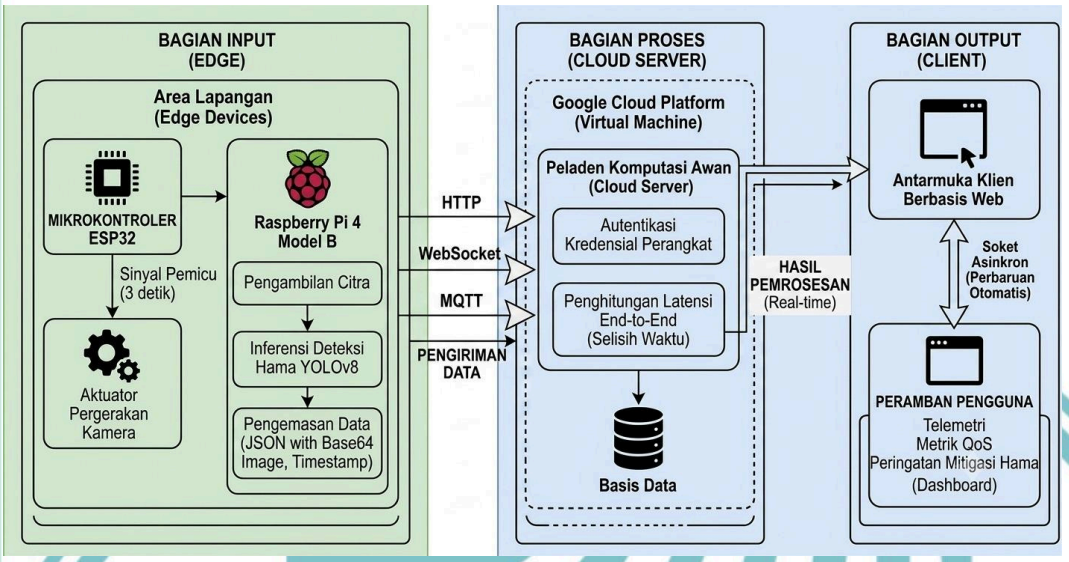
4.2.2 Diagram Blok

Gambar 4.1 menunjukkan diagram blok dari sistem yang dibangun. Sistem menerima input berupa citra daun tanaman bayam hidroponik yang diperoleh melalui modul kamera OV5647 yang terhubung langsung dengan Raspberry Pi 4 melalui antarmuka CSI.

Citra yang diperoleh dari kamera selanjutnya diproses pada tahap pemrosesan. Pada tahap ini, citra terlebih dahulu melalui proses akuisisi dan pra-pemrosesan menggunakan OpenCV berupa konversi format dan resize dimensi menjadi 640×640 piksel sesuai standar input model YOLOv8n. Setelah itu, citra diproses menggunakan model YOLOv8n yang telah dioptimasi ke format NCNN dan dijalankan secara on-device pada Raspberry Pi 4 untuk melakukan proses inferensi. Model ini bertugas untuk mengenali dan mengklasifikasikan jenis hama yang terdapat pada citra daun tanaman bayam hidroponik.

Hasil dari proses inferensi kemudian menjadi output sistem. Output yang dihasilkan berupa informasi jenis hama yang terdeteksi, yaitu whitefly atau winged aphid, beserta nilai confidence yang menunjukkan tingkat kepercayaan model terhadap hasil deteksi tersebut. Hasil deteksi kemudian ditampilkan secara

real-time kepada pengguna melalui antarmuka live stream berbasis Flask web server.



Gambar 4.1 Diagram Blok Sistem

4.2.3 Flowchart Sistem

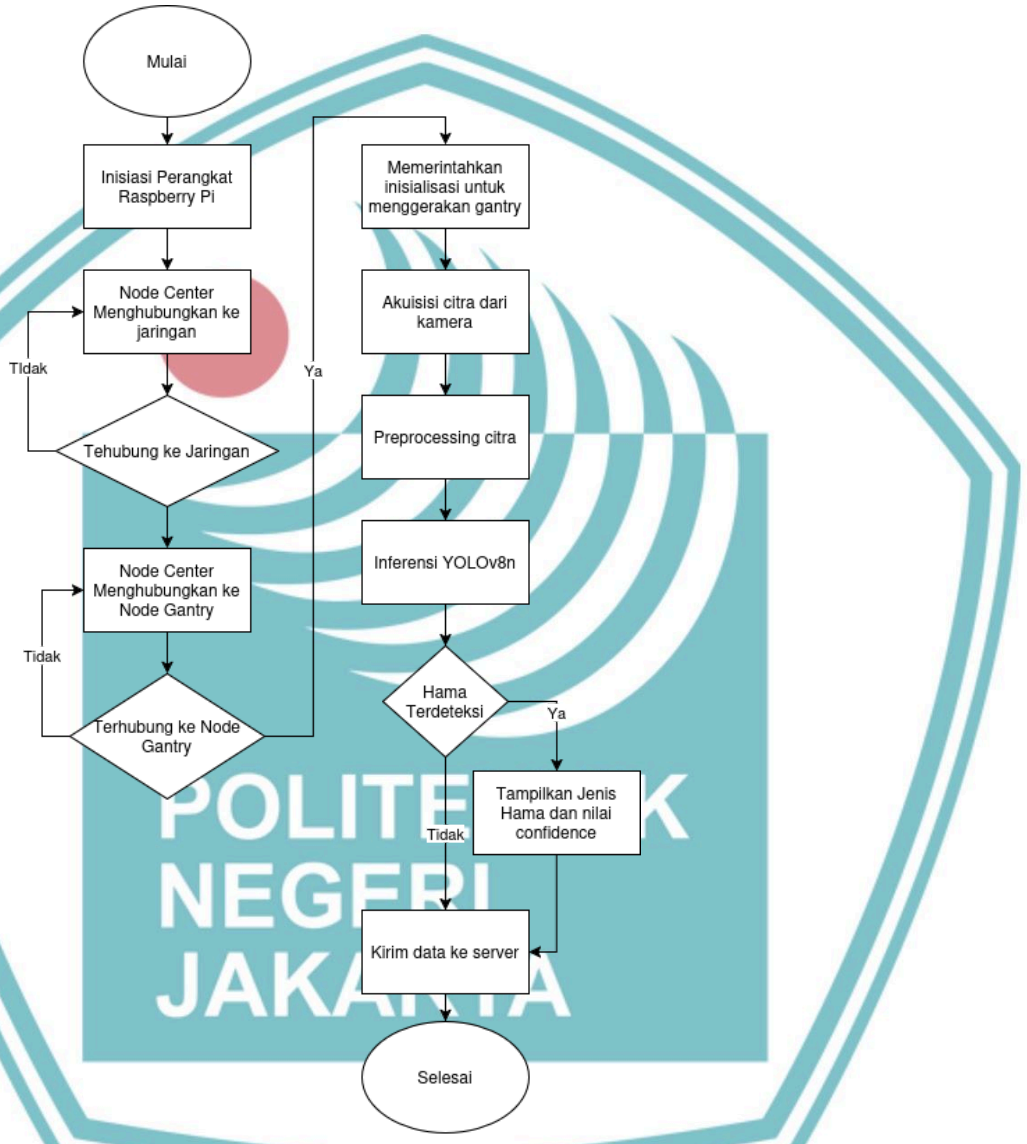
4.2.3.1 Flowchart Sistem Node Camera

Gambar 4.2 menunjukkan *flowchart* dari sistem node kamera yang dibangun. *Flowchart* mendeskripsikan siklus jalannya program utama pada perangkat Raspberry Pi 4 Model B yang berfokus pada manajemen akuisisi citra optik dan eksekusi model kecerdasan buatan secara lokal (*on-device AI*). Proses dimulai dengan fase inisialisasi komponen sensor kamera makro dan pemuatan model YOLOv8n yang telah dikompresi ke dalam format Tencent NCNN. Setelah sistem siap, kamera akan secara kontinu melakukan akuisisi citra dari permukaan daun tanaman bayam pada tray hidroponik. Citra matriks mentah (*raw image*) yang ditangkap kemudian masuk ke dalam tahap *preprocessing* data berupa penyesuaian dimensi piksel (*resizing*) menjadi format 640x640 piksel agar sesuai dengan arsitektur input model.

Selanjutnya, proses inferensi deteksi hama dieksekusi secara lokal oleh model YOLOv8n NCNN. Jika objek hama (*whitefly* atau *aphid*) terdeteksi di atas ambang batas konfiden, sistem secara otomatis akan melokalisasi posisi target menggunakan kotak pembatas (*bounding box*), menentukan jenis label kelasnya,

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

serta menampilkan nilai tingkat kepercayaan (*confidence score*). Hasil ekstraksi data tekstual berupa jumlah akumulasi hama dan berkas potongan gambar bukti visual (*cropped image*) kemudian dikirim menuju *cloud server* secara asinkronus melalui protokol MQTT sebagai basis monitoring jarak jauh.



Gambar 4.2 Flowchart Sistem Node Camera

4.2.3.2 Flowchart Sistem Node Gantry

Gambar 4.3 menunjukkan *flowchart* dari sistem node kamera yang dibangun. *Flowchart* ini mendeskripsikan matriks logika pergerakan mekanis otomatis yang dieksekusi oleh mikrokontroler ESP8266 selaku pengendali motor penggerak. Proses dimulai dengan inisialisasi pin masukan dari sensor keselamatan *limit*

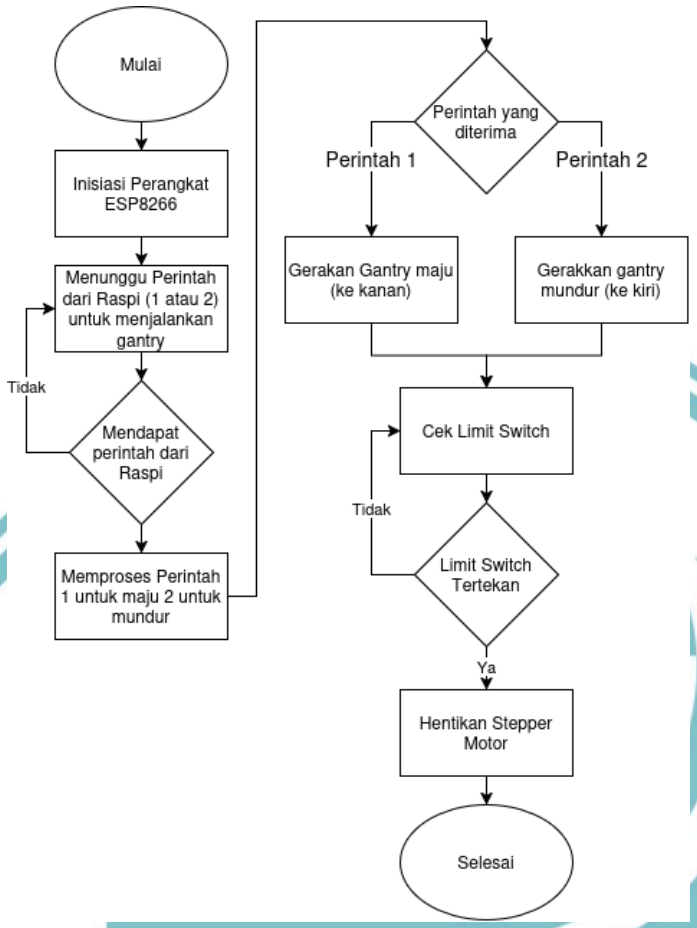
© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

switch serta pin keluaran menuju komponen *driver* motor stepper A4988. Alur kendali berjalan berbasis waktu (*timer-activated mechanism*) atau berdasarkan instruksi eksternal yang diterima melalui jaringan lokal. Ketika *timer* internal memicu jadwal pemindaian, ESP8266 akan mengirimkan sinyal pulsa frekuensi tinggi menuju *driver* motor untuk memutar aktuator Nema 17 Stepper Motor.

Gerakan rotasi motor diubah oleh sabuk transmisi (*timing belt*) menjadi mobilisasi linear horizontal untuk menggerakkan pelat kamera (*gantry plate*) menyusuri pipa tray hidroponik tanaman bayam secara perlahan. Sepanjang pergerakan gantry berlangsung, program secara kontinu melakukan pemeriksaan kondisi (*polling state*) terhadap pin *limit switch*. Jika pelat kamera menyentuh ujung batas rel dan memicu saklar *limit switch*, sistem akan langsung mengeksekusi instruksi interupsi darurat (*hardware interrupt*) untuk menghentikan suplai arus pulsa pada *driver* motor secara instan. Langkah pencegahan ini memastikan pergerakan mekanis gantry berhenti tepat di titik koordinat akhir dengan aman tanpa risiko benturan fisik struktural alat.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



Gambar 4.3 Flowchart Sistem Node Camera

4.2.4 Rancangan Perangkat Keras

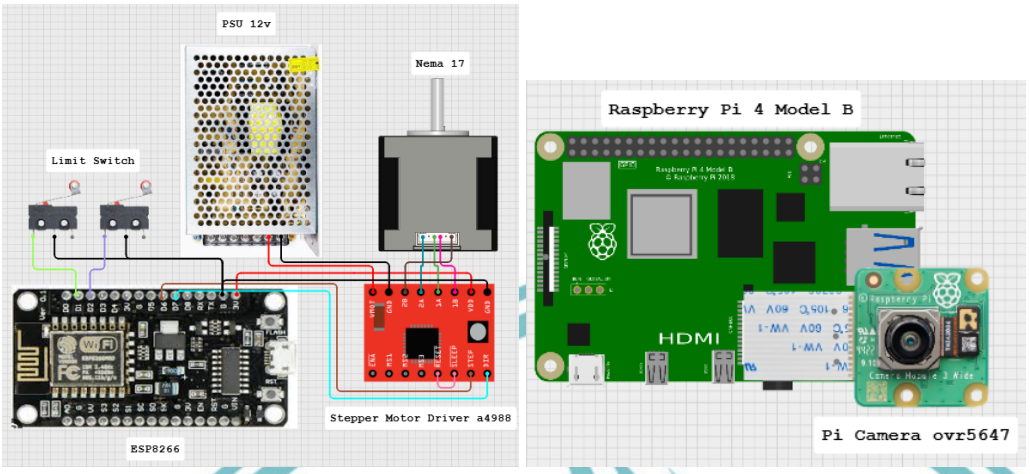
Gambar 4.4 menunjukkan rancangan perangkat keras sistem deteksi hama pada tanaman bayam hidroponik yang dikembangkan dalam penelitian ini. Sistem dirancang menggunakan dua node yang saling terintegrasi, yaitu Node Camera dan Node Gantry, yang berkomunikasi melalui antarmuka serial USB. Pembagian fungsi ke dalam dua node bertujuan untuk memisahkan proses komputasi deteksi hama dan proses pengendalian aktuator sehingga kinerja sistem menjadi lebih stabil dan mudah dikembangkan.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

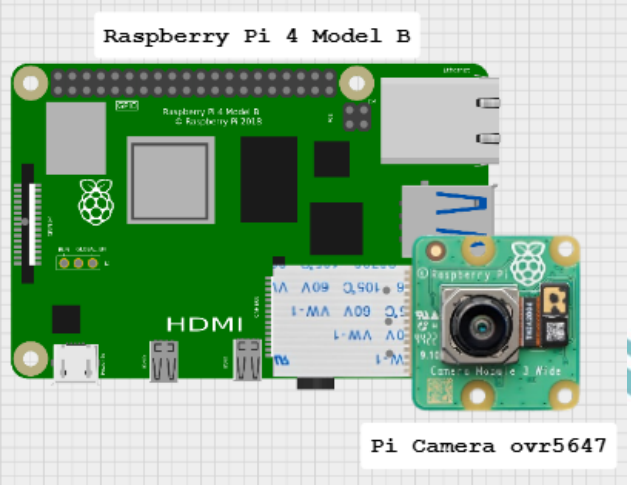


Gambar 4.4 Rancangan Perangkat Keras

4.2.4.1 Racangan Node Camera dan Pinout

Node Camera menggunakan Raspberry Pi 4 Model B sebagai pusat pemrosesan sistem. Perangkat ini bertugas mengendalikan seluruh proses deteksi hama, dimulai dari akuisisi citra menggunakan modul kamera Raspberry Pi, menjalankan proses inferensi model YOLOv8n yang telah dikonversi ke format NCNN, hingga menghasilkan informasi berupa lokasi (*bounding box*), kelas objek, dan nilai *confidence* dari setiap objek yang terdeteksi.

Selain melakukan proses inferensi, Raspberry Pi 4 juga mengirimkan instruksi pergerakan kepada Node Gantry melalui komunikasi serial USB serta mengirimkan hasil deteksi berupa data dan citra ke server melalui jaringan Wi-Fi menggunakan protokol HTTP.



Gambar 4.5 Rancangan Node Camera

4.2.4.2 Racangan Node Gantry dan Pinout

Node gantry merupakan subsistem aktuator yang berfungsi menggerakkan modul kamera secara horizontal di sepanjang instalasi hidroponik. Pada penelitian ini, node gantry menggunakan ESP8266 sebagai mikrokontroler utama yang menerima perintah pergerakan dari Raspberry Pi 4 melalui komunikasi serial USB. Berdasarkan perintah yang diterima, ESP8266 menghasilkan sinyal kendali untuk mengoperasikan driver motor stepper sehingga motor dapat bergerak sesuai arah dan jumlah langkah yang ditentukan.

Motor stepper yang digunakan adalah NEMA 17 yang dikendalikan menggunakan driver A4988. Driver A4988 berfungsi mengubah sinyal logika berarus rendah dari ESP8266 menjadi arus yang diperlukan oleh lilitan motor stepper sehingga motor dapat berputar secara presisi. Driver ini memperoleh catu daya sebesar 12 V DC dari power supply eksternal sebagai sumber daya utama motor, sedangkan ESP8266 tetap memperoleh tegangan kerja sebesar 5 V melalui koneksi USB.

Untuk mencegah gantry bergerak melebihi batas lintasan, sistem dilengkapi dengan dua buah limit switch yang dipasang pada kedua ujung rel. Limit switch sisi maju digunakan untuk mendeteksi batas maksimum pergerakan ke arah kanan, sedangkan limit switch sisi mundur digunakan untuk mendeteksi batas maksimum pergerakan ke arah kiri. Ketika salah satu limit switch aktif, ESP8266 akan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

menghentikan pergerakan motor sehingga gantry tidak mengalami benturan dengan struktur mekanis.

Konfigurasi pin yang digunakan pada ESP8266 ditunjukkan pada Tabel 4.4.

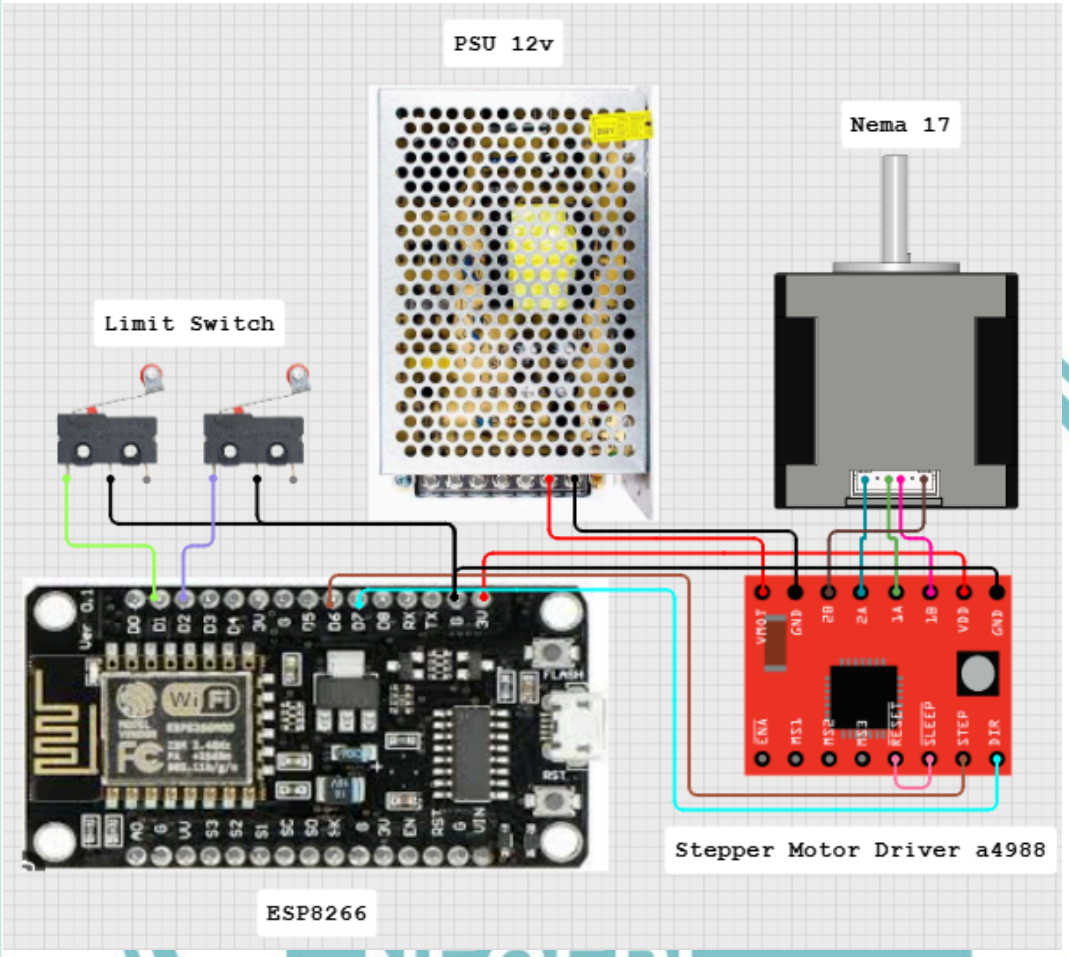
Tabel 4.4 Pinout Node Gantry

ESP8266 ke Stepper Motor Driver A4988	
3V	VDD
GND	GND
D6	STEP
D7	DIR
ESP8266 ke Limit Switch 1	
D1	NO (Normally Open)
GND	COM
ESP8266 ke Limit Switch 2	
D2	NO (Normally Open)
GND	COM
Stepper Motor Driver A4988 ke Stepper Motor Nema 17	
1A	Coil A
1B	Coil A
2A	Coil B
2B	Coil B
Power Supply 12v ke Stepper Motor Driver A4988	
V+	VMOT

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

V-	GND



Gambar 4.6 Rancangan Node Gantry

4.2.5 Rancangan Model

4.2.5.1 Labelling

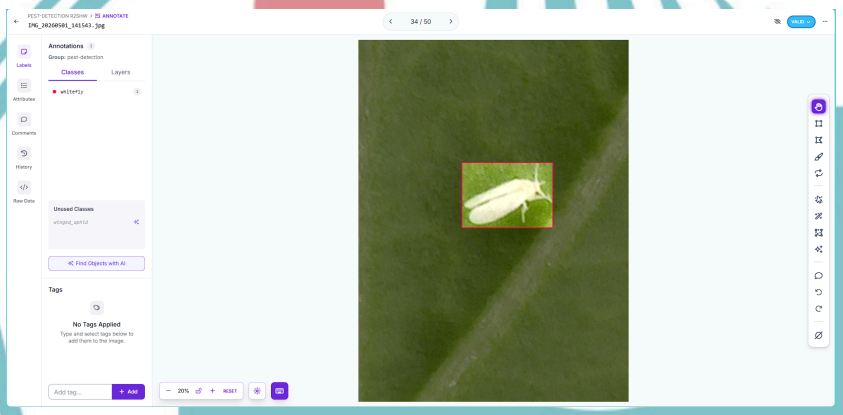
Seluruh sampel dataset diperoleh melalui pengumpulan data langsung (shooting) di area perkebunan hidroponik tanaman bayam. Guna membangun model yang robust dan kaya akan variasi fitur visual, akuisisi citra dilakukan menggunakan kombinasi dari beberapa perangkat optik berikut:

1. Modul Raspberry Pi Camera: Digunakan untuk mengambil sampel citra dengan sudut pandang dan jarak fokus tetap sesuai dengan kondisi riil instalasi pada mesin gantry

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2. Kamera Smartphone dengan tambahan Lensa Makro Eksternal: Beberapa variasi tipe kamera ponsel digunakan dengan tambahan komponen lensa makro. Komponen ini bertujuan untuk menangkap detail fisik hama secara *close-up* dengan resolusi tinggi, sehingga lapisan konvolusi awal model dapat mempelajari detail hama secara tajam.

Setelah seluruh data citra mentah dari berbagai kamera terkumpul, proses pelabelan citra dilakukan dengan menggunakan Roboflow. Data citra dianotasi dibagi menjadi dua kelas target utama, yaitu kelas *whitefly* dan *winged_aphid* menggunakan fitur *bounding box* persegi yang membungkus ketat koordinat fisik hama pada permukaan daun.



Gambar 4.7 Proses Labeling Data

Sebelum dilakukan proses kompilasi dataset, diterapkan komponen *preprocessing* standar pada seluruh dataset guna menyeragamkan format input sebelum masuk ke arsitektur matriks konvolusi (Murdopo, 2021). Komponen *preprocessing* yang diterapkan meliputi:

1. *Auto-Orient*: Memastikan orientasi spasial seluruh gambar.
2. *Resize*: Untuk menemukan titik optimum antara akurasi (*mAP*) dan efisiensi daya komputasi pada CPU Raspberry Pi 4, dilakukan pengujian terhadap tiga variasi dimensi *input layer*, yaitu:
    a. 320 x 320 piksel: Direkayasa untuk meminimalkan beban komputasi secara ekstrem dan mempercepat *inference time*, namun memiliki risiko reduksi informasi spasial untuk objek hama berukuran mikro.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

b. 416 x 416 piksel: Berperan sebagai nilai tengah (*trade-off*) untuk menyeimbangkan kecepatan ekstraksi fitur dan pertahanan detail geometri objek.
c. 640 x 640 piksel: Merupakan resolusi *native*/standar arsitektur YOLOv8n untuk mendapatkan akurasi paling optimal dalam mengenali detail taksonomi serangga kecil, meskipun mengkonsumsi memori RAM yang lebih besar.

Dari hasil komparasi pra pelatihan, resolusi 320 x 320 dipilih sebagai format final karena mempertahankan struktur piksel hama mikro secara tajam saat di-deploy pada NCNN, tanpa memicu *bottleneck* memori yang signifikan pada RAM Raspberry Pi 4.

Setelah tahap *preprocessing* terpenuhi, dilakukan penerapan komponen augmentasi data. Langkah ini diaplikasikan hanya pada training set untuk melipatgandakan jumlah sampel data dengan memanipulasi visual citra asli. Skenario augmentasi diatur untuk menghasilkan 3 output citra baru dari setiap 1 sampel gambar yang meliputi parameter berikut:

1. *Flip*: Pembalikan pada sumbu horizontal dan vertikal. Hal ini dilakukan agar model dapat mengenali orientasi posisi hama dari berbagai arah sudut pandang.
2. *Brightness*: Manipulasi tingkat pencahayaan citra buatan diatur pada batas rentang redup -15% hingga terang +15%. Hal ini dilakukan untuk mengantisipasi fluktuasi intensitas cahaya alami di area instalasi hidroponik dari kondisi siang, sore, hingga malam hari.
3. *Blur*: Efek buram buatan diaplikasikan dengan batas radius maksimal 1 piksel. Hal ini bertujuan sebagai kompensasi teknis untuk melatih model dalam mengantisipasi terjadinya kondisi penurunan fokus citra (*motion blur*) akibat getaran atau pergerakan konstan dari perangkat *gantry plate* saat proses pemindaian *real-time* berlangsung.
4. Grayscale: Penerapan opsi *grayscale* 100% pada seluruh *training set* didasarkan pada strategi rekayasa fitur (*feature engineering*) untuk meminimalkan *domain gap* antara citra akuisisi siang dan malam hari.

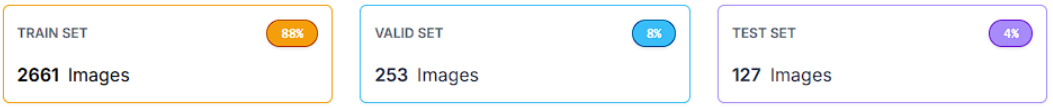
© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pada kondisi riil lapangan, citra siang hari memiliki saturasi warna hijau yang tinggi, sedangkan citra malam hari cenderung kehilangan informasi warna dan didominasi oleh performa pencahayaan buatan (*low-light/monochrome*). Dengan mengonversi seluruh data latih menjadi skala keabuan secara total, model dipaksa untuk mengabaikan informasi kromatisitas (warna) dan fokus secara seimbang (50:50) pada ekstraksi fitur struktural seperti kecerahan lokal (*local luminance*), gradien kontras, dan geometri hama. Pendekatan ini memastikan model YOLOv8n-NCNN memiliki keandalan inferensi yang setara dan tidak mengalami degradasi akurasi (*recurrent bias*) saat terjadi transisi operasional dari siang ke malam hari pada perangkat Raspberry Pi 4.

Setelah proses *preprocessing* dan augmentasi data selesai, dataset keseluruhan yang terdiri dari 3000 data, telah dibagi menjadi tiga bagian: training set yang terdiri dari 2661 digunakan untuk melatih model deteksi hama, dan 253 data validation set digunakan untuk mengukur performa model secara berkala selama proses training, dan membantu mengoptimalkan model agar mendapatkan hasil yang baik. Sementara itu, test set yang berisi 127 data akan digunakan untuk melakukan pengujian pada model yang telah dilatih untuk memberikan gambaran tentang sejauh mana model dapat mengenali objek.



Gambar 4.8 Total Dataset Training, Validation dan Testing

4.2.5.2 Training

Proses training ini memanfaatkan repository Ultralytics untuk implementasi YOLOv8n (Nano) untuk mendapatkan model dengan ukuran file paling minimal namun memiliki akurasi tinggi.Proses training ini memanfaatkan Google Collab yang menggunakan spesifikasi RAM sebesar 12,7GB dan menggunakan GPU T4. Proses training dilakukan dengan menjalankan kode di bawah ini untuk menghasilkan output training yang akan digunakan untuk melakukan testing:

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
model = YOLO("yolov8n.pt")
results = model.train(
    data=f"{dataset.location}/data.yaml",
    epochs=300,
    patience=20,
    imgsz=320,
    batch=-1,
    workers=4,
    device=0,
    name="yolov8n_bayam_ncnn"
)
yolo_output_dir = results.save_dir
```

Proses *training* dilakukan dengan menggunakan beberapa parameter sebagai berikut: (referensi penggunaan parameter)

1. epoch=300: Digunakan untuk menetapkan batas maksimal siklus pelatihan sebanyak 300 iterasi untuk memperkecil nilai *loss function*. Jumlah iterasi yang cukup memberikan kesempatan bagi algoritma backpropagation untuk memperbarui bobot (*weights*) secara stabil hingga mencapai titik konvergensi global (Goodfellow et al., 2016).
2. patience=20: Digunakan untuk mengaktifkan fitur penghentian dini (*Early Stopping*). Jika dalam 20 iterasi berturut-turut nilai metrik validasi tidak menunjukkan adanya perbaikan performa (*loss* tidak turun atau *mAP* tidak naik), proses *training* akan dihentikan secara otomatis oleh sistem sebelum menyentuh batas 300 iterasi. Pendekatan ini merupakan regulasi standar dalam *deep learning* untuk mencegah kondisi *overfitting* serta menghemat waktu komputasi (Prechelt, 2012).
3. imgsz=320: Mengatur dimensi spasial resolusi gambar masukan (*input layer*). Variasi ukuran resolusi ini diuji secara bertahap untuk menganalisis dampaknya terhadap performa ekstraksi fitur. Resolusi tinggi (640) dipertahankan untuk meminimalkan hilangnya informasi piksel hama mikro selama proses dekonvolusi dan *downsampling*, sementara resolusi yang lebih kecil (416 dan 320) dirancang untuk menguji efisiensi komputasi pada *hardware constraint* (Redmon & Farhadi, 2018).

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4. batch=-1: Penggunaan batch -1 mengaktifkan fitur AutoBatch otomatis dari Ultralytics. Komponen sistem akan melakukan benchmark terhadap kapasitas VRAM pada perangkat GPU yang tersedia. Kemudian sistem akan memilih ukuran batch size tertinggi yang paling optimal untuk menghindari terjadinya kondisi error kehabisan memori.
5. workers=4: Komponen ini berfungsi untuk mempercepat proses alokasi data citra dari penyimpanan ke dalam memori sebelum diproses sehingga tidak terjadi bottleneck pada performa komputasi
6. device=0: Komponen ini akan menginstruksikan sistem untuk mengeksekusi pelatihan penuh pada GPU, dan tidak menggunakan CPU, sehingga mempercepat waktu pelatihan secara signifikan.

4.2.5.3 Perancangan Transmisi Data

Sistem ini dirancang untuk dapat mengirimkan data hasil deteksi dari perangkat lokal menuju *cloud server* secara otomatis melalui mekanisme transmisi asinkron. Proses transmisi ini diawali dengan mengekstraksi parameter hasil inferensi berupa data teks yang mencakup informasi waktu (*timestamp*), ID kamera, kelas hama, dan jumlah objek setiap kali model YOLOv8n berhasil mengidentifikasi hama *whitefly* atau *aphid*. Bersamaan dengan proses tersebut, sistem juga mengambil potongan gambar (*cropped image*) dari objek hama terkait sebagai bukti visual lapangan. Seluruh data teks dan potongan gambar digital yang telah diekstraksi kemudian dibungkus menjadi satu kesatuan paket data untuk dikirimkan oleh Raspberry Pi 4 menuju *cloud server* memanfaatkan jaringan internet. Pengiriman paket data ini dieksekusi sepenuhnya di latar belakang (*background process*) secara asinkron, sehingga aktivitas transmisi data menuju *cloud* tidak akan menginterupsi, mengganggu, ataupun memperlambat performa *live inference* pada kamera utama di lapangan.

4.3 Implementasi Sistem

4.3.1 Implementasi Perangkat Keras

Implementasi perangkat keras dilakukan berdasarkan rancangan sistem yang telah disusun pada tahap sebelumnya. Seluruh komponen dirakit dan diintegrasikan

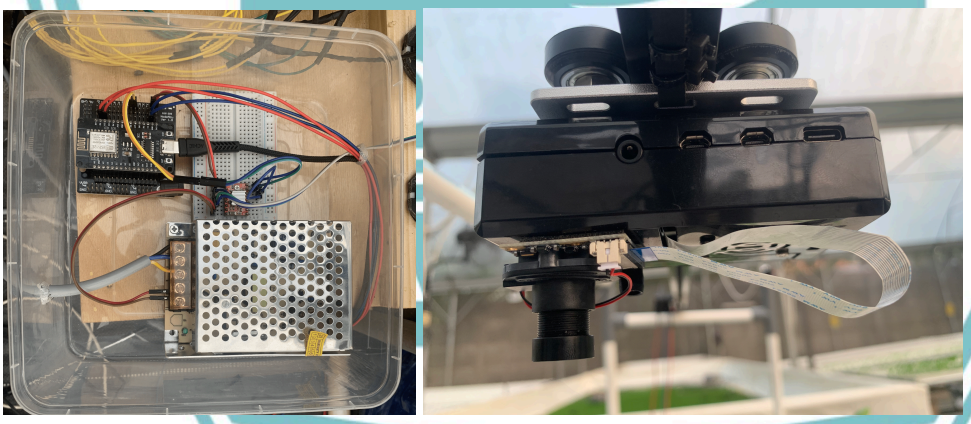
© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

sehingga membentuk sistem deteksi hama yang mampu melakukan proses akuisisi citra, deteksi objek, pengendalian mekanisme gantry, serta pengiriman data hasil deteksi ke server.

Sebagaimana ditunjukkan pada Gambar 4.10, perangkat keras dibagi menjadi dua unit utama, yaitu Node Center dan Node Gantry. Node Center menggunakan Raspberry Pi 4 yang dipasang pada gantry plate sebagai pusat pemrosesan sistem. Perangkat ini bertugas melakukan akuisisi citra dari kamera, menjalankan proses inferensi model YOLOv8n yang telah dioptimalkan menggunakan NCNN, serta mengelola penyimpanan dan pengiriman data hasil deteksi ke server.

Sementara itu, Node Gantry menggunakan ESP8266 sebagai pengendali aktuator yang bertugas menerima instruksi dari Raspberry Pi 4 melalui komunikasi serial. ESP8266 mengendalikan driver motor stepper A4988 untuk menggerakkan motor NEMA17 sehingga gantry plate dapat bergerak secara horizontal di sepanjang instalasi hidroponik. Selain itu, dua buah limit switch digunakan sebagai pembatas pergerakan agar gantry berhenti ketika mencapai ujung lintasan.



Gambar 4.10 Implementasi Perangkat Keras

4.3.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak menjelaskan realisasi rancangan sistem ke dalam bentuk program yang dijalankan pada Raspberry Pi 4 dan ESP8266. Perangkat lunak dikembangkan menggunakan bahasa pemrograman Python untuk Raspberry Pi dan Arduino IDE untuk ESP8266. Implementasi meliputi proses akuisisi citra,

deteksi hama menggunakan model YOLOv8n-NCNN, komunikasi antar perangkat, pengendalian motor stepper, serta pengiriman data ke server.

4.3.2.1 Implementasi Akuisisi Citra

Implementasi akuisisi citra merupakan tahap awal dalam proses deteksi hama pada sistem yang dikembangkan. Tahap ini bertujuan untuk memperoleh citra tanaman bayam secara kontinu menggunakan Raspberry Pi Camera sebagai masukan bagi proses inferensi model YOLOv8n. Selain melakukan pengambilan gambar, sistem juga mengatur parameter kamera agar citra yang dihasilkan memiliki kualitas yang memadai untuk mendeteksi objek hama berukuran kecil.

Pada implementasinya, Raspberry Pi Camera diinisialisasi menggunakan pustaka Picamera2 dengan konfigurasi format warna BGR888 dan resolusi 1280 × 960 piksel. Format BGR dipilih karena kompatibel dengan pustaka OpenCV yang digunakan pada proses pengolahan citra, sedangkan resolusi tersebut dipilih agar detail objek hama tetap terlihat jelas sebelum dilakukan proses inferensi.

Selanjutnya dilakukan pengaturan beberapa parameter kamera, meliputi Auto Exposure (AE), Auto White Balance (AWB), Saturation, Contrast, dan Sharpness. Pengaturan ini bertujuan untuk menghasilkan citra yang lebih stabil pada berbagai kondisi pencahayaan, meningkatkan kontras antara daun dan hama, serta mempertajam detail objek sehingga mempermudah proses deteksi.

Setelah konfigurasi selesai dilakukan, kamera dijalankan secara kontinu menggunakan fungsi *background worker*. Fungsi ini bertugas mengambil frame secara berulang dari Raspberry Pi Camera, kemudian menyimpannya ke dalam variabel global sehingga dapat langsung digunakan oleh modul inferensi tanpa harus melakukan inisialisasi ulang kamera pada setiap proses deteksi. Pengambilan citra dilakukan dengan interval sekitar 0,03 detik, sehingga sistem mampu memperoleh sekitar 30 frame per detik (FPS) dengan tetap menjaga penggunaan sumber daya Raspberry Pi agar tetap efisien.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
picam2 = Picamera2()
config = picam2.create_preview_configuration(main={"format":
"BGR888", "size": (1280, 960)})
picam2.configure(config)
picam2.set_controls({
    "AeConstraintMode": 1,
    "AeExposureMode": 0,
    "AwbMode": 0,
    "Saturation": 1.3,
    "Contrast": 1.2,
    "Sharpness": 2.0,
})
picam2.start()

def camera_stream_worker():
    global global_frame_raw
    log.info("🚀 Background Camera Worker started.")
    while not shutdown_event.is_set():
        try:
            frame_rgb = picam2.capture_array()
            with frame_lock:
                global_frame_raw = frame_rgb
            time.sleep(0.03)
        except Exception as e:
            log.error(f"Error pada thread kamera: {e}")
            time.sleep(1)
```

Gambar 4.11 Potongan Kode Implementasi Akuisisi Citra

Potongan kode pada gambar 4.11 menunjukkan implementasi proses akuisisi citra menggunakan modul Raspberry Pi Camera melalui pustaka Picamera2. Implementasi diawali dengan menginisialisasi kamera, kemudian mengatur format citra menjadi BGR888 dengan resolusi 1280 × 960 piksel. Selanjutnya dilakukan konfigurasi beberapa parameter kamera, seperti Auto Exposure (AE), Auto White Balance (AWB), Saturation, Contrast, dan Sharpness, untuk menghasilkan kualitas citra yang lebih optimal sebelum diproses oleh model deteksi hama.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Setelah seluruh konfigurasi selesai dilakukan, kamera diaktifkan menggunakan fungsi start() sehingga siap melakukan akuisisi citra secara kontinu.

Selanjutnya fungsi camera_stream_worker() yang dijalankan sebagai background thread. Fungsi ini bertugas mengambil citra dari Raspberry Pi Camera secara terus-menerus menggunakan fungsi capture_array(). Setiap frame yang berhasil diperoleh kemudian disimpan ke dalam variabel global sehingga dapat langsung diakses oleh modul inferensi tanpa perlu melakukan proses pengambilan gambar kembali. Selain itu, sistem memberikan jeda selama 0,03 detik pada setiap siklus pengambilan citra untuk menjaga kestabilan proses akuisisi sekaligus mengurangi beban komputasi pada Raspberry Pi 4.

4.3.2.2 Implementasi Inferensi Model YOLOv8n-NCNN

Implementasi model deteksi hama dilakukan menggunakan kelas InferenceEngine yang bertugas mengelola seluruh proses inferensi, mulai dari memuat model YOLOv8, melakukan pra-pemrosesan citra, menjalankan proses deteksi objek, hingga menghasilkan keluaran berupa lokasi objek, jenis hama, tingkat kepercayaan (confidence score), dan jumlah hama yang terdeteksi.

Pada tahap inisialisasi, sistem memuat model hasil pelatihan yang telah dikonversi ke format deployment menggunakan pustaka Ultralytics. Selain itu, sistem juga menginisialisasi parameter confidence threshold, ukuran citra (imgsz), serta pemetaan nama kelas objek yang digunakan selama proses inferensi.

```
class InferenceEngine:
    def __init__(self, model_path, min_conf=0.25, imgsz=320):
                log.info(f"🔄 Menginisialisasi YOLO Engine
menggunakan model: {model_path}")
        self.model = YOLO(model_path)
        self.min_conf = min_conf
        self.imgsz = imgsz
        if not self.model.names:
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
                log.warning("⚠️ Metadata kelas tidak ditemukan
pada file Mengaktifkan fallback mapping.")
            self.class_mapping = {
                0: "whitefly",
                1: "winged_aphid"
            }
        else:
            self.class_mapping = self.model.names
               log.info(f"✅ Pemetaan Kelas Terkonfigurasi:
{self.class_mapping}")
```

Gambar 4.12 Potongan Kode Implementasi Inferensi Model

Potongan program pada gambar 4.12 menunjukkan proses inisialisasi model YOLOv8 beserta parameter yang digunakan selama inferensi. Model dimuat melalui kelas YOLO, kemudian sistem menetapkan nilai minimum confidence sebesar 0,25 sebagai batas minimal suatu objek dianggap valid. Selanjutnya dilakukan pemetaan indeks kelas menjadi nama hama sehingga hasil deteksi dapat ditampilkan dalam bentuk label yang mudah dipahami.

Selanjutnya, setiap citra yang diperoleh dari kamera diproses sebelum dilakukan inferensi. Tahap pra-pemrosesan meliputi konversi format warna dari RGB menjadi BGR, kemudian dilakukan center cropping untuk mengambil bagian tengah citra sebagai area deteksi. Teknik ini digunakan agar proses inferensi lebih ringan serta memfokuskan deteksi pada area tanaman yang menjadi objek pengamatan.

```
img_bgr = cv2.cvtColor(frame_rgb, cv2.COLOR_RGB2BGR)
        h, w = img_bgr.shape[:2]
        cy1, cy2 = h//4, 3*h//4
        cx1, cx2 = w//4, 3*w//4
        cropped_frame = img_bgr[cy1:cy2, cx1:cx2]
        t_input = time.time()
        results = self.model(
```

```
            cropped_frame,
            imgsz=self.imgsz,
            conf=self.min_conf,
            verbose=False,
            workers=0
        )
```

Gambar 4.13 Potongan Kode Implementasi Inferensi Model

Potongan kode pada gambar 4.13 memperlihatkan proses pra-pemrosesan citra sebelum inferensi dilakukan. Setelah citra dipotong pada area tengah (center crop), model YOLOv8 melakukan proses deteksi objek menggunakan ukuran citra yang telah ditentukan. Nilai workers=0 digunakan agar proses inferensi berjalan pada satu thread sehingga penggunaan sumber daya Raspberry Pi tetap stabil.

Setelah proses inferensi selesai, sistem melakukan post-processing dengan mengambil seluruh hasil deteksi yang dihasilkan model. Informasi yang diperoleh meliputi koordinat bounding box, nama kelas objek, nilai confidence, serta jumlah objek yang berhasil terdeteksi pada masing-masing kelas.

```
payload.append({
                "class_name": cls_name,
                "confidence": round(conf, 4),
                "bbox": [
                    round(bx1, 1),
                    round(by1, 1),
                    round(bx2 - bx1, 1),
                    round(by2 - by1, 1)
                ],
            })
```

Gambar 4.14 Potongan Kode Implementasi Inferensi Model

Potongan program pada gambar 4.14 menunjukkan proses penyusunan hasil deteksi ke dalam struktur data (payload). Setiap objek yang berhasil dideteksi akan disimpan dalam bentuk nama kelas, nilai confidence, serta koordinat

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

bounding box. Data tersebut selanjutnya digunakan untuk menampilkan kotak deteksi pada citra, menghitung jumlah hama yang terdeteksi, serta dikirim ke server sebagai hasil pemantauan sistem.

4.3.2.3 Implementasi Komunikasi Raspberry Pi dengan ESP8266

Komunikasi antara Raspberry Pi 4 dan ESP8266 diimplementasikan menggunakan komunikasi serial USB melalui pustaka **pySerial**. Raspberry Pi bertindak sebagai pengirim instruksi (master), sedangkan ESP8266 bertugas menerima perintah untuk mengendalikan pergerakan motor stepper pada sistem gantry.

```
ser = serial.Serial()
ser.port = args.serial_port
ser.baudrate = 115200
ser.timeout = 0.1
ser.dtr = False
ser.rts = False
ser.open()

ser.write(b'P\n')
ser.flush()

if trigger == "SYSTEM_READY":
      ser.write(b'1')
elif trigger == "START":
      shared_state["is_scanning"] = True
elif trigger == "END":
      shared_state["is_scanning"] = False
```

Gambar 4.15 Potongan Kode Implementasi Komunikasi Raspberry Pi dengan ESP8266

Potongan program pada gambar 4.15 menunjukkan implementasi komunikasi serial antara Raspberry Pi 4 dan ESP8266 menggunakan pustaka **pySerial**. Tahap pertama dilakukan dengan menginisialisasi port komunikasi menggunakan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

baudrate 115200 bps. Setelah koneksi berhasil dibuka, Raspberry Pi mengirimkan sinyal handshake (P) untuk memastikan ESP8266 telah siap menerima instruksi.

Selanjutnya Raspberry Pi membaca data yang dikirimkan oleh ESP8266 melalui komunikasi serial. Beberapa status yang diterima antara lain SYSTEM_READY, START, dan END. Status SYSTEM_READY menunjukkan bahwa ESP8266 telah selesai melakukan inisialisasi sehingga Raspberry Pi dapat mengirimkan perintah awal. Status START digunakan untuk memulai proses pemindaian tanaman, sedangkan END menandakan bahwa proses pemindaian telah selesai sehingga sistem menghentikan proses akuisisi citra sementara.

4.3.2.4 Implementasi Pengiriman Data

Implementasi pengiriman data dilakukan menggunakan protokol MQTT dengan memanfaatkan pustaka Paho MQTT pada Raspberry Pi 4. Modul ini bertugas membangun koneksi dengan MQTT Broker, kemudian mengirimkan hasil deteksi secara real-time menuju server setelah proses inferensi selesai dilakukan.

Pada saat sistem dijalankan, objek MQTTClient melakukan inisialisasi koneksi ke broker menggunakan alamat host dan port yang telah dikonfigurasi. Setelah koneksi berhasil dibangun, proses komunikasi dijalankan secara asynchronous menggunakan loop_start(), sehingga proses pengiriman data tidak mengganggu proses deteksi yang sedang berjalan.

Setiap hasil deteksi dikemas dalam bentuk payload berformat JSON yang berisi informasi API Key, waktu deteksi (*timestamp*), citra hasil deteksi yang telah dikonversi ke format Base64, data hasil deteksi objek (*detections*), serta identitas sesi pemindaian (*scan_session_id*). Payload tersebut kemudian dipublikasikan ke topik MQTT lokatani/detect/up sehingga dapat diterima dan diproses oleh server secara real-time.

```
class MQTTClient(BaseCloudClient):
    def __init__(self, host, broker_url_arg, api_key):
        self._client.connect(broker_host, broker_port, 60)
```

```
        self._client.loop_start()
    def send_detection(self, session_id: str, payload_data:
dict)
        payload = {
            "api_key": self.api_key,
            "timestamp": payload_data.get("timestamp"),
            "image_base64": payload_data["image_base64"],
            "detections": payload_data["detections"],
            "scan_session_id": session_id,
    self._client.publish(
            "lokatani/detect/up",
            json.dumps(payload),
            qos=0)
```

Gambar 4.16 Potongan Kode Implementasi Pengiriman Data

4.3.2.5 Implementasi Pergerakan Gantry Pada ESP8266

Implementasi dimulai dengan menginisialisasi komunikasi serial pada baud rate 115200 bps serta mengatur konfigurasi pin yang digunakan. Pin **STEP** dan **DIR** dikonfigurasikan sebagai keluaran (*output*) untuk mengendalikan driver motor A4988, sedangkan dua buah limit switch dikonfigurasikan sebagai masukan (*input pull-up*) untuk mendeteksi batas lintasan gantry. Setelah seluruh komponen berhasil diinisialisasi, ESP8266 mengirimkan status **SYSTEM_READY** kepada Raspberry Pi sebagai tanda bahwa perangkat siap menerima perintah.

```
const int pinStep        = 12;
const int pinDir         = 13;
const int pinLimitMaju   = 5;
const int pinLimitMundur = 4;
void setup() {
  Serial.begin(115200);
  delay(500);
  Serial.flush();
  while (!Serial) { ; }
  pinMode(pinStep, OUTPUT);
  pinMode(pinDir, OUTPUT);
```


© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
  pinMode(pinLimitMaju, INPUT_PULLUP);
  pinMode(pinLimitMundur, INPUT_PULLUP);
  Serial.println("\nSYSTEM_READY");
  Serial.flush();
}
```

Gambar 4.17 Potongan Kode Implementasi Pergerakan Gantry

ESP8266 menerima karakter yang dikirimkan Raspberry Pi melalui komunikasi serial. Karakter '1' digunakan untuk menggerakkan gantry ke arah maju, sedangkan karakter '2' digunakan untuk menggerakkan gantry ke arah mundur. Setelah perintah diterima, ESP8266 mengubah status sistem menjadi aktif dan mengirimkan pesan START sebagai konfirmasi bahwa proses pergerakan gantry telah dimulai.

```
if (inputUser == '1') {
    arahSiklus = '1';
    Serial.println("START");
    sistemAktif = true;
}

else if (inputUser == '2') {
    arahSiklus = '2';
    Serial.println("START");
    sistemAktif = true;
}
```

Gambar 4.18 Potongan Kode Implementasi Pergerakan Gantry

Selama proses pergerakan, ESP8266 menghasilkan pulsa pada pin STEP untuk menggerakkan motor stepper, sedangkan arah putaran ditentukan oleh kondisi pin DIR. Sistem secara terus-menerus memantau status limit switch. Ketika gantry mencapai ujung lintasan dan limit switch aktif, motor dihentikan dan ESP8266 mengirimkan pesan END kepada Raspberry Pi sebagai informasi bahwa satu siklus pergerakan telah selesai.

```
digitalWrite(pinDir, HIGH);
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
digitalWrite(pinStep, HIGH);
delayMicroseconds(JEDA_SPEED_MOTOR);
digitalWrite(pinStep, LOW);
delayMicroseconds(JEDA_SPEED_MOTOR);
if (digitalRead(pinLimitMaju) == LOW) {
    Serial.println("END");
}
```

Gambar 4.19 Potongan Kode Implementasi Pergerakan Gantry

4.4 Pengujian

4.4.1 Deskripsi Pengujian

Tahapan pengujian dilakukan untuk memenuhi tujuan penelitian. Pengujian dilakukan dengan berbagai aspek dari sistem yang dikembangkan. Terdapat 4 metode pengujian sebagai berikut:

1. Analisis Hyperparameter Model
2. Pengujian Performa Deteksi Model
3. Pengujian Performa Perangkat
4. Pengujian Fungsionalitas Perangkat
5. Pengujian Reliabilitas Sistem

4.4.2 Prosedur Pengujian

4.4.2.1 Prosedur Analisis Hyperparameter Model

Prosedur ini bertujuan untuk menganalisis pengaruh variasi ukuran citra (*image size*) sebagai salah satu hyperparameter terhadap performa model YOLOv8n selama proses pelatihan dan validasi. Selain itu, dilakukan perbandingan performa antara model YOLOv8n dengan model pembanding, yaitu YOLOv5s dan YOLO26n, untuk memperoleh model yang memiliki keseimbangan terbaik antara akurasi dan efisiensi komputasi sebelum diimplementasikan pada Raspberry Pi 4 dalam format NCNN.

Prosedur analisis *hyperparameter*, pelatihan, validasi, dan perbandingan model dilakukan melalui tahapan sebagai berikut:

- Analisis Variasi Hyperparameter Ukuran Citra: Pelatihan dan validasi model YOLOv8n (*Nano*) dilakukan menggunakan *framework* Ultralytics

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

dengan menguji tiga variasi ukuran gambar *input*, yaitu resolusi 640, 416, dan 320 piksel. Selama proses pelatihan, perkembangan performa model diamati pada setiap *epoch* menggunakan metrik *training loss*, *validation loss*, *precision*, *recall*, dan *mAP@50*. Fitur *Early Stopping* diterapkan untuk menghentikan proses pelatihan secara otomatis apabila tidak terjadi peningkatan performa dalam sejumlah *epoch* tertentu, sehingga diperoleh model dengan performa optimal.

- Pelatihan dan Validasi Model Pembanding: Model YOLOv5s dan YOLO26n dilatih serta divalidasi menggunakan dataset, konfigurasi pelatihan, dan lingkungan komputasi yang sama sehingga hasil pengujian dapat dibandingkan secara objektif dengan model YOLOv8n.
- Evaluasi Komparatif Model: Hasil validasi setiap model dibandingkan berdasarkan parameter *precision*, *recall*, *F1-score*, *mAP@50*, serta waktu inferensi pada lingkungan GPU. Analisis dilakukan untuk mengevaluasi keseimbangan antara tingkat akurasi dan efisiensi komputasi dari masing-masing model.
- Kriteria Pemilihan Model: Model yang menunjukkan keseimbangan terbaik antara performa deteksi dan efisiensi komputasi dipilih sebagai kandidat untuk diimplementasikan pada Raspberry Pi 4 sebagai sistem deteksi hama berbasis *edge computing*.

4.4.2.2 Prosedur Pengujian Performa Deteksi Model

Prosedur pengujian performa deteksi model bertujuan untuk mengevaluasi performa deteksi dari setiap model yang telah dikonversi ke format NCNN dan diimplementasikan pada Raspberry Pi 4. Pengujian dilakukan terhadap lima model, yaitu YOLOv8n dengan resolusi masukan 640 piksel, 416 piksel, dan 320 piksel, serta model pembanding YOLOv5s dan YOLO26n. Seluruh model diuji menggunakan perangkat keras, konfigurasi sistem, parameter inferensi, dan 100 citra hasil pengambilan langsung pada instalasi hidroponik yang sama sehingga hasil evaluasi dapat dibandingkan secara objektif.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Citra uji diperoleh melalui proses akuisisi menggunakan kamera Raspberry Pi pada kondisi nyata di instalasi hidroponik. Seluruh citra kemudian dianotasi untuk menghasilkan ground truth yang memuat 145 objek hama sebagai acuan evaluasi. Masing-masing model melakukan inferensi terhadap citra uji tersebut, kemudian hasil prediksi dibandingkan dengan ground truth melalui analisis frame-by-frame.

Setiap model dijalankan secara *real-time* menggunakan citra yang diperoleh dari sensor kamera Raspberry Pi. Hasil deteksi kemudian dievaluasi melalui analisis berbasis bingkai (*frame-by-frame analysis*) dengan membandingkan prediksi model terhadap *ground truth* yang telah divalidasi sebelumnya.

Parameter utama yang dieksekusi dalam prosedur ini meliputi:

- Verifikasi Klasifikasi dan Lokalisasi: Setiap objek yang terdeteksi akan dihitung nilai *Intersection over Union* (IoU) untuk menentukan akurasi posisi *bounding box*.
- Kategorisasi Metrik Evaluasi: Prediksi model diklasifikasikan ke dalam tiga komponen data mentah:
    - True Positive (TP), apabila model berhasil mendeteksi objek dengan kelas yang benar dan memiliki nilai IoU (*Intersection over Union*) ≥ 0.5.
    - False Positive (FP), apabila model mendeteksi objek pada lokasi yang tidak sesuai atau memberikan klasifikasi kelas yang salah.
    - False Negative (FN), apabila objek yang terdapat pada *ground truth* gagal dideteksi oleh model.

Seluruh hasil inferensi, meliputi koordinat *bounding box*, kelas prediksi, dan nilai *confidence score*, dicatat secara otomatis ke dalam berkas log berformat *Comma-Separated Values* (CSV). Data tersebut selanjutnya diolah menggunakan matriks konfusi untuk memperoleh nilai *Precision*, *Recall*, dan *F1-Score* pada tahap analisis data.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4.4.2.3 Prosedur Pengujian Performa Perangkat

Prosedur pengujian performa perangkat bertujuan untuk mengevaluasi efisiensi komputasi dan penggunaan sumber daya Raspberry Pi 4 saat menjalankan sistem deteksi hama secara on-device. Pengujian dilakukan terhadap seluruh model yang telah diimplementasikan ke perangkat, yaitu YOLOv8n dengan resolusi masukan 640 piksel, 416 piksel, dan 320 piksel, serta model pembanding YOLOv5s dan YOLO26n. Setiap model dijalankan menggunakan skenario pengujian yang sama sehingga hasil pengukuran dapat dibandingkan secara objektif.

Selama proses pengujian berlangsung, sistem secara otomatis merekam parameter performa perangkat pada interval waktu tertentu. Parameter yang diamati meliputi penggunaan CPU (%), penggunaan RAM (%), suhu prosesor (°C), Frames Per Second (FPS), serta pipeline latency (s) yang dibutuhkan untuk memproses satu citra mulai dari akuisisi hingga diperoleh hasil deteksi.

Pengujian performa perangkat dilakukan secara bersamaan dengan pengujian performa deteksi model menggunakan 214 citra sampel. Seluruh data hasil pengukuran direkam secara otomatis ke dalam berkas log untuk selanjutnya dianalisis dengan menghitung nilai minimum, maksimum, dan rata-rata dari setiap parameter sebagai dasar evaluasi efisiensi komputasi masing-masing model.

4.4.2.4 Prosedur Pengujian Fungsionalitas

Pengujian fungsionalitas sistem dilakukan untuk memvalidasi bahwa setiap fungsi yang dibangun dapat bekerja sesuai dengan kebutuhan sistem dan logika program yang telah dirancang. Pengujian dilakukan menggunakan metode *Black-Box Testing*, yaitu dengan memberikan masukan (*input*) pada setiap fungsi sistem dan mengamati kesesuaian keluaran (*output*) yang dihasilkan tanpa memperhatikan proses internal program. Tahapan pengujian yang dilakukan meliputi sebagai berikut.

1. Pengujian Konektifitas Jaringan

   Menguji kemampuan Raspberry Pi 4 untuk terhubung ke jaringan Wi-Fi

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

sehingga seluruh layanan sistem yang memerlukan komunikasi jaringan dapat berjalan dengan baik.

2. Pengujian Kamera

   Memastikan modul kamera Raspberry Pi dapat berfungsi dengan baik serta mampu menghasilkan citra yang dapat digunakan oleh sistem.

3. Pengujian Akuisisi Citra

   Memverifikasi bahwa sistem mampu mengambil gambar tanaman secara otomatis sesuai dengan alur kerja yang telah dirancang.

4. Pengujian Deteksi dan Klasifikasi Hama

   Memastikan model YOLOv8n-NCNN mampu mendeteksi keberadaan objek hama serta mengklasifikasikan jenis hama sesuai dengan kelas yang telah ditentukan.

5. Pengujian Skenario dengan dan tanpa Hama

   Menguji respons sistem menggunakan citra yang mengandung objek hama maupun citra tanpa hama untuk memastikan sistem mampu membedakan kedua kondisi tersebut dengan benar.

6. Pengujian Komunikasi Raspberry Pi dengan ESP8266

   Memverifikasi proses pengiriman instruksi dari Raspberry Pi 4 menuju ESP8266 berlangsung dengan benar sesuai dengan protokol komunikasi yang digunakan.

7. Pengujian Penggerak Motor Stepper

   Memastikan ESP8266 mampu mengendalikan driver motor stepper sehingga motor dapat bergerak sesuai dengan instruksi yang diterima, serta berhenti pada posisi yang telah ditentukan.

8. Pengujian Pengiriman Data ke Server

   Memverifikasi bahwa hasil deteksi berupa gambar dan informasi jumlah hama berhasil dikirim dari Raspberry Pi 4 menuju server tanpa mengalami kegagalan transmisi.

4.4.2.5 Prosedur Pengujian Reliabilitas

Prosedur pengujian reliabilitas bertujuan untuk mengevaluasi tingkat ketahanan, stabilitas, dan konsistensi operasional sistem selama dijalankan secara

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

terus-menerus (*continuous monitoring*) dalam jangka waktu tertentu. Pada penelitian ini, pengujian reliabilitas dilakukan menggunakan metode *Durability Testing* untuk mengetahui kemampuan sistem dalam mempertahankan kinerjanya tanpa mengalami kegagalan fungsi (*system crash*) maupun penurunan performa yang signifikan.

Prosedur pengujian dilakukan melalui tahapan sebagai berikut.

1. Pengoperasian Sistem Secara Kontinu

   Seluruh sistem, yang meliputi mekanisme pergerakan gantry, proses akuisisi citra, deteksi hama menggunakan model YOLOv8n dengan optimisasi NCNN, serta pengiriman data ke server, dijalankan secara kontinu selama 6 jam tanpa intervensi pengguna.

2. Pemantauan Performa Sistem

   Selama proses pengujian berlangsung, sistem secara otomatis merekam parameter performa perangkat setiap interval 1 detik. Parameter yang diamati meliputi penggunaan CPU (%), penggunaan RAM (%), suhu prosesor (°C), serta status operasional sistem untuk mengetahui apakah sistem tetap berjalan secara stabil selama periode pengujian.

3. Pengumpulan dan Analisis Data

   Seluruh data hasil pemantauan direkam ke dalam berkas log untuk selanjutnya dianalisis. Analisis dilakukan terhadap nilai minimum, maksimum, dan rata-rata setiap parameter, serta dievaluasi apakah selama pengujian terjadi kegagalan sistem (*system crash*), *memory leak*, *thermal throttling*, atau gangguan lain yang dapat mempengaruhi keandalan sistem.

Hasil pengujian reliabilitas digunakan untuk menilai kemampuan sistem dalam mempertahankan performa dan stabilitas operasional selama dijalankan secara kontinu pada lingkungan implementasi.

4.4.3 Data Hasil Pengujian

4.4.3.1 Data Analisis Hyperparameter Model

Validasi model dilakukan untuk mengevaluasi performa setiap model hasil pelatihan sebelum diimplementasikan pada Raspberry Pi 4. Tahap ini bertujuan untuk menganalisis pengaruh variasi ukuran citra (*image size*) sebagai salah satu

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

hyperparameter terhadap performa model YOLOv8n, serta membandingkan hasil pelatihan model YOLOv8n dengan model pembanding YOLOv5s dan YOLO26n. Hasil validasi digunakan sebagai dasar dalam menentukan model yang memiliki keseimbangan terbaik antara tingkat akurasi dan efisiensi komputasi untuk diimplementasikan pada perangkat *edge*.

Pada penelitian ini, validasi dilakukan terhadap lima konfigurasi model, yaitu YOLOv8n dengan resolusi masukan 640 × 640 piksel, 416 × 416 piksel, dan 320 × 320 piksel, serta model YOLOv5s dan YOLO26n dengan resolusi 320 × 320 piksel. Parameter yang dianalisis meliputi *early stopping epoch*, nilai *mean Average Precision* pada *Intersection over Union* (IoU) 0,5 atau *mAP@50*, serta waktu inferensi pada lingkungan GPU (*GPU inference time*). Rekapitulasi hasil analisis *hyperparameter* dan hasil pelatihan setiap model disajikan pada Tabel 4.5.

Tabel 4.5 Data Analisis Hyperparameter Model

No	Model	Ukuran Gambar	Early Stop Epoch	mAP50 (all)	Waktu Inferensi (GPU)
1	YOLOv8n	640	90	0.967	2,5ms
2	YOLOv8n	416	127	0.965	1,3ms
3	YOLOv8n	320	110	0.959	0,8ms
4	YOLOv5s	320	150	0.960	1,7ms
5	YOLO26N	320	192	0.966	0.8ms

4.4.3.2 Data Hasil Pengujian Performa Deteksi Model

Pengujian performa deteksi model dilakukan untuk mengevaluasi kemampuan setiap model dalam mendeteksi dan mengklasifikasikan objek hama pada tanaman bayam hidroponik. Pengujian dilaksanakan secara langsung pada instalasi hidroponik menggunakan Raspberry Pi 4 dan kamera sebagai perangkat akuisisi

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

citra. Seluruh model diuji menggunakan 100 citra hasil pengambilan langsung pada kondisi lapangan yang sama sehingga hasil evaluasi dapat dibandingkan secara objektif.

Lima model yang diuji terdiri atas YOLOv8n dengan resolusi masukan 640 × 640 piksel, 416 × 416 piksel, dan 320 × 320 piksel, serta model pembanding YOLOv5s 320 dan YOLO26n 320. Seluruh pengujian dilakukan menggunakan perangkat keras, konfigurasi sistem, parameter inferensi, dan sampel citra yang sama. Perbedaan pada setiap pengujian hanya terletak pada model yang digunakan.

Sebanyak 100 citra uji yang diperoleh dari proses pengambilan gambar secara langsung memuat 145 objek hama berdasarkan hasil anotasi ground truth. Seluruh objek tersebut digunakan sebagai acuan dalam proses evaluasi sehingga nilai True Positive (TP), False Positive (FP), dan False Negative (FN) pada setiap model dapat dihitung secara konsisten. Berdasarkan nilai-nilai tersebut kemudian dihitung metrik Precision, Recall, dan F1-Score sebagai indikator performa deteksi masing-masing model.

Tabel 4.6 Ringkasan Hasil Pengujian Model YOLOv8n 640

Parameter Evaluasi	**Jumlah Akumulasi Objek**
True Positive (TP)	145
False Positive (FP)	0
False Negative (FN)	0
Total Citra yang Diambil	100
Rata - Rata Nilai Confidence	83,9

Gambar 4.20 Hasil Pengujian Model YOLOv8n 640

Tabel 4.7 Ringkasan Hasil Pengujian Model YOLOv8n 416

Parameter Evaluasi	**Jumlah Akumulasi Objek**
True Positive (TP)	145
False Positive (FP)	0
False Negative (FN)	0
Total Citra yang Diambil	100
Rata - Rata Nilai Confidence	84

Gambar 4.21 Hasil Pengujian Model YOLOv8n 416

Tabel 4.8 Ringkasan Hasil Pengujian Model YOLOv8n 320

Parameter Evaluasi	**Jumlah Akumulasi Objek**

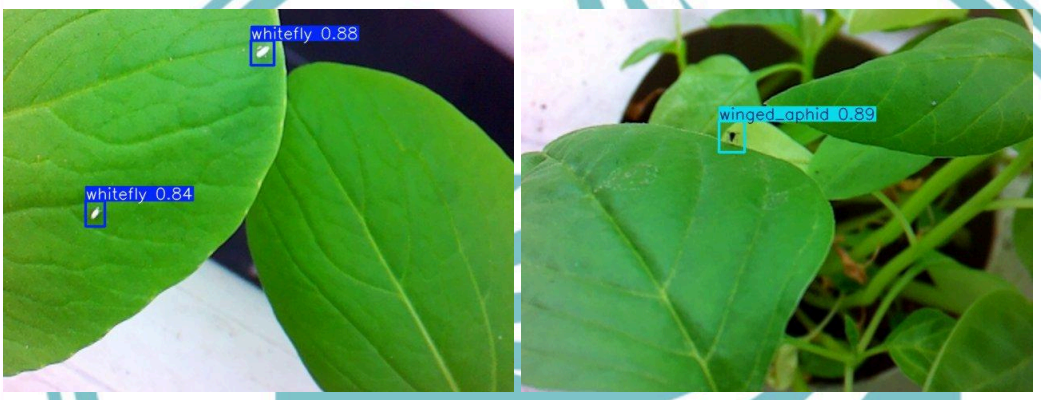
© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

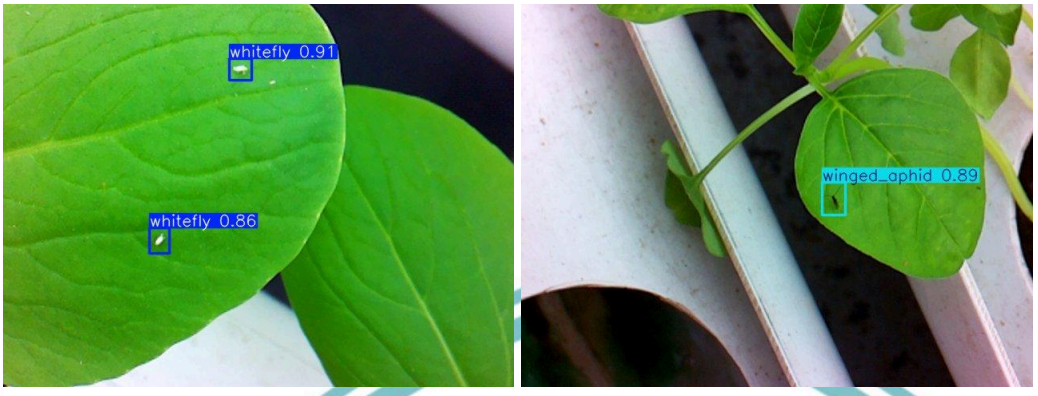
True Positive (TP)	145
False Positive (FP)	4
False Negative (FN)	0
Total Citra yang Diambil	100
Rata - Rata Nilai Confidence	83,4



Gambar 4.22 Hasil Pengujian Model YOLOv8n 320

Tabel 4.9 Ringkasan Hasil Pengujian Model YOLOv5s 320

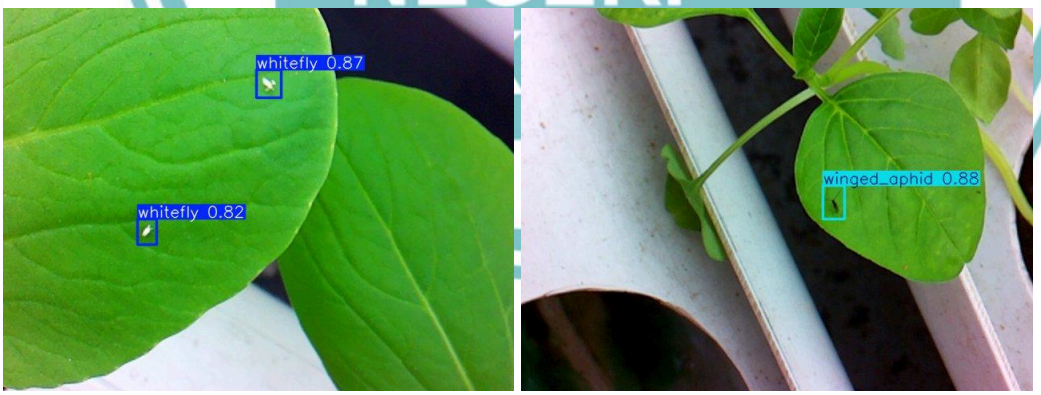
Parameter Evaluasi	**Jumlah Akumulasi Objek**
True Positive (TP)	145
False Positive (FP)	1
False Negative (FN)	0
Total Citra yang Diambil	100
Rata - Rata Nilai Confidence	84,3



Gambar 4.23 Hasil Pengujian Model YOLOv5s 320

Tabel 4.10 Ringkasan Hasil Pengujian Model YOLO26n 320

Parameter Evaluasi	**Jumlah Akumulasi Objek**
True Positive (TP)	145
False Positive (FP)	2
False Negative (FN)	0
Total Citra yang Diambil	100
Rata - Rata Nilai Confidence	83,3



Gambar 4.24 Hasil Pengujian Model YOLO26n 320

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4.4.3.3 Data Hasil Pengujian Performa Perangkat

Pengujian performa perangkat dilakukan untuk mengevaluasi efisiensi komputasi Raspberry Pi 4 dalam menjalankan setiap model deteksi hama yang telah diimplementasikan. Pengujian dilakukan terhadap kelima konfigurasi model menggunakan skenario dan perangkat yang sama sehingga penggunaan sumber daya perangkat dapat dibandingkan secara objektif.

Parameter yang diamati meliputi waktu inferensi (*inference latency*), penggunaan CPU (*CPU Usage*), penggunaan RAM (*RAM Usage*), laju pemrosesan citra (*Frames Per Second/FPS*), serta suhu prosesor selama proses inferensi berlangsung. Seluruh parameter direkam secara berkala yang kemudian diolah untuk memperoleh nilai minimum, maksimum, dan rata-rata sebagai indikator performa masing-masing model.

Rekapitulasi hasil pengujian performa perangkat untuk setiap model disajikan pada tabel-tabel berikut.

Tabel 4.11 Ringkasan Hasil Pengujian Performa Perangkat dengan Model YOLOv8n 640

Parameter Performa	**Nilai Minimum**	**Nilai Maximum**	**Nilai Rata-rata**
Inference Latency (Detik)	0.5775	0.7185	0.661
CPU Usage (%)	80.9	86.9	85.03
RAM Usage (%)	21.4	22.7	21.47
Frames Per Second (FPS)	1.19	1.57	1.46
Core Temperature (°C)	44.3	51.1	49.29

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 4.12 Ringkasan Hasil Pengujian Performa Perangkat dengan Model YOLOv8n 416

Parameter Performa	**Nilai Minimum**	**Nilai Maximum**	**Nilai Rata-rata**
Inference Latency (Detik)	0.3101	0.4938	0.320
CPU Usage (%)	75.2	81.7	79.20
RAM Usage (%)	20.1	22.1	20.23
Frames Per Second (FPS)	2.25	3	2.94
Core Temperature (°C)	46.3	50.1	48.43

Tabel 4.13 Ringkasan Hasil Pengujian Performa Perangkat dengan Model YOLOv8n 320

Parameter Performa	**Nilai Minimum**	**Nilai Maximum**	**Nilai Rata-rata**
Inference Latency (Detik)	0.1984	0.309	0.200
CPU Usage (%)	68.8	77.8	75.44
RAM Usage (%)	19.8	21.1	19.85
Frames Per Second (FPS)	3.39	4.51	4.41
Core Temperature	44.3	49.2	47.01

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

(°C)			

Tabel 4.14 Ringkasan Hasil Pengujian Performa Perangkat dengan Model YOLOv5s 320

Parameter Performa	**Nilai Minimum**	**Nilai Maximum**	**Nilai Rata-rata**
Inference Latency (Detik)	1.0278	1.479	1.1736
CPU Usage (%)	63.9	79.2	71.03
RAM Usage (%)	20.1	21.5	20.48
Frames Per Second (FPS)	0.67	0.96	0.83
Core Temperature (°C)	44.8	49.2	47.0

Tabel 4.15 Ringkasan Hasil Pengujian Performa Perangkat dengan Model YOLO26n 320

Parameter Performa	**Nilai Minimum**	**Nilai Maximum**	**Nilai Rata-rata**
Inference Latency (Detik)	0.093	0.339	0.135
CPU Usage (%)	63.4	75.6	68.93
RAM Usage (%)	19.4	21.4	19.59
Frames Per Second (FPS)	3.48	8.88	7.13

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Core Temperature (°C)	52.1	68.7	62.6

4.4.3.4 Data Hasil Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan untuk memverifikasi bahwa seluruh fungsi yang telah diimplementasikan pada sistem dapat berjalan sesuai dengan kebutuhan dan logika program yang telah dirancang. Pengujian menggunakan metode Black-Box Testing, yaitu dengan memberikan masukan pada setiap fungsi sistem kemudian mengamati kesesuaian keluaran yang dihasilkan tanpa memperhatikan proses internal program.

Fungsi yang diuji meliputi konektivitas jaringan, kamera, akuisisi citra, deteksi dan klasifikasi hama, pengujian pada kondisi terdapat hama maupun tanpa hama, komunikasi antara Raspberry Pi 4 dan ESP8266, penggerak motor stepper, serta pengiriman data ke server. Hasil pengujian fungsionalitas sistem disajikan pada Tabel 4.16.:

Tabel 4.16 Data Hasil Pengujian Fungsionalitas

Fitur/Sub-sistem	Skenario Pengujian	Hasil	Status
Konektivitas Jaringan	Raspberry Pi dihubungkan ke jaringan Wi-Fi	Raspberry Pi berhasil terhubung ke jaringan Wi-Fi dan memperoleh akses internet	Berhasil
Kamera	Menjalankan modul kamera Raspberry Pi	Kamera berhasil menangkap gambar dengan baik	Berhasil
Akuisisi Citra	Menjalankan proses	Citra berhasil diambil dan disimpan pada penyimpanan lokal	Berhasil

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

	pengambilan gambar otomatis		
Deteksi dan Klasifikasi Hama	Memberikan citra yang mengandung hama	Hama berhasil dideteksi dan diklasifikasikan sesuai kelas model	Berhasil
Pengujian dengan dan tanpa Hama	Memberikan citra yang mengandung hama dan citra tanpa hama	Sistem mendeteksi hama pada citra yang mengandung hama dan tidak memberikan deteksi pada citra tanpa hama	Berhasil
Komunikasi Raspberry Pi–ESP8266	Raspberry Pi mengirimkan perintah ke ESP8266	Perintah berhasil diterima oleh ESP8266	Berhasil
Penggerak Motor Stepper	ESP8266 mengirimkan sinyal ke driver motor	Motor stepper bergerak dan berhenti sesuai instruksi program	Berhasil
Pengiriman Data ke Server	Mengirim hasil deteksi ke server	Data hasil deteksi beserta gambar berhasil diterima server	Berhasil

4.4.3.5 Data Hasil Pengujian Reliabilitas

Pengujian reliabilitas dilakukan untuk mengevaluasi kemampuan sistem dalam mempertahankan kestabilan operasional selama dijalankan secara terus-menerus selama 6 jam. Selama pengujian berlangsung, sistem menjalankan seluruh fungsi utama yang meliputi pergerakan gantry, akuisisi citra, deteksi hama menggunakan model YOLOv8n-NCNN, serta pengiriman data ke server.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Parameter yang diamati meliputi penggunaan CPU, penggunaan RAM, dan suhu prosesor yang direkam secara otomatis setiap interval satu detik. Selain itu, dilakukan pula pengamatan terhadap kondisi operasional sistem untuk mengetahui apakah terjadi *system crash*, *memory leak*, atau *thermal throttling* selama proses pengujian. Ringkasan hasil pengujian reliabilitas disajikan pada Tabel 4.17 dan Tabel 4.18.

Tabel 4.17 Ringkasan Hasil Pengujian Reliabilitas

Parameter Performa	**Nilai Minimum**	**Nilai Maximum**	**Nilai Rata-rata**
CPU Usage (%)	63.9	86.9	76.62
RAM Usage (%)	19.8	22.7	20.24
Core Temperature (°C)	44.3	51.1	47.79

Tabel 4.18 Status Operasional Selama Pengujian Reliabilitas

Parameter	**Hasil**
Durasi Pengujian	6 jam
System Crash	Tidak terjadi
Memory Leak	Tidak terindikasi
Thermal Throttling	Tidak terjadi
Status Operasional Sistem	Berjalan normal selama pengujian

4.4.4 Analisis Data

4.4.4.1 Analisis Hyperparameter Model

Berdasarkan hasil validasi dan perbandingan model pada Tabel 4.5, terlihat bahwa setiap konfigurasi model memiliki karakteristik performa yang berbeda dari sisi

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

kemampuan deteksi, proses pelatihan yang ditunjukkan oleh *early stopping*, serta kecepatan inferensi. Analisis ini dilakukan untuk mengevaluasi pengaruh variasi ukuran citra (*image size*) sebagai salah satu *hyperparameter* terhadap performa model YOLOv8n, serta membandingkan performa YOLOv8n dengan model pembanding YOLOv5s dan YOLO26n. Hasil analisis digunakan sebagai dasar dalam menentukan konfigurasi YOLOv8n yang paling sesuai untuk diimplementasikan pada Raspberry Pi 4.

Hasil pengujian menunjukkan bahwa variasi ukuran citra memberikan pengaruh terhadap performa model YOLOv8n. Model dengan resolusi masukan 640 × 640 piksel memperoleh nilai mAP@50 tertinggi sebesar 0,967 dan mencapai kondisi *early stopping* pada epoch ke-90. Namun, konfigurasi ini memiliki waktu inferensi paling lama, yaitu 2,5 ms. Ketika ukuran citra diturunkan menjadi 416 × 416 piksel, nilai mAP@50 hanya mengalami penurunan sebesar 0,002 menjadi 0,965, sedangkan waktu inferensi berkurang menjadi 1,3 ms. Pada resolusi 320 × 320 piksel, model mencapai *early stopping* pada epoch ke-110, dengan nilai mAP@50 sebesar 0,959 dan waktu inferensi 0,8 ms. Hasil tersebut menunjukkan bahwa semakin kecil ukuran citra masukan, proses inferensi menjadi lebih cepat, namun diikuti dengan penurunan nilai mAP@50.

Dari sisi proses pelatihan, konfigurasi YOLOv8n dengan resolusi 640 × 640 piksel memenuhi kriteria *early stopping* pada epoch ke-90, lebih awal dibandingkan konfigurasi 416 × 416 piksel (epoch ke-127) dan 320 × 320 piksel (epoch ke-110). Perbedaan tersebut mengindikasikan bahwa variasi ukuran citra sebagai *hyperparameter* turut mempengaruhi proses pelatihan model yang ditunjukkan oleh perbedaan epoch saat mekanisme *early stopping* aktif.

Apabila dibandingkan dengan model pembanding, YOLOv5s memperoleh nilai mAP@50 sebesar 0,960 dengan waktu inferensi 1,7 ms, sedangkan YOLO26n memperoleh nilai mAP@50 sebesar 0,966 dengan waktu inferensi 0,8 ms. Performa tersebut menunjukkan bahwa YOLO26n mampu memberikan kemampuan deteksi yang hampir setara dengan YOLOv8n resolusi 640 × 640

piksel (mAP@50 0,967), namun dengan efisiensi komputasi yang lebih tinggi. Meskipun demikian, pada penelitian ini YOLO26n digunakan sebagai model pembanding untuk memberikan gambaran perbandingan performa antararsitektur YOLO, sedangkan fokus utama penelitian adalah menganalisis pengaruh variasi ukuran citra terhadap performa model YOLOv8n.

Berdasarkan hasil analisis tersebut, konfigurasi YOLOv8n dengan resolusi masukan 320 × 320 piksel dipilih sebagai model yang digunakan pada tahap implementasi dan pengujian sistem. Pemilihan ini didasarkan pada pertimbangan bahwa model tersebut merupakan varian YOLOv8n dengan resolusi masukan 320 × 320 piksel yang memiliki waktu inferensi tercepat, yaitu 0,8 ms, sehingga lebih sesuai untuk implementasi pada Raspberry Pi 4 yang memiliki keterbatasan sumber daya komputasi. Meskipun nilai mAP@50 yang diperoleh (0,959) lebih rendah dibandingkan konfigurasi 640 × 640 piksel (0,967), selisihnya hanya sebesar 0,008, sedangkan waktu inferensi berkurang dari 2,5 ms menjadi 0,8 ms. Hal tersebut menunjukkan bahwa konfigurasi 320 × 320 piksel mampu memberikan peningkatan efisiensi komputasi yang signifikan dengan penurunan performa deteksi yang relatif kecil.

Dengan demikian, pemilihan resolusi 320 × 320 piksel tidak hanya didasarkan pada waktu inferensi yang lebih singkat, tetapi juga mempertimbangkan kebutuhan implementasi pada Raspberry Pi 4 yang memiliki keterbatasan sumber daya komputasi. Konfigurasi ini dinilai telah memenuhi kebutuhan sistem untuk melakukan deteksi hama secara *real-time* dengan tetap mempertahankan performa deteksi yang baik.

4.4.4.2 Analisis Data Hasil Pengujian Performa Deteksi Model

Berdasarkan hasil pengujian pada Tabel 4.6 sampai dengan Tabel 4.10, diperoleh data jumlah True Positive (TP), False Positive (FP), dan False Negative (FN) dari masing-masing model yang diuji. Nilai tersebut kemudian digunakan untuk menghitung metrik Precision, Recall, dan F1-Score sebagai indikator performa deteksi setiap model. Hasil perhitungan ditunjukkan pada Tabel 4.17.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 4.17 Hasil Perhitungan Precision, Recall, dan F1-Score Setiap Model

Model	Precision	Recall	F1-Score
YOLOv8n 640	100%	100%	100%
YOLOv8n 416	100%	100%	100%
YOLOv8n 320	97,32%	100%	98,96%
YOLOv5s 320	99,32%	100%	99,66%
YOLO26n 320	98,64%	100%	99,31%

Berdasarkan hasil perhitungan tersebut, model YOLOv8n dengan resolusi masukan 640 × 640 piksel dan 416 × 416 piksel memperoleh performa deteksi terbaik, dengan nilai Precision, Recall, dan F1-Score masing-masing sebesar 100%. Hasil ini menunjukkan bahwa kedua konfigurasi tersebut mampu mendeteksi seluruh 145 objek hama pada 100 citra uji tanpa menghasilkan False Positive (FP) maupun False Negative (FN).

Sementara itu, YOLOv5s 320 memperoleh nilai Precision sebesar 99,32%, Recall sebesar 100%, dan F1-Score sebesar 99,66%. Model ini berhasil mendeteksi seluruh objek hama, namun masih menghasilkan 1 False Positive, sehingga nilai precision sedikit lebih rendah dibandingkan YOLOv8n 640 dan YOLOv8n 416.

Model YOLO26n 320 menghasilkan Precision sebesar 98,64%, Recall sebesar 100%, dan F1-Score sebesar 99,31%. Seluruh objek hama berhasil dideteksi tanpa False Negative, tetapi terdapat 2 False Positive, sehingga nilai precision lebih rendah dibandingkan YOLOv5s 320.

Di antara seluruh konfigurasi YOLOv8n, YOLOv8n 320 memperoleh nilai Precision sebesar 97,32%, Recall sebesar 100%, dan F1-Score sebesar 98,64%. Walaupun seluruh objek hama berhasil dideteksi, model ini menghasilkan 4 False

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Positive, sehingga menjadi konfigurasi YOLOv8n dengan nilai precision terendah pada pengujian ini.

Menariknya, seluruh model yang diuji memperoleh Recall sebesar 100%, yang menunjukkan bahwa 145 objek hama yang terdapat pada 100 citra uji berhasil dideteksi seluruhnya oleh masing-masing model. Perbedaan performa antar model tidak dipengaruhi oleh kemampuan menemukan objek hama, melainkan oleh jumlah False Positive yang dihasilkan selama proses deteksi. Semakin sedikit False Positive yang muncul, semakin tinggi nilai precision dan F1-Score yang diperoleh.

Berdasarkan hasil tersebut, dapat disimpulkan bahwa YOLOv8n 640 dan YOLOv8n 416 memberikan performa deteksi paling baik pada pengujian ini karena mampu menghasilkan deteksi sempurna tanpa kesalahan klasifikasi maupun deteksi tambahan. Namun demikian, penelitian ini tidak hanya mempertimbangkan aspek akurasi deteksi, tetapi juga mempertimbangkan efisiensi komputasi pada perangkat edge. Oleh karena itu, pemilihan model implementasi akhir juga mempertimbangkan hasil pengujian performa sistem, seperti waktu inferensi, penggunaan CPU, penggunaan memori, serta kemampuan menjalankan inferensi secara real-time pada Raspberry Pi 4.

Hasil pengujian ini menunjukkan bahwa peningkatan ukuran citra masukan tidak selalu memberikan perbedaan yang signifikan terhadap kemampuan mendeteksi objek hama pada kondisi pengujian penelitian ini. Seluruh model mampu mendeteksi seluruh objek hama yang terdapat pada citra uji, sedangkan perbedaan performa terutama dipengaruhi oleh jumlah deteksi berlebih (False Positive) yang dihasilkan oleh masing-masing model. Oleh karena itu, pemilihan model untuk implementasi akhir perlu mempertimbangkan keseimbangan antara akurasi deteksi dan efisiensi komputasi sehingga sistem mampu bekerja secara optimal pada perangkat Raspberry Pi 4.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4.4.4.3 Analisis Data Hasil Pengujian Performa Perangkat

Berdasarkan hasil pengujian performa perangkat pada Tabel 4.11 sampai dengan Tabel 4.15, terlihat bahwa setiap model memberikan karakteristik penggunaan sumber daya Raspberry Pi 4 yang berbeda. Parameter yang dianalisis meliputi inference latency, penggunaan CPU, penggunaan RAM, Frames Per Second (FPS), serta suhu prosesor selama proses inferensi berlangsung.

Pengaruh ukuran citra masukan terhadap beban komputasi terlihat pada ketiga konfigurasi YOLOv8n. Model YOLOv8n dengan resolusi 640 × 640 piksel menghasilkan rata-rata inference latency sebesar 0,661 detik, penggunaan CPU sebesar 85,03%, serta FPS sebesar 1,46. Ketika ukuran citra diturunkan menjadi 416 × 416 piksel, rata-rata inference latency menurun menjadi 0,320 detik, penggunaan CPU menjadi 79,20%, sedangkan FPS meningkat menjadi 2,94. Pada resolusi 320 × 320 piksel, inference latency kembali menurun menjadi 0,200 detik, penggunaan CPU menjadi 75,44%, dan FPS meningkat hingga 4,41. Hasil tersebut menunjukkan bahwa semakin kecil ukuran citra masukan, beban komputasi yang diproses Raspberry Pi menjadi lebih ringan sehingga proses inferensi dapat dilakukan dengan lebih cepat.

Dari sisi penggunaan memori, seluruh konfigurasi YOLOv8n menunjukkan pemakaian RAM yang relatif stabil, yaitu berada pada kisaran 19,85% hingga 21,47%. Hal ini menunjukkan bahwa perubahan ukuran citra masukan tidak memberikan pengaruh yang signifikan terhadap konsumsi memori selama proses inferensi, sehingga Raspberry Pi 4 masih mampu menjalankan model dengan penggunaan RAM yang efisien.

Apabila dibandingkan dengan model pembanding, YOLOv5s menghasilkan rata-rata inference latency sebesar 1,1736 detik dengan penggunaan CPU sebesar 71,03% dan FPS sebesar 0,83. Hasil tersebut menunjukkan bahwa YOLOv5s membutuhkan waktu pemrosesan yang lebih lama dibandingkan konfigurasi YOLOv8n maupun YOLO26n pada perangkat Raspberry Pi 4. Sementara itu, YOLO26n menghasilkan rata-rata inference latency sebesar 0,135 detik,

penggunaan CPU sebesar 68,93%, serta FPS sebesar 7,13, sehingga menunjukkan efisiensi komputasi yang lebih tinggi dibandingkan model lainnya. Namun, suhu prosesor pada YOLO26n mencapai rata-rata 62,6°C, lebih tinggi dibandingkan seluruh konfigurasi YOLOv8n yang berada pada kisaran 47–49°C. Meskipun demikian, suhu tersebut masih berada di bawah batas terjadinya *thermal throttling* pada Raspberry Pi 4 sehingga sistem tetap dapat beroperasi secara stabil selama pengujian.

Berdasarkan keseluruhan hasil pengujian performa perangkat, konfigurasi YOLOv8n dengan resolusi masukan 320 × 320 piksel dipilih sebagai model implementasi pada penelitian ini. Pemilihan tersebut didasarkan pada fokus penelitian yang menganalisis variasi hyperparameter pada YOLOv8n. Dibandingkan konfigurasi YOLOv8n lainnya, model ini memberikan inference latency paling rendah, penggunaan CPU yang lebih ringan, penggunaan RAM yang tetap efisien, serta FPS tertinggi. Hasil tersebut menunjukkan bahwa konfigurasi YOLOv8n 320 × 320 piksel mampu memberikan keseimbangan yang baik antara efisiensi komputasi dan performa deteksi sehingga sesuai untuk diimplementasikan pada Raspberry Pi 4 sebagai perangkat *edge computing*.

4.4.4.4 Analisis Data Hasil Pengujian Fungsionalitas

Berdasarkan hasil pengujian fungsionalitas pada Tabel 4.16, seluruh fungsi yang diimplementasikan pada sistem berhasil dijalankan sesuai dengan kebutuhan dan spesifikasi yang telah dirancang. Seluruh skenario pengujian memperoleh status berhasil, sehingga tingkat keberhasilan fungsional sistem mencapai 100%.

Hasil pengujian menunjukkan bahwa Raspberry Pi 4 mampu terhubung dengan jaringan Wi-Fi secara stabil sehingga komunikasi data dengan server dapat berlangsung dengan baik. Modul kamera juga berfungsi sesuai harapan dengan mampu menangkap citra tanaman, sedangkan proses akuisisi citra berhasil menyimpan hasil pengambilan gambar ke media penyimpanan lokal secara otomatis.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Pada aspek deteksi, model YOLOv8n-NCNN mampu mendeteksi keberadaan hama sekaligus mengklasifikasikan jenis hama sesuai dengan kelas yang telah dilatih. Selain itu, pengujian menggunakan citra yang mengandung hama maupun citra tanpa hama menunjukkan bahwa sistem dapat memberikan respons yang sesuai terhadap kedua kondisi tersebut.

Hasil pengujian juga menunjukkan bahwa komunikasi antara Raspberry Pi 4 dan ESP8266 berjalan dengan baik, sehingga setiap instruksi yang dikirimkan dapat diterima dan dieksekusi oleh mikrokontroler. Hal tersebut memungkinkan motor stepper bergerak sesuai dengan perintah program dan berhenti pada kondisi yang telah ditentukan. Di sisi lain, proses pengiriman data hasil deteksi beserta citra menuju server juga berhasil dilakukan tanpa mengalami kegagalan transmisi selama pengujian berlangsung.

Berdasarkan keseluruhan hasil pengujian tersebut, dapat disimpulkan bahwa seluruh komponen perangkat keras maupun perangkat lunak telah terintegrasi dengan baik dan mampu menjalankan fungsi yang dirancang. Dengan demikian, sistem dinyatakan memenuhi aspek fungsional sebagai pendukung implementasi deteksi hama pada tanaman bayam hidroponik berbasis Raspberry Pi 4.

4.4.4.5 Analisis Data Hasil Pengujian Reliabilitas

Berdasarkan hasil pengujian reliabilitas pada Tabel 4.17 dan Tabel 4.18, sistem mampu beroperasi secara kontinu selama 6 jam tanpa mengalami gangguan yang menyebabkan proses pemantauan berhenti. Selama periode pengujian, seluruh fungsi utama sistem yang meliputi pergerakan gantry, akuisisi citra, deteksi hama menggunakan model YOLOv8n-NCNN, serta pengiriman data ke server dapat dijalankan secara berkelanjutan.

Berdasarkan parameter performa perangkat, penggunaan CPU berada pada rentang 63,9% hingga 86,9% dengan rata-rata 76,62%. Nilai tersebut menunjukkan bahwa beban komputasi yang dihasilkan selama proses inferensi masih dapat ditangani oleh Raspberry Pi 4 tanpa menyebabkan kegagalan

operasional. Sementara itu, penggunaan RAM berada pada kisaran 19,8% hingga 22,7% dengan rata-rata 20,24%, yang menunjukkan pemanfaatan memori relatif stabil selama sistem dijalankan secara terus-menerus.

Dari sisi suhu operasional, prosesor Raspberry Pi 4 memiliki suhu minimum 44,3°C, suhu maksimum 51,1°C, dan suhu rata-rata 47,79°C selama pengujian. Nilai tersebut masih berada jauh di bawah batas suhu yang umumnya memicu mekanisme thermal throttling pada Raspberry Pi 4, sehingga tidak ditemukan indikasi penurunan performa akibat peningkatan temperatur prosesor.

Hasil pengamatan terhadap kondisi operasional sistem juga menunjukkan bahwa selama pengujian tidak terjadi system crash, tidak ditemukan indikasi memory leak, serta tidak terjadi thermal throttling. Hal ini ditunjukkan oleh kestabilan penggunaan memori, suhu prosesor yang tetap berada pada rentang normal, serta kemampuan sistem dalam menjalankan seluruh proses secara kontinu hingga pengujian selesai.

Berdasarkan keseluruhan hasil pengujian tersebut, dapat disimpulkan bahwa sistem memiliki tingkat reliabilitas yang baik untuk dioperasikan secara kontinu pada implementasi deteksi hama tanaman bayam hidroponik berbasis Raspberry Pi 4. Hasil ini menunjukkan bahwa integrasi perangkat keras, model YOLOv8n-NCNN, dan perangkat lunak pendukung mampu mempertahankan kestabilan operasional selama periode pengujian.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem yang telah dilakukan, maka dapat diperoleh beberapa kesimpulan sebagai berikut.

1. Sistem deteksi dan klasifikasi multi-hama pada tanaman bayam hidroponik berhasil dirancang menggunakan arsitektur berbasis Raspberry Pi 4 sebagai pusat pemrosesan, ESP8266 sebagai pengendali mekanisme gantry, serta model YOLOv8n untuk proses deteksi objek. Sistem mampu melakukan akuisisi citra, mendeteksi dua jenis hama yaitu *whitefly* dan *winged aphid*, menggerakkan gantry secara otomatis, serta mengirimkan hasil deteksi ke server sebagai bagian dari sistem monitoring.
2. Model YOLOv8n berhasil diimplementasikan pada Raspberry Pi 4 menggunakan framework Tencent NCNN sehingga dapat menjalankan proses deteksi hama secara real-time. Seluruh fungsi sistem, mulai dari pengambilan citra, proses inferensi, komunikasi dengan ESP8266, hingga pengiriman data ke server, berhasil berjalan sesuai dengan hasil pengujian fungsionalitas menggunakan metode Black-Box Testing.
3. Berdasarkan hasil perbandingan performa model, YOLOv8n dengan resolusi masukan 640 × 640 piksel dan 416 × 416 piksel memperoleh performa deteksi terbaik dengan nilai Precision, Recall, dan F1-Score masing-masing sebesar 100%, serta mampu mendeteksi seluruh 145 objek hama pada 100 citra uji tanpa menghasilkan False Positive maupun False Negative. Sementara itu, YOLOv5s 320, YOLO26n 320, dan YOLOv8n 320 juga menunjukkan performa deteksi yang sangat baik dengan nilai F1-Score berturut-turut sebesar 99,66%, 99,31%, dan 98,64%. Hasil pengujian validasi model juga menunjukkan bahwa seluruh model memiliki nilai mAP@50 yang tinggi, sehingga seluruh model layak digunakan untuk proses deteksi hama pada tanaman hidroponik.
4. Dari sisi performa komputasi pada Raspberry Pi 4, model YOLOv8n 320 mampu dijalankan secara stabil dengan rata-rata penggunaan CPU sebesar 76,62%, RAM sebesar 20,24%, dan suhu prosesor sebesar 47,79°C selama

pengujian reliabilitas selama 6 jam. Selama periode tersebut tidak ditemukan system crash, memory leak, maupun thermal throttling, sehingga sistem dinyatakan mampu bekerja secara stabil dan layak digunakan untuk pemantauan hama secara berkelanjutan pada lingkungan hidroponik.

5.2 Saran

Berdasarkan keterbatasan sistem yang ditemukan selama proses penelitian dan pengujian, berikut adalah beberapa saran untuk pengembangan selanjutnya:

1. Peningkatan Kualitas dan Variasi Dataset

   Dataset yang digunakan sebaiknya diperluas dengan menambahkan variasi kondisi pencahayaan, seperti kondisi sangat terang, mendung, maupun pencahayaan rendah menggunakan bantuan lampu (LED ring atau flash). Selain itu, penambahan variasi sudut pengambilan gambar, jarak kamera, serta jumlah sampel setiap kelas hama diharapkan dapat meningkatkan kemampuan generalisasi model sehingga nilai precision, recall, dan F1-score menjadi lebih baik pada kondisi lapangan.

2. Peningkatan Kualitas Kamera

   Penelitian selanjutnya disarankan menggunakan kamera dengan resolusi yang lebih tinggi serta dilengkapi fitur autofocus atau kualitas lensa yang lebih baik. Hal tersebut dapat menghasilkan citra yang lebih tajam sehingga objek hama berukuran kecil, seperti *whitefly* dan *winged aphid*, dapat dideteksi dengan lebih akurat serta mengurangi kemungkinan terjadinya false positive maupun false negative.

3. Pengembangan Sistem untuk implementasi dengan skala lebih luas

   Sistem dapat dikembangkan agar mampu memantau area budidaya yang lebih luas dengan menambahkan jumlah lintasan gantry atau mengintegrasikan beberapa node Raspberry Pi dalam satu jaringan. Selain itu, penelitian selanjutnya dapat menambahkan fitur notifikasi otomatis dan analisis perkembangan populasi hama sebagai pendukung pengambilan keputusan dalam pengelolaan tanaman hidroponik.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

DAFTAR PUSTAKA

Adinda, D., Lubis, E.D.L.S., Harlin, F.I. dan Fevria, R., 2024. Budidaya Tanaman Bayam (*Amaranthus* spp.) Secara Hidroponik Menggunakan Sistem Nutrient Film Technique (NFT). *Prosiding Seminar Nasional Biologi*, 3(2), pp. 1111–1119.

Cantica, C., dkk., 2022. Sistem pemantauan dan kendali budidaya selada dalam ruangan pada media tanah berbasis Internet of Things. *Journal of Computing Engineering, System and Science*, 7(2), pp. 510–525.

Carneiro, T., Da Nóbrega, R.V.M., Nepomuceno, T., Bian, G., De Albuquerque, V.H.C. dan Rebouças Filho, P.P., 2018. Performance analysis of Google Colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, 6, pp. 61677–61685. doi: 10.1109/ACCESS.2018.2874767.

Chen, C.J., Huang, Y.Y., Li, Y.S., Chang, C.Y. dan Huang, Y.M., 2020. An AIoT based smart agricultural system for pests detection. *IEEE Access*, 8, pp. 180750–180761. doi: 10.1109/ACCESS.2020.3027471.

Dai, M., Dorjoy, M.M.H., Miao, H. dan Zhang, S., 2023. A new pest detection method based on improved YOLOv5m. *Insects*, 14(1), p. 54. Tersedia di: https://doi.org/10.3390/insects14010054.

Dinata, R., 2023. Implementasi algoritma YOLOv5 untuk deteksi hama belalang pada tanaman selada hidroponik. *Jurnal Teknologi Informasi dan Komputer*, 8(2), pp. 101–110.

Fandisyah, F., dkk., 2021. Deteksi Kapal di Laut Indonesia Menggunakan YOLOv3. *Jurnal Sains dan Seni ITS*, 10(1), pp. 45–52. doi: 10.12962/j23373520.v10i1.59312.

Fecko, B., Vince, T. dan Darák, L., 2020. Design and realization of a feeder part of the automated production line. *Electromechanical and Energy Saving Systems*, 2(50), pp. 53–60. doi: 10.30929/2072-2052.2020.2.50.53-60.

Goodfellow, I., Bengio, Y. dan Courville, A., 2016. *Deep Learning*. Cambridge: MIT Press.

Guo, Z., dkk., 2022. MSFT-YOLO: Improved YOLOv5 Based on Transformer for Detecting Defects of Steel Surface. *Sensors*, 22(9), p. 3467. doi: 10.3390/s22093467.

Hakim, A., Srivastava, A.K., Hamza, A., Owais, M., Habib-ur-Rahman, M., Qadri, S., Qayyum, M.A., Khan, F.Z.A., Mahmood, M.T. dan Gaiser, T., 2025. Yolo-pest: an optimized YoloV8x for detection of small insect pests using smart traps. *Scientific Reports*, 15, art. 14029. doi: 10.1038/s41598-025-97825-3.

Hidayanti, L. dan Kartika, T., 2019. Pengaruh Nutrisi AB Mix Terhadap Pertumbuhan Tanaman Bayam Merah (*Amaranthus tricolor* L.) secara Hidroponik. *Sainmatika: Jurnal Ilmiah Matematika dan Ilmu Pengetahuan Alam*, 16(2), pp. 166–175. doi: 10.31851/sainmatika.v16i2.3214.

Jocher, G., Chaurasia, A. dan Qiu, J., 2023. *YOLO by Ultralytics (Version 8.0)* https://github.com/ultralytics/ultralytics.

Kardinan, A., 2011. Penggunaan Pestisida Nabati Sebagai Kearifan Lokal Dalam Pengendalian Hama Tanaman Menuju Sistem Pertanian Organik. *Pengembangan Inovasi Pertanian*, 4(4), hlm. 262–278.

Kartika, V.S., Aji, A.F., Kusumastuti, S., Rochmanto, R.A., Hidayat, S.N. dan Sadewa, A.P., 2025. Penerapan Smart Greenhouse Hidroponik Berbasis IoT Menggunakan EBT di Nurus Sunnah Farm. *ABSYARA: Jurnal Pengabdian Pada Masyarakat*, 6(1), pp. 52–60. doi: 10.29408/ab.v6i1.28692.

Lazarevich, I., dkk., 2023. YOLOBench: Benchmarking Efficient Object Detectors on Embedded Systems. Dalam: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. Paris, Prancis, 2-6 Oktober 2023. New York: IEEE, pp. 1461-1470.

Mahmud, M. dan Aprizal, Y., 2022. Penerapan Metode Rekayasa Sistem Jaringan Komputer dalam Merancang Blueprint Jaringan Komputer (Studi Kasus: Hotel Maxone Palembang). *Teknomatika*, 12(1), hlm. 13-22.

Mubarok, S., dkk., 2021. Pertumbuhan dan hasil dua kultivar bayam pada sistem budidaya hidroponik untuk lahan sempit. *Jurnal Kultivasi*, 20(3), pp. 196-202. doi: 10.24198/kultivasi.v20i3.35123.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Mureithi, D.M., Fiaboe, K.K.M., Ekesi, S. dan Meyhöfer, R., 2017. Important arthropod pests on leafy Amaranth (*Amaranthus viridis*, *A. tricolor* and *A. blitum*) and broad-leafed African nightshade (*Solanum scabrum*) with a special focus on host-plant ranges. *African Journal of Horticultural Science*, 11, pp. 1–17.

Murshed, M.S., Murphy, C., Hou, D., Khan, N.A., Ananthanarayanan, G. dan Hussain, F., 2019. Machine Learning at the Network Edge: A Survey. *ACM Computing Surveys (CSUR)*, 54(8), hlm. 1-37. doi: 10.1145/3469029.

Myers, G.J., Sandler, C. dan Badgett, T., 2011. *The Art of Software Testing*. Edisi ke-3. Hoboken: John Wiley & Sons.

Nihui, 2017. *ncnn: A high-performance neural network inference computing framework optimized for mobile platforms*. https://github.com/Tencent/ncnn.

Nugroho, B., Fajri, Pratama, B.M., dkk., 2025. Implementation of sparrow pest detection using YOLOv8 method on Raspberry Pi and Google Coral USB accelerator. *Sebatik*, 29(1), pp. 129–135. doi: 10.46984/sebatik.v29i1.2558.

OmniVision Technologies, 2012. *OV5647 color CMOS 5-megapixel image sensor datasheet*. https://cdn.sparkfun.com/datasheets/Dev/RaspberryPi/ov5647_full.pdf.

Pallets Projects, 2024. *Flask documentation (3.1.x)*. https://flask.palletsprojects.com/.

Parihar, Y., 2019. Internet of Things and Nodemcu: a Review of Use of Nodemcu ESP8266 in IoT Products. *Journal of Emerging Technologies and Innovative Research (JETIR)*, 6(6), pp. 104-112. doi: 10.13140/RG.2.2.34456.75525.

Parlika, R., Erayanti, A.E., Orissa, D.F., Ummam, M.A.I. dan Abrori, M.R., 2021. Studi Literatur Kaitan Antara Rekayasa Kebutuhan dan Rekayasa Sistem. *Jurnal Sistem Informasi dan Bisnis Cerdas (SIBC)*, 14(1), hlm. 37-44.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Patterson, A.E., 2018. Nominal stiffness of GT-2 rubber-fiberglass timing belts for dynamic system modeling and design. *Robotics*, 7(4), p. 75. doi: 10.3390/robotics7040075.

Prechelt, L., 2012. Early Stopping—But When?. Dalam: G. Montavon, G.B. Orr dan K.R. Müller, ed. *Neural Networks: Tricks of the Trade*. Edisi ke-2. Berlin: Springer, hal. 53–67.

Raspberry Pi Ltd, 2024. *Camera hardware documentation*. https://www.raspberrypi.com/documentation/accessories/camera.html.

Redmon, J. dan Farhadi, A., 2018. YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*.

Sakib, S., dkk., 2018. An Overview of Convolutional Neural Network: Its Architecture and Applications. *Preprints 2018*, 2018080116. doi: 10.20944/preprints201808.0116.v1.

Sanders, J. dan Kandrot, E., 2010. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Boston: Addison-Wesley Professional.

Sarwono, A.E. dan Handayani, A., 2021. *Metode Kuantitatif*. Surakarta: Unisri Press.

Seni, A., 2018. Insect pests of amaranthus and their management. *International Journal of Environment, Agriculture and Biotechnology*, 3(3), pp. 1100–1103. doi: 10.22161/ijeab/3.3.50.

Setiawan, F.B., dkk., 2021. Pattern Recognition untuk Deteksi Posisi pada AGV Berbasis Raspberry Pi. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, 10(1), pp. 76-83. doi: 10.22146/jnteti.v10i1.652.

Shorten, C. dan Khoshgoftaar, T.M., 2019. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), pp. 1-48. doi: 10.1186/s40537-019-0197-0.

Terven, J., Córdova-Esparza, D. dan Romero-González, J.A., 2023. A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond. *arXiv preprint arXiv:2304.00501*.

Varghese, B., Wang, N., Barbhuiya, S., Kilpatrick, P. dan Nikolopoulos, D.S., 2019. Challenges and Opportunities in Edge Computing. Dalam: *2019*

IEEE International Conference on Smart Computing (SMARTCOMP). Washington, DC, USA, 12-15 Juni 2019. New York: IEEE, pp. 20-26.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

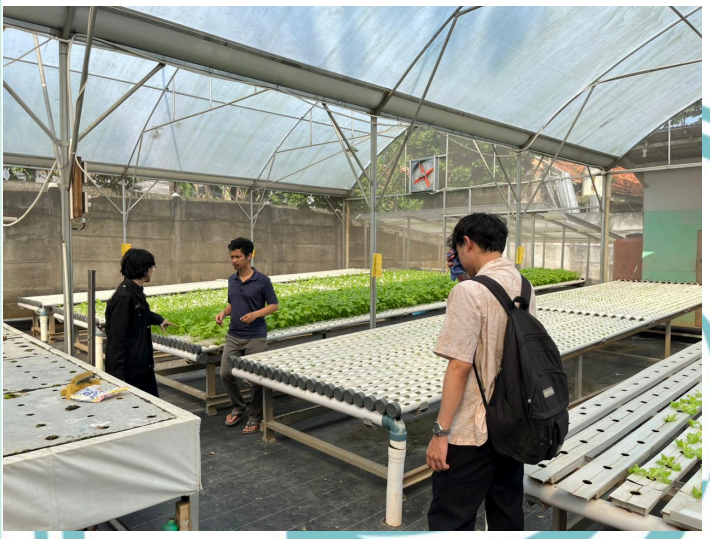
DAFTAR RIWAYAT HIDUP

Alfarizki Nurachman

Lahir pada tahun 2003 dan saat ini berdomisili di Jakarta. Penulis merupakan anak ketiga dari tiga bersaudara. Penulis menyelesaikan pendidikan Sekolah Dasar (SD) pada tahun 2015, Sekolah Menengah Pertama (SMP) pada tahun 2018, dan Sekolah Menengah Atas (SMA) pada tahun 2021. Pada tahun 2022, penulis berkesempatan untuk melanjutkan pendidikan Diploma IV pada Program Studi Teknik Multimedia dan Jaringan, Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta. Selama menempuh pendidikan, penulis memiliki ketertarikan pada bidang *computer vision*, *artificial intelligence*, *Internet of Things* (IoT), dan pengembangan sistem berbasis *embedded system*.

LAMPIRAN

Lampiran 1. Dokumentasi Observasi dan Implementasi



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Lampiran 2. Dokumentasi Pengujian Model YOLOv8n dan YOLOv5s

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Lampiran 3. Raw data pengujian reliabilitas

No	CPU Usage	RAM Usage	Suhu
1	82.6	22.3	44.8
2	82.9	22.4	44.8
3	84.8	22.5	44.3
4	85.7	22.5	45.8
5	85.4	22.3	44.8
6	83.9	22.2	44.3
7	85.6	22.3	44.8
8	84.4	22.1	44.8
9	84.8	22	45.8
10	84.5	22	45.3
11	85.8	21.8	46.3
12	83.7	21.9	46.7
13	84.9	21.8	45.3
14	80.9	21.7	46.3
15	83.7	21.8	46.3
16	84.1	21.7	45.3
17	84.3	21.7	46.3
18	85.5	21.7	46.7

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

19	84.5	21.7	47.2
20	82.2	21.7	45.8
21	84	21.7	46.7
22	83.5	21.7	46.7
23	85.3	21.7	46.7
24	85.1	22.2	45.3
25	84.4	22.1	45.8
26	84.3	21.8	46.3
27	86.2	21.9	46.3
28	86.6	21.8	45.3
29	84.7	21.8	46.3
30	85	21.8	46.7
31	85.5	21.8	46.7
32	84.3	21.7	46.3
33	84.5	21.7	45.8
34	85.2	21.7	47.2
35	85	21.7	46.7
36	84.9	21.7	46.3
37	84.8	21.6	46.3
38	85.1	21.6	47.7
39	84.9	21.6	47.7

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

40	85.4	21.6	46.7
41	85	21.6	47.7
42	85.6	21.6	47.2
43	84.5	21.6	47.2
44	86	21.6	47.2
45	84.3	21.6	46.7
46	84.6	21.6	47.7
47	85.3	21.6	47.7
48	85.4	21.6	46.7
49	85.2	21.6	46.7
50	84.2	21.5	46.7
51	85.8	21.5	46.3
52	85.9	21.5	46.7
53	85.6	21.5	47.7
54	84.7	21.5	47.7
55	84.7	21.5	46.7
56	84.5	21.5	47.2
57	86.2	21.5	48.2
58	85.3	21.5	48.2
59	84.6	21.5	47.7
60	85.1	21.5	47.7

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

61	85.5	21.5	47.2
62	84.7	21.5	48.2
63	85.5	21.5	47.7
64	84	21.5	47.2
65	86.4	21.5	47.2
66	85.8	21.5	47.7
67	84.5	21.5	48.2
68	85.1	21.5	46.7
69	84.4	21.5	47.2
70	85.4	21.5	47.2
71	85.1	21.5	49.2
72	84.7	21.5	46.7
73	85.2	21.5	48.2
74	85.7	21.5	47.7
75	86	21.5	48.2
76	84.6	21.5	47.2
77	85.4	21.5	48.2
78	85.2	21.5	48.2
79	85.7	21.5	47.2
80	84.8	21.5	48.2
81	85.2	21.5	48.7

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

82	85	21.5	47.2
83	84.4	21.5	48.7
84	84.3	21.5	47.7
85	85.5	21.5	49.2
86	85.2	21.5	48.7
87	84.7	21.5	47.7
88	84.6	21.5	48.2
89	85	21.5	48.2
90	84.4	21.5	47.2
91	84.4	21.5	48.2
92	84.6	21.5	48.7
93	85.6	21.5	47.7
94	85.2	21.5	47.7
95	84.9	21.5	48.7
96	84.8	21.5	48.2
97	84.9	21.5	48.7
98	84.5	21.5	48.2
99	85.7	21.5	47.7
100	85.1	21.5	48.2

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Lampiran 4. Pengujian Performa Perangkat dengan Model YOLOv8n 640

Timestamp	FPS	CPU Usage	RAM Usage	Suhu	Waktu Inferensi
2026-06-23T11:59:44.170Z	1.43	82.6	22.3	44.8	0.7052
2026-06-23T11:59:45.635Z	1.39	82.9	22.4	44.8	0.7011
2026-06-23T11:59:47.022Z	1.37	84.8	22.5	44.3	0.6826
2026-06-23T11:59:48.385Z	1.44	85.7	22.5	45.8	0.6603
2026-06-23T11:59:49.756Z	1.47	85.4	22.3	44.8	0.673
2026-06-23T11:59:51.151Z	1.45	83.9	22.2	44.3	0.6721
2026-06-23T11:59:52.532Z	1.44	85.6	22.3	44.8	0.652
2026-06-23T11:59:53.907Z	1.45	84.4	22.1	44.8	0.6763
2026-06-23T11:59:55.269Z	1.45	84.8	22	45.8	0.6598
2026-06-23T11:59:56.6	1.47	84.5	22	45.3	0.6724

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

57Z					
2026-06-23T11:59:58.032Z	1.44	85.8	21.8	46.3	0.6602
2026-06-23T11:59:59.429Z	1.46	83.7	21.9	46.7	0.6598
2026-06-23T12:00:00.782Z	1.43	84.9	21.8	45.3	0.6647
2026-06-23T12:00:02.232Z	1.48	80.9	21.7	46.3	0.7185
2026-06-23T12:00:03.629Z	1.38	83.7	21.8	46.3	0.7029
2026-06-23T12:00:05.006Z	1.43	84.1	21.7	45.3	0.66
2026-06-23T12:00:06.383Z	1.46	84.3	21.7	46.3	0.6701
2026-06-23T12:00:07.731Z	1.45	85.5	21.7	46.7	0.6547
2026-06-23T12:00:09.117Z	1.48	84.5	21.7	47.2	0.6644
2026-06-23T12:00:10.515Z	1.44	82.2	21.7	45.8	0.6694

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2026-06-23T12:00:11.895Z	1.43	84	21.7	46.7	0.6776
2026-06-23T12:00:13.306Z	1.45	83.5	21.7	46.7	0.6657
2026-06-23T12:00:14.692Z	1.41	85.3	21.7	46.7	0.6709
2026-06-23T12:01:47.165Z	1.46	85.1	22.2	45.3	0.6633
2026-06-23T12:01:48.552Z	1.45	84.4	22.1	45.8	0.6778
2026-06-23T12:01:49.923Z	1.44	84.3	21.8	46.3	0.671
2026-06-23T12:01:51.279Z	1.48	86.2	21.9	46.3	0.657
2026-06-23T12:01:52.629Z	1.47	86.6	21.8	45.3	0.6554
2026-06-23T12:01:53.987Z	1.48	84.7	21.8	46.3	0.6612
2026-06-23T12:01:55.355Z	1.48	85	21.8	46.7	0.6579
2026-06-23T12:01:56.7	1.45	85.5	21.8	46.7	0.6664

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

15Z					
2026-06-23T12:01:58.080Z	1.48	84.3	21.7	46.3	0.6627
2026-06-23T12:01:59.457Z	1.47	84.5	21.7	45.8	0.653
2026-06-23T12:02:00.816Z	1.45	85.2	21.7	47.2	0.6587
2026-06-23T12:02:02.188Z	1.46	85	21.7	46.7	0.6718
2026-06-23T12:02:03.550Z	1.48	84.9	21.7	46.3	0.6503
2026-06-23T12:02:04.910Z	1.47	84.8	21.6	46.3	0.655
2026-06-23T12:02:06.279Z	1.46	85.1	21.6	47.7	0.6587
2026-06-23T12:02:07.647Z	1.45	84.9	21.6	47.7	0.6705
2026-06-23T12:02:09.002Z	1.48	85.4	21.6	46.7	0.6584
2026-06-23T12:02:10.369Z	1.46	85	21.6	47.7	0.6706

2026-06-23T12:02:11.721Z	1.47	85.6	21.6	47.2	0.6615
2026-06-23T12:02:13.103Z	1.46	84.5	21.6	47.2	0.6771
2026-06-23T12:02:14.450Z	1.47	86	21.6	47.2	0.6515
2026-06-23T12:02:15.814Z	1.46	84.3	21.6	46.7	0.6705
2026-06-23T12:02:17.181Z	1.48	84.6	21.6	47.7	0.6608
2026-06-23T12:02:18.551Z	1.47	85.3	21.6	47.7	0.6528
2026-06-23T12:02:19.914Z	1.45	85.4	21.6	46.7	0.6628
2026-06-23T12:02:21.265Z	1.48	85.2	21.6	46.7	0.6552
2026-06-23T12:02:34.176Z	1.48	84.2	21.5	46.7	0.658

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Lampiran 5. Pengujian Performa Perangkat dengan Model YOLOv8n 416

Timestamp	FPS	CPU Usage	RAM Usage	Suhu	Waktu Inferensi
2026-06-23T13:00:15.907Z	2.91	79.6	20.8	46.7	0.3151
2026-06-23T13:00:16.920Z	2.96	79.5	20.9	47.2	0.3185
2026-06-23T13:00:17.952Z	2.95	78.6	20.9	47.7	0.3243
2026-06-23T13:00:18.961Z	2.95	79.2	20.9	47.7	0.3166
2026-06-23T13:00:19.988Z	2.96	79.3	20.9	47.2	0.3159
2026-06-23T13:00:21.012Z	2.9	79.1	20.8	46.3	0.3261
2026-06-23T13:00:22.030Z	2.96	80.1	20.8	47.2	0.3157
2026-06-23T13:00:23.042Z	2.93	79.5	20.7	47.7	0.3218
2026-06-23T13:00:24.052Z	2.98	79.7	20.6	47.7	0.3169
2026-06-23T13:00:25.083Z	2.94	78.8	20.5	46.7	0.3212
2026-06-23T13:00:26.091Z	2.96	79.5	20.5	46.7	0.3129
2026-06-23T13:00:27.107Z	2.96	80	20.4	48.7	0.3188
2026-06-23	2.9	78.6	20.4	47.7	0.3186

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

T13:00:28.139Z					
2026-06-23T13:00:29.154Z	2.94	78	20.3	47.7	0.32
2026-06-23T13:00:30.171Z	2.98	80.5	20.3	46.7	0.3141
2026-06-23T13:00:31.182Z	2.96	80.4	20.3	46.7	0.3145
2026-06-23T13:00:32.192Z	2.95	79.9	20.2	49.2	0.3182
2026-06-23T13:00:33.211Z	2.95	78.7	20.2	46.7	0.3233
2026-06-23T13:00:34.223Z	2.98	79.5	20.2	48.7	0.3154
2026-06-23T13:00:35.238Z	2.95	80.5	20.2	47.7	0.3215
2026-06-23T13:00:36.258Z	2.96	78.7	20.2	46.7	0.3205
2026-06-23T13:00:37.279Z	2.9	78.6	20.2	47.2	0.3254
2026-06-23T13:00:38.310Z	2.92	78.2	20.2	46.7	0.3214
2026-06-23T13:00:39.320Z	2.96	79.8	20.2	48.2	0.3217
2026-06-23T13:00:40.334Z	2.96	79.8	20.2	48.2	0.3242
2026-06-23T13:00:41.339Z	3	80.1	20.1	48.2	0.3139

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2026-06-23T13:00:42.364Z	2.93	78.7	20.1	48.2	0.3282
2026-06-23T13:00:43.376Z	2.96	80.1	20.1	48.2	0.3235
2026-06-23T13:00:44.385Z	2.96	79.8	20.1	48.7	0.321
2026-06-23T13:00:45.407Z	2.94	79.8	20.1	49.2	0.3237
2026-06-23T13:00:46.420Z	2.97	79.5	20.1	47.2	0.3223
2026-06-23T13:00:47.432Z	2.97	80.2	20.1	47.7	0.3181
2026-06-23T13:00:48.456Z	2.93	79.1	20.1	48.7	0.3234
2026-06-23T13:00:49.466Z	2.96	79.4	20.1	47.7	0.3189
2026-06-23T13:00:50.477Z	2.97	79.6	20.1	47.7	0.3171
2026-06-23T13:00:51.500Z	2.94	78.1	20.1	48.2	0.3284
2026-06-23T13:00:52.513Z	2.95	79.1	20.1	48.2	0.3183
2026-06-23T13:00:53.520Z	2.98	80.3	20.1	48.2	0.3171
2026-06-23T13:00:54.544Z	2.98	77.9	20.1	48.2	0.3218
2026-06-23T13:00:55.5	2.94	80.8	20.1	48.2	0.3173

53Z					
2026-06-23T13:00:56.566Z	2.97	79.2	20.1	48.2	0.3234
2026-06-23T13:00:57.597Z	2.94	78.4	20.1	48.2	0.3205
2026-06-23T13:00:58.604Z	2.93	79.6	20.1	47.7	0.3151
2026-06-23T13:00:59.620Z	2.96	79.5	20.1	49.2	0.3149
2026-06-23T13:01:00.633Z	2.97	79.4	20.1	47.7	0.3142
2026-06-23T13:01:01.644Z	2.95	79	20.1	48.7	0.3189
2026-06-23T13:01:02.658Z	2.96	79.7	20.1	49.2	0.3196
2026-06-23T13:01:03.668Z	2.97	79.7	20.1	47.7	0.3164
2026-06-23T13:01:04.683Z	2.96	79.5	20.1	48.2	0.3184
2026-06-23T13:01:05.704Z	2.95	80	20.1	47.7	0.3204

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Lampiran 6. Pengujian Performa Perangkat dengan Model YOLOv8n 320

Timestamp	FPS	CPU Usage	RAM Usage	Suhu	Waktu Inferensi
2026-06-23T15:01:50.568Z	4.38	75.3	20.5	45.3	0.2073
2026-06-23T15:01:51.696Z	4.39	75.9	20.5	44.3	0.2039
2026-06-23T15:01:52.819Z	4.45	76.1	20.6	44.8	0.2046
2026-06-23T15:01:53.949Z	4.45	75.8	20.6	45.8	0.2051
2026-06-23T15:01:55.102Z	4.43	75.1	20.6	45.3	0.2171
2026-06-23T15:01:56.225Z	4.4	74.9	20.6	45.3	0.2147
2026-06-23T15:01:57.339Z	4.42	77	20.5	45.3	0.2042
2026-06-23T15:01:58.458Z	4.46	76.9	20.4	45.8	0.203
2026-06-23T15:01:59.598Z	4.41	75.3	20.4	45.8	0.2047
2026-06-23T15:02:00.7	4.43	76.7	20.4	46.3	0.2009

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

18Z					
2026-06-23T15:02:01.840Z	4.47	75.5	20.3	44.8	0.203
2026-06-23T15:02:02.979Z	4.44	75.4	20.3	44.8	0.2057
2026-06-23T15:02:04.101Z	4.44	75.4	20.2	45.3	0.2059
2026-06-23T15:02:05.237Z	4.4	76.8	20.1	45.3	0.2101
2026-06-23T15:02:06.364Z	4.43	76	20.1	46.7	0.2035
2026-06-23T15:02:07.491Z	4.42	76	20	46.7	0.2032
2026-06-23T15:02:08.626Z	4.43	75	20	46.7	0.2037
2026-06-23T15:02:09.765Z	4.4	74.2	19.9	46.3	0.2017
2026-06-23T15:02:10.895Z	4.39	75.5	19.9	46.7	0.2026
2026-06-23T15:02:12.031Z	4.45	75.1	19.9	46.3	0.2045

2026-06-23T15:02:13.155Z	4.43	75.2	19.9	44.8	0.2043
2026-06-23T15:02:14.295Z	4.42	74.4	19.9	45.8	0.2061
2026-06-23T15:02:15.445Z	4.38	75.7	19.9	45.3	0.2116
2026-06-23T15:02:16.577Z	4.36	74.7	19.9	46.7	0.2093
2026-06-23T15:02:17.703Z	4.46	75.8	19.9	46.3	0.2106
2026-06-23T15:02:18.822Z	4.44	76.8	19.9	46.7	0.2072
2026-06-23T15:02:19.946Z	4.48	75.9	19.9	46.3	0.2113
2026-06-23T15:02:21.073Z	4.41	76.1	19.9	47.2	0.2037
2026-06-23T15:02:22.221Z	4.38	74.4	19.9	46.3	0.2149
2026-06-23T15:02:23.344Z	4.44	76.2	19.9	46.3	0.212
2026-06-23T15:02:24.4	4.45	77.2	19.8	46.7	0.2072

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

58Z					
2026-06-23T15:02:25.604Z	4.39	74.7	19.9	46.3	0.2018
2026-06-23T15:02:26.737Z	4.38	75.2	19.8	46.7	0.2029
2026-06-23T15:02:27.864Z	4.44	75.2	19.8	46.3	0.201
2026-06-23T15:02:28.997Z	4.45	75.9	19.8	46.3	0.2031
2026-06-23T15:02:30.133Z	4.38	75.6	19.8	45.8	0.2081
2026-06-23T15:02:41.426Z	4.42	75	20.4	45.8	0.2003
2026-06-23T15:02:42.554Z	4.43	74.6	20.4	45.3	0.2042
2026-06-23T15:02:43.680Z	4.44	76.1	20.3	45.8	0.2055
2026-06-23T15:02:44.828Z	4.39	74.2	20.2	45.8	0.2194
2026-06-23T15:02:45.978Z	4.33	75.1	20.2	47.2	0.2092

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2026-06-23T15:02:47.112Z	4.42	75.1	20.1	46.3	0.2099
2026-06-23T15:02:48.253Z	4.36	75	20.1	45.8	0.2149
2026-06-23T15:02:49.373Z	4.46	76.2	20.1	45.8	0.2041
2026-06-23T15:02:50.511Z	4.4	74.7	20	46.3	0.2062
2026-06-23T15:02:51.654Z	4.36	75.5	20	47.2	0.2163
2026-06-23T15:02:52.795Z	4.4	74.4	20	46.3	0.2047
2026-06-23T15:02:53.917Z	4.44	75.5	20	46.7	0.2023
2026-06-23T15:02:55.066Z	4.41	74.9	19.9	46.7	0.2102
2026-06-23T15:02:56.204Z	4.35	75.9	19.9	46.3	0.2129

Lampiran 7. Pengujian Performa Perangkat dengan Model YOLOv5s 320

Timestamp	FPS	CPU Usage	RAM Usage	Suhu	Waktu Inferensi
2026-06-23T14:01:09.635Z	0.84	72	20.8	46.3	0.1158
2026-06-23T14:01:10.641Z	0.87	70.5	20.8	45.3	0.11272
2026-06-23T14:01:11.643Z	0.87	70.8	20.8	45.8	0.11272
2026-06-23T14:01:12.648Z	0.83	70.7	20.9	46.7	0.11795
2026-06-23T14:01:13.650Z	0.82	71.9	20.9	46.3	0.11975
2026-06-23T14:01:14.653Z	0.83	71.6	20.9	46.7	0.11827
2026-06-23T14:01:15.655Z	0.82	71.7	20.9	46.3	0.12013
2026-06-23T14:01:16.659Z	0.83	71	20.9	46.7	0.11832
2026-06-23T14:01:17.664Z	0.83	71.9	20.9	45.8	0.11832
2026-06-23T14:01:18.667Z	0.83	79.2	21.5	46.3	0.11919
2026-06-23T14:01:19.669Z	0.69	71.7	21.3	47.2	0.14299
2026-06-23T14:01:20.671Z	0.84	72.5	21.3	46.7	0.11693
2026-06-23T14:01:21.674Z	0.84	71.5	21.1	46.7	0.11693
2026-06-23T14:01:22.676Z	0.84	70.6	21	46.3	0.11739

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2026-06-23T14:01:23.679Z	0.83	72.4	20.9	46.7	0.11873
2026-06-23T14:01:24.681Z	0.83	72.6	20.6	46.7	0.11871
2026-06-23T14:01:25.683Z	0.83	71.5	20.4	46.7	0.11857
2026-06-23T14:01:26.686Z	0.84	71.8	20.6	46.3	0.11743
2026-06-23T14:01:27.689Z	0.84	70	20.4	46.7	0.11743
2026-06-23T14:01:28.696Z	0.83	70.7	20.5	46.3	0.11846
2026-06-23T14:01:29.698Z	0.85	71.6	20.7	47.7	0.11636
2026-06-23T14:01:30.708Z	0.84	72.8	20.5	47.7	0.11748
2026-06-23T14:01:31.712Z	0.86	72.3	20.4	46.7	0.11446
2026-06-23T14:01:32.718Z	0.85	70.8	20.3	47.7	0.116
2026-06-23T14:01:33.720Z	0.85	76.1	21.2	46.7	0.116
2026-06-23T14:01:34.722Z	0.76	68.7	21.2	47.7	0.12875
2026-06-23T14:01:35.725Z	0.82	71	21.2	47.2	0.12017
2026-06-23T14:01:36.730Z	0.83	69.9	21.2	47.2	0.1179
2026-06-23T14:01:37.732Z	0.83	70.6	21.2	46.7	0.11876
2026-06-23T1	0.86	70.8	20.9	48.2	0.11468

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4:01:38.737Z					
2026-06-23T14:01:39.743Z	0.94	65.8	20.8	47.2	0.10416
2026-06-23T14:01:40.746Z	0.94	72.1	20.8	47.2	0.10416
2026-06-23T14:01:41.748Z	0.85	69.7	20.8	47.2	0.11503
2026-06-23T14:01:42.750Z	0.85	70.4	20.7	48.2	0.11557
2026-06-23T14:01:43.753Z	0.86	70.9	20.7	47.2	0.11436
2026-06-23T14:01:44.756Z	0.84	72.3	20.6	47.2	0.11721
2026-06-23T14:01:45.758Z	0.83	69.2	20.8	45.8	0.11856
2026-06-23T14:01:46.760Z	0.83	67.6	20.8	48.2	0.11835
2026-06-23T14:01:47.765Z	0.83	70.7	20.8	46.7	0.11835
2026-06-23T14:01:48.776Z	0.84	70.7	20.8	48.2	0.11621
2026-06-23T14:01:49.779Z	0.85	70.7	20.6	46.7	0.11593
2026-06-23T14:01:50.782Z	0.85	70.3	20.5	47.2	0.11505
2026-06-23T14:01:51.787Z	0.83	69.9	20.6	46.7	0.11769
2026-06-23T14:01:52.790Z	0.85	71.5	20.6	47.7	0.11591
2026-06-23T14:01:53.798Z	0.85	70.1	20.7	47.2	0.11591

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2026-06-23T14:01:54.802Z	0.82	69.8	20.8	47.2	0.11963
2026-06-23T14:01:55.804Z	0.83	69.8	20.6	46.3	0.11838
2026-06-23T14:01:56.812Z	0.82	70	20.9	47.2	0.11991
2026-06-23T14:02:09.842Z	0.84	68.7	20.2	46.3	0.11764
2026-06-23T14:02:10.849Z	0.84	69.4	20.6	47.2	0.11634

Lampiran 8. Source Code Skrip Raspberry Pi

```
import argparse
import base64
import cv2
import csv
import logging
import os
import threading
import time
import queue
import signal
import psutil
import socket
from datetime import datetime, timezone
from flask import Flask, Response
from picamera2 import Picamera2
from clients import create_protocol_client
from inference.processor import InferenceEngine
from hardware.serial_worker import esp8266_usb_worker

# ——— LOGGING CONFIGURATION ———————————————————————————

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s [%(levelname)s] %(message)s",
    datefmt="%H:%M:%S",
)
```

```
log = logging.getLogger("LokataniEdge")

app = Flask(__name__)

MODEL_REGISTRY = {
    "v8n_640": {"path": "models/best_v8n_ncnn_model",
"imgsz": 640},
    "v8n_416": {"path": "models/yolov8n_best_416_ncnn_model",
"imgsz": 416},
    "v8n_320": {"path": "models/yolov8n_best_320_ncnn_model",
"imgsz": 320},
    "v5s_320": {"path": "models/yolov5s_best_320.pt",
"imgsz": 320},
}

# ─── ARGUMENT PARSER
──────────────────────────────────────────────
parser = argparse.ArgumentParser(description="Lokatani
Integrated Edge Stream & Cloud Gateway.")
parser.add_argument("--protocol", type=str, choices=["HTTP",
"WS", "MQTT"], default="HTTP", help="Protokol komunikasi
cloud")
parser.add_argument("--host", type=str, default="localhost",
help="IP/Hostname Cloud Backend")
parser.add_argument("--http-port", type=int, default=3000,
help="Port HTTP API")
parser.add_argument("--ws-port", type=int, default=8080,
help="Port WebSocket")
parser.add_argument("--broker", type=str, default=None,
help="MQTT Broker URL")
parser.add_argument("--api-key", type=str,
default="lokatani-edge-device-key", help="API Key autentikasi
device")
parser.add_argument("--serial-port", type=str,
default="/dev/ttyUSB0", help="Port USB Serial ESP8266")
parser.add_argument("--min-confidence", type=float,
default=0.25, help="Threshold minimum confidence YOLO")
parser.add_argument("--run-duration", type=int, default=0,
help="Durasi jalan dalam detik (0 = selamanya)")

parser.add_argument(
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
    "--model",
    choices=list(MODEL_REGISTRY.keys()),
    default="v8n_320",
    help="Pilih model yang dipakai untuk inferensi (default:
v8n_320)"
)

parser.add_argument(
    "--imgsz",
    type=int,
    default=None,
    help="Override ukuran input manual (opsional, default ikut
konfigurasi model)"
)

args, unknown = parser.parse_known_args()

model_cfg = MODEL_REGISTRY[args.model]
model_path = model_cfg["path"]
final_imgsz = args.imgsz if args.imgsz is not None else
model_cfg["imgsz"]

# ——— CONFIGURATION & STORAGE
————————————————————————————————

RUN_TAG = args.model
BASE_OUTPUT_DIR = os.path.join("hasil_pengujian", RUN_TAG)
LOG_FILE = os.path.join(BASE_OUTPUT_DIR,
"hasil_pengujian_hama.csv")
IMAGE_SAVE_DIR = os.path.join(BASE_OUTPUT_DIR, "bukti_hama")
RAW_DATASET_DIR = os.path.join(BASE_OUTPUT_DIR, "raw_dataset")
PERF_LOG_FILE = os.path.join(BASE_OUTPUT_DIR,
"performa_sistem.csv")
ESP_CONNECT_TIMEOUT = 5

os.makedirs(IMAGE_SAVE_DIR, exist_ok=True)
os.makedirs(RAW_DATASET_DIR, exist_ok=True)

if not os.path.exists(LOG_FILE):
    with open(LOG_FILE, mode='w', newline='') as file:
        writer = csv.writer(file)
        writer.writerow([
```

```
            "Timestamp",
            "Session_ID",
            "File_Foto",
            "AI_Winged_Aphid",
            "AI_Whitefly",
            "Real_Winged_Aphid",
            "Real_Whitefly",
            "Waktu_Inferensi(s)",
            "Status_TP_FP_FN"
        ])

# ─── DATA SHARE REPOSITORY (THREAD-SAFE CHANNELS) ──────────────────────
global_frame_raw = None
global_frame_annotated = None

state_lock = threading.Lock()
frame_lock = threading.Lock()
shutdown_event = threading.Event()
_shutdown_in_progress = threading.Event()

io_queue = queue.Queue(maxsize=50)

shared_state = {
    "esp_hardware_ready": False,
    "esp_handshake_done": False,
    "is_scanning": False,
    "session_id":
f"LOCAL_BOOT_{datetime.now().strftime('%Y%m%d_%H%M%S')}",
    "internet_connected": False,
    "is_cloud_active": False,
    "last_inf_time": 0.0,
    "current_cpu_util": 0.0,
    "current_ram_util": 0.0,
    "current_cpu_temp": 25.0,
}

# ─── HARDWARE & MODEL INITIALIZATION ──────────────────────────────────
picam2 = Picamera2()
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
config = picam2.create_preview_configuration(main={"format":
"BGR888", "size": (1280, 960)})
picam2.configure(config)
picam2.set_controls({
    "AeConstraintMode": 1,
    "AeExposureMode": 0,
    "AwbMode": 0,
    "Saturation": 1.3,
    "Contrast": 1.2,
    "Sharpness": 2.0,
})
picam2.start()

engine = InferenceEngine(model_path, args.min_confidence,
imgsz=final_imgsz)
log.info(f"🧠 Model [{args.model}] dimuat | path={model_path}
| imgsz={final_imgsz}")

# ——— CORE CAMERA BACKGROUND WORKER
————————————————————————————

def camera_stream_worker():
    global global_frame_raw
    log.info("🚀 Background Camera Worker started.")
    while not shutdown_event.is_set():
        try:
            frame_rgb = picam2.capture_array()
            with frame_lock:
                global_frame_raw = frame_rgb
            time.sleep(0.03)
        except Exception as e:
            log.error(f"Error pada thread kamera: {e}")
            time.sleep(1)

def check_internet_connection(host="1.1.1.1", port=53,
timeout=3.0) -> bool:
    try:
        socket.setdefaulttimeout(timeout)
        with socket.create_connection((host, port),
timeout=timeout):
            return True
    except OSError:
```

```
        return False

def network_watchdog_worker():
    log.info("🌐 Network Access Watchdog Worker started.")
    last_state = False

    initial_check = check_internet_connection()
    with state_lock:
        shared_state["internet_connected"] = initial_check
    last_state = initial_check

    if initial_check:
        log.info("🌐 [Network Status]: Terkoneksi ke Internet
Global.")
    else:
        log.warning("🌐 [Network Status]: Offline / Tidak ada
akses internet.")

    while not shutdown_event.is_set():
        try:
            current_state =
check_internet_connection(host="1.1.1.1", port=53,
timeout=2.0)

            if current_state != last_state:
                with state_lock:
                    shared_state["internet_connected"] =
current_state

                if current_state:
                    log.info("🌐 [Network Event]: RPi BERHASIL
TERKONEKSI KE INTERNET (Akses WAN Aktif).")
                else:
                    log.warning("🌐 [Network Event]: RPi
KEHILANGAN AKSES INTERNET (Koneksi Terputus).")

                last_state = current_state

            time.sleep(3.0)
        except Exception as e:
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
            log.error(f"Error pada Network Watchdog Worker:
{e}")
            time.sleep(5)

# ─── STATS OVERLAY HELPERS
──────────────────────────────────────────────

_fps_counter = {"count": 0, "last_time": time.time(), "fps":
0.0}
_fps_lock = threading.Lock()

def update_fps():
    with _fps_lock:
        _fps_counter["count"] += 1
        now = time.time()
        elapsed = now - _fps_counter["last_time"]
        if elapsed >= 1.0:
            _fps_counter["fps"] = round(_fps_counter["count"]
/ elapsed, 2)
            _fps_counter["count"] = 0
            _fps_counter["last_time"] = now

def get_cpu_temp():
    try:
        with open("/sys/class/thermal/thermal_zone0/temp",
"r") as f:
            return round(int(f.read().strip()) / 1000, 1)
    except Exception:
        return None

# ─── PERFORMANCE LOGGER
──────────────────────────────────────────────

def perf_logger_worker():
    log.info("📊 Performance Logger Worker started.")
    with open(PERF_LOG_FILE, mode='a', newline='') as f:
        writer = csv.writer(f)
        while not shutdown_event.is_set():
            try:
                with _fps_lock:
                    fps = _fps_counter["fps"]
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
                timestamp =
datetime.now(timezone.utc).strftime('%Y-%m-%dT%H:%M:%S.%f')[:-
3] + 'Z'
                cpu = psutil.cpu_percent(interval=None)
                ram = psutil.virtual_memory().percent
                suhu = get_cpu_temp()

                with state_lock:
                    shared_state["current_cpu_util"] = cpu
                    shared_state["current_ram_util"] = ram
                    if suhu is not None:
                        shared_state["current_cpu_temp"] =
suhu
                    inf_time =
shared_state.get("last_inf_time", 0.0)

                writer.writerow([timestamp, fps, cpu, ram,
suhu if suhu is not None else "", f"{inf_time:.4f}"])
                f.flush()
                time.sleep(1.0)
            except Exception as e:
                log.error(f"Error pada Performance Logger:
{e}")
                time.sleep(1)

# ——— I/O & CLOUD WORKER (CONSUMER THREAD)
———————————————————————

def io_worker(client):
    log.info("📦 Background I/O Worker started.")

    with open(LOG_FILE, mode='a', newline='') as file:
        writer = csv.writer(file)

        while True:
            try:
                data_paket = io_queue.get()

                if data_paket is None:
                    io_queue.task_done()
                    break
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
                writer.writerow(data_paket['csv_row'])
                file.flush()

                if data_paket['image_to_save'] is not None:
                    cv2.imwrite(data_paket['filepath'],
data_paket['image_to_save'], [cv2.IMWRITE_JPEG_QUALITY, 85])

                if data_paket.get('raw_image_to_save') is not
None:
                    cv2.imwrite(data_paket['raw_filepath'],
data_paket['raw_image_to_save'], [cv2.IMWRITE_JPEG_QUALITY,
85])

                with state_lock:
                    is_network_up =
shared_state["internet_connected"]

                if data_paket['send_to_cloud'] and client and
is_network_up:
                    try:
                        img_to_encode =
data_paket['image_to_save'] if data_paket['image_to_save'] is
not None else data_paket['fallback_raw_image']
                        ret, jpeg_buf = cv2.imencode('.jpg',
img_to_encode, [cv2.IMWRITE_JPEG_QUALITY, 85])

                        if ret:
                            b64_str =
base64.b64encode(jpeg_buf.tobytes()).decode("utf-8")
                            payload_data = {
                                "scan_session_id":
data_paket['session_id'],
                                "timestamp":
data_paket['timestamp'],
                                "image_base64":
f"{b64_str}",
                                "detections":
data_paket['detections'],
                                "min_confidence":
args.min_confidence
                            }
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
client.send_detection(data_paket['session_id'], payload_data)
                            log.info(f"📤 Cloud: Data dikirim
[{data_paket['filename']}]")
                    except Exception as e:
                        log.error(f"⚠️ Gagal mengirim data ke
cloud: {e}")

                io_queue.task_done()

            except Exception as e:
                log.error(f"Error pada I/O Worker: {e}")

def esp_watchdog(client):
    log.info(f"⏳ ESP Watchdog aktif — menunggu handshake maks
{ESP_CONNECT_TIMEOUT} detik...")
    deadline = time.time() + ESP_CONNECT_TIMEOUT

    while time.time() < deadline:
        with state_lock:
            if shared_state["esp_handshake_done"]:
                log.info("✅ ESP terdeteksi — mode:
ESP-Controlled.")
                return
        time.sleep(0.5)

    new_session = None
    if client:
        try:
            new_session = client.start_session()
        except Exception as e:
            log.error(f"⚠️ Gagal inisialisasi sesi cloud saat
boot (Server offline/RTO): {e}")

    with state_lock:
        if not shared_state["esp_handshake_done"]:
            shared_state["is_scanning"] = True
            if new_session:
                shared_state["session_id"] = new_session
                shared_state["is_cloud_active"] = True
```

```
                log.info(f"☁️  [STANDALONE CLOUD] Sesi server
aktif: #{new_session}")
            else:
                shared_state["session_id"] =
f"STANDALONE_{datetime.now().strftime('%Y%m%d_%H%M%S')}"
                shared_state["is_cloud_active"] = False
                log.warning("☁️  [STANDALONE LOCAL] Cloud
tidak tersedia, simpan lokal.")
    log.warning("⚠️  ESP tidak terdeteksi → STANDALONE MODE
aktif, scanning berjalan otomatis.")

# ─── PIPELINE ENGINE: INFERENCE & DATA PACKAGING WORKER
──────────────

def ai_inference_worker():
    global global_frame_annotated
    log.info("🚀 Background AI Inference Worker started.")
    last_save_time = 0
    last_sent_time = 0
    SAVE_INTERVAL = 5.0
    SEND_INTERVAL = 3.0

    while not shutdown_event.is_set():
        try:
            with state_lock:
                current_scanning = shared_state["is_scanning"]
                current_session_id =
shared_state["session_id"]
                current_cloud_active =
shared_state["is_cloud_active"]

            if not current_scanning:
                time.sleep(0.1)
                continue

            with frame_lock:
                if global_frame_raw is None:
                    time.sleep(0.01)
                    continue
                local_raw = global_frame_raw.copy()
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
            annotated_img, detections, counts, inf_time,
cropped_raw = engine.detect_and_process(local_raw)
            update_fps()

            with frame_lock:
                global_frame_annotated = annotated_img.copy()

            count_winged_aphids = counts.get("winged_aphid",
0)
            count_whitefly = counts.get("whitefly", 0)

            with state_lock:
                shared_state["last_inf_time"] = inf_time

            timestamp_sekarang =
datetime.now(timezone.utc).strftime('%Y-%m-%dT%H:%M:%S.%f')[:-
3] + 'Z'
            current_time = time.time()
            hama_detected = (count_winged_aphids > 0 or
count_whitefly > 0)

            should_send = (current_time - last_sent_time >=
SEND_INTERVAL)

            filename = "-"
            if hama_detected or (current_time - last_save_time
>= SAVE_INTERVAL):
                waktu_detail =
datetime.now().strftime("%Y%m%d-%H%M%S_%f")[:-3]
                if hama_detected:
                    filename =
f"hama_{waktu_detail}_A{count_winged_aphids}_K{count_whitefly}
.jpg"
                else:
                    filename = f"negatif_{waktu_detail}.jpg"

            if current_scanning:
                img_save_payload = annotated_img.copy() if
filename != "-" else None
                raw_save_payload = cropped_raw.copy() if
filename != "-" else None
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
                fallback_payload = annotated_img.copy() if
(should_send and current_cloud_active and filename == "-")
else None

                paket = {
                    'csv_row': [
                        timestamp_sekarang,
                        current_session_id,
                        filename,
                        count_winged_aphids,
                        count_whitefly,
                        "",
                        "",
                        f"{inf_time:.4f}",
                        ""
                    ],
                    'image_to_save': img_save_payload,
                    'raw_image_to_save': raw_save_payload,
                    'fallback_raw_image': fallback_payload,
                    'filepath': os.path.join(IMAGE_SAVE_DIR,
filename) if filename != "-" else "",
                    'raw_filepath':
os.path.join(RAW_DATASET_DIR, filename.replace('.jpg',
'_raw.jpg')) if filename != "-" else "",
                    'send_to_cloud': should_send and
current_cloud_active,
                    'session_id': current_session_id,
                    'timestamp': timestamp_sekarang,
                    'detections': detections,
                    'filename': filename
                }

                try:
                    io_queue.put(paket, timeout=0.1)
                except queue.Full:
                    log.warning("⚠️ Antrean I/O Penuh!
Melewati frame ini biar AI gak ngelag.")

                if filename != "-":
                    last_save_time = current_time
                if should_send:
```

```
                    last_sent_time = current_time

            time.sleep(0.01)

        except Exception as e:
            log.error(f"Error pada thread AI Inference Worker:
{e}")
            time.sleep(1)

# ——— FLASK STREAMING GENERATOR
———————————————————————————————

def gen_frames():
    log.info("📺 Client baru terhubung ke Flask Live Stream.")
    while True:
        with frame_lock:
            if global_frame_annotated is None:
                time.sleep(0.04)
                continue
            frame_bgr = global_frame_annotated.copy()

        with state_lock:
            scanning = shared_state.get("is_scanning", False)
            cloud = shared_state.get("is_cloud_active", False)
            cpu = shared_state.get("current_cpu_util", 0.0)
            ram = shared_state.get("current_ram_util", 0.0)
            suhu = shared_state.get("current_cpu_temp", 25.0)

        with _fps_lock:
            fps = _fps_counter["fps"]

        overlay_lines = [
            f"FPS: {fps:.1f}",
            f"CPU: {cpu:.1f}%  RAM: {ram:.1f}%",
            f"Suhu: {suhu:.1f}C",
            f"Scan: {'ON' if scanning else 'OFF'}  Cloud:
{'ON' if cloud else 'OFF'}",
        ]

        y = 20
        for line in overlay_lines:
            cv2.putText(frame_bgr, line, (10, y),
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
                        cv2.FONT_HERSHEY_SIMPLEX, 0.55, (0, 0,
0), 3)
            cv2.putText(frame_bgr, line, (10, y),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.55, (0,
255, 0), 1)
            y += 22

        ret, buffer = cv2.imencode('.jpg', frame_bgr,
[cv2.IMWRITE_JPEG_QUALITY, 70])
        if not ret:
            time.sleep(0.04)
            continue

        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' +
buffer.tobytes() + b'\r\n')

        time.sleep(0.04)

shutdown_lock = threading.Lock()

def graceful_shutdown(signum=None, frame=None):
    if not shutdown_lock.acquire(blocking=False):
        return

    if shutdown_event.is_set():
        return

    log.warning(f"🛑 Sinyal shutdown ({signum}) diterima —
memicu event stop ke semua thread...")

    shutdown_event.set()

    try:
        io_queue.put(None, timeout=1.0)
    except queue.Full:
        log.warning("⚠️ Antrean penuh, poison pill terpaksa
dilewati.")

    log.info("⏳ Menunggu io_worker mengosongkan sisa buffer
ke SD Card...")
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
    timeout_deadline = time.time() + 5.0
    while not io_queue.empty() and time.time() <
timeout_deadline:
        time.sleep(0.1)

    if not io_queue.empty():
        log.warning("⚠️ Batas waktu habis! Sisa antrean
terpaksa di-drop demi stabilitas sistem.")

    try:
        picam2.stop()
        log.info("📷 Perangkat keras kamera berhasil
dilepaskan.")
    except Exception as e:
        log.error(f"Gagal menghentikan kamera: {e}")

    if cloud_client and hasattr(cloud_client, "close"):
        try:
            cloud_client.close()
            log.info("🔌 Soket komunikasi cloud berhasil
ditutup.")
        except Exception as e:
            log.error(f"Gagal menutup client cloud secara
bersih: {e}")

    log.info("✅ Semua resource selesai dibersihkan. Keluar.")
    os._exit(0)

@app.route('/')
def index():
    return Response(gen_frames(),
mimetype='multipart/x-mixed-replace; boundary=frame')

# ——— MAIN ENTRY POINT
_____________________________________________

if __name__ == '__main__':

log.info("==================================================
===")
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
    log.info("  Lokatani Edge Runtime - Raspberry Pi
Initialization  ")

log.info("=============================================================
===")

    cloud_client = create_protocol_client(
        protocol_type=args.protocol,
        host=args.host,
        http_port=args.http_port,
        ws_port=args.ws_port,
        broker=args.broker,
        api_key=args.api_key
    )

    signal.signal(signal.SIGTERM, graceful_shutdown)
    signal.signal(signal.SIGINT, graceful_shutdown)

    threads = [
        threading.Thread(target=camera_stream_worker,
daemon=True),
        threading.Thread(target=ai_inference_worker,
daemon=True),
        threading.Thread(target=perf_logger_worker,
daemon=True),
        threading.Thread(target=io_worker,
args=(cloud_client,), daemon=True),
        threading.Thread(target=esp8266_usb_worker,
args=(args, state_lock, shared_state, cloud_client, log),
daemon=True),
        threading.Thread(target=esp_watchdog,
args=(cloud_client,), daemon=True),
        threading.Thread(target=network_watchdog_worker,
daemon=True)
    ]

    for t in threads:
        t.start()

    if args.run_duration > 0:
        def auto_shutdown():
```

```
            time.sleep(args.run_duration)
            log.info(f"⏰ Durasi {args.run_duration} detik
selesai — sistem mati otomatis.")
            os.kill(os.getpid(), signal.SIGTERM)
        threading.Thread(target=auto_shutdown,
daemon=True).start()
        log.info(f"⏱️  Auto-shutdown aktif dalam
{args.run_duration} detik.")

    app.run(host='0.0.0.0', port=5000, threaded=True,
debug=False)
```

Lampiran 9. Source Code Skrip Raspberry Pi Lanjutan

```
import cv2
import time
import logging
import os
os.environ["OMP_NUM_THREADS"] = "3"
from ultralytics import YOLO

log = logging.getLogger("LokataniEdge.Inference")

class InferenceEngine:
    def __init__(self, model_path, min_conf=0.25, imgsz=320):
        log.info(f"🔄 Menginisialisasi YOLO Engine menggunakan
model: {model_path}")
        self.model = YOLO(model_path)
        self.min_conf = min_conf
        self.imgsz = imgsz

        if not self.model.names:
            log.warning("⚠️ Metadata kelas tidak ditemukan
pada file Mengaktifkan fallback mapping.")
            self.class_mapping = {
                0: "whitefly",
                1: "winged_aphid"
            }
        else:
            self.class_mapping = self.model.names
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
        log.info(f"✅ Pemetaan Kelas Terkonfigurasi:
{self.class_mapping}")

    def detect_and_process(self, frame_rgb):
        img_bgr = cv2.cvtColor(frame_rgb, cv2.COLOR_RGB2BGR)

        h, w = img_bgr.shape[:2]
        cy1, cy2 = h//4, 3*h//4
        cx1, cx2 = w//4, 3*w//4
        cropped_frame = img_bgr[cy1:cy2, cx1:cx2]

        t_input = time.time()
        results = self.model(
            cropped_frame,
            imgsz=self.imgsz,
            conf=self.min_conf,
            verbose=False,
            workers=0
        )
        inference_time = time.time() - t_input

        detected_boxes = results[0].boxes
        payload = []
        counts = {"whitefly": 0, "winged_aphid": 0}

        for box in detected_boxes:
            conf = float(box.conf[0])
            cls_id = int(box.cls[0])

            cls_name = self.class_mapping.get(cls_id,
f"class_{cls_id}")

            if cls_name in counts:
                counts[cls_name] += 1

            bx1, by1, bx2, by2 = box.xyxy[0].tolist()

            payload.append({
                "class_name": cls_name,
                "confidence": round(conf, 4),
                "bbox": [
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
                    round(bx1, 1),
                    round(by1, 1),
                    round(bx2 - bx1, 1),
                    round(by2 - by1, 1)
                ],
            })

        annotated_img = results[0].plot()

        return annotated_img, payload, counts, inference_time,
cropped_frame
```

Lampiran 10. Source Code Skrip Raspberry Pi Lanjutan

```
import time
import serial
import logging
from datetime import datetime

def esp8266_usb_worker(args, state_lock, shared_state, client,
log=None):
    if log is None:
        log = logging.getLogger("LokataniEdge.Serial")

    ser = None
    last_ping_time = 0
    last_direction = b'1'

    while True:
        try:
            if ser is None or not ser.is_open:
                try:
                    ser = serial.Serial()
                    ser.port = args.serial_port
                    ser.baudrate = 115200
                    ser.timeout = 0.1

                    ser.dtr = False
                    ser.rts = False
                    ser.open()
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
                    log.info("⏳ Port terbuka. Memberikan jeda
2 detik untuk inisialisasi hardware ESP...")
                    time.sleep(2)

                    ser.reset_input_buffer()
                    ser.reset_output_buffer()

                    with state_lock:
                        shared_state["esp_hardware_ready"] =
True
                    log.info(f"🔌 Koneksi Fisik Terbuka di
port {args.serial_port}. Menunggu sinkronisasi...")
                except Exception as e:
                    ser = None
                    with state_lock:
                        shared_state["esp_hardware_ready"] =
False
                        shared_state["esp_handshake_done"] =
False
                    time.sleep(5)
                    continue

            now = time.time()

            if now - last_ping_time > 1.5:
                if ser and ser.is_open:
                    try:
                        with state_lock:
                            is_handshaked =
shared_state["esp_handshake_done"]

                        if not is_handshaked:
                            log.info("📤 Mengirim interupsi
jabat tangan 'P\\n' ke ESP8266...")

                        ser.write(b'P\n')
                        ser.flush()
                        last_ping_time = now
                    except (serial.SerialException, OSError)
as e:
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
                        log.error(f"Koneksi terputus saat
ping: {e}")
                        ser.close()
                        ser = None
                else:
                    ser = None

            if ser.in_waiting > 0:
                try:
                    raw_data = ser.read_until(b'\n')
                    if not raw_data:
                        continue
                    trigger = raw_data.decode("utf-8",
errors='ignore').strip().upper()

                    if trigger.startswith("ERROR_"):
                        log.warning(f"⚠️ [ESP ERROR]
{trigger}")
                        with state_lock:
                            if
shared_state["esp_handshake_done"]:
                                shared_state["is_scanning"] =
False

shared_state["is_cloud_active"] = False

                        last_direction = b'2' if
last_direction == b'1' else b'1'
                        arah_label = "MAJU" if last_direction
== b'1' else "MUNDUR"
                        log.warning(f"🔄 Recovery: retry ke
arah {arah_label} dalam 3 detik...")

                        time.sleep(3)

                        try:
                            ser.write(last_direction)
                            ser.flush()
                            log.info(f"📤 Recovery dikirim:
'{last_direction.decode()}'")
                        except Exception as e:
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
                        log.error(f"Gagal recovery: {e}")

                    continue

                if trigger not in ["SYSTEM_READY",
"START", "END", "PAUSED", "RESUMED"]:
                    log.debug(f"ESP Debug: {trigger}")
                    continue
            except Exception as e:
                log.error(f"Error membaca serial: {e}")
                ser.close()
                ser = None
                continue

            if trigger == "SYSTEM_READY":
                with state_lock:
                    is_already_ready =
shared_state["esp_handshake_done"]

                if not is_already_ready:
                    log.info("📥 [HANDSHAKE SUCCESS]
ESP8266 Merespon Sinyal Kesiapan Awal.")
                    ser.write(b'1')
                    last_direction = b'1'
                    log.info("📤 Instruksi '1' (MAJU)
dikirim ke Driver Motor.")
                    with state_lock:
                        shared_state["esp_handshake_done"]
= True
                        shared_state["is_scanning"] = True
                else:
                    pass

            elif trigger == "START":
                last_direction = b'2' if last_direction ==
b'1' else b'1'
                log.info("🟢 Perintah START Diterima dari
Gantry. Membuka Sesi Baru.")
                new_session = None
                if client:
                    new_session = client.start_session()
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
                    with state_lock:
                        shared_state["is_scanning"] = True
                        if new_session:
                            shared_state["session_id"] =
new_session
                            shared_state["is_cloud_active"] =
True
                            log.info(f"   [CLOUD CONNECTED] ID
Sesi Server: #{new_session}")
                        else:
                            local_id =
f"LOCAL_{datetime.now().strftime('%Y%m%d_%H%M%S')}"
                            shared_state["session_id"] =
local_id
                            shared_state["is_cloud_active"] =
False
                            log.warning(f"   [OFFLINE MODE]
Sesi Dialihkan ke Penyimpanan Lokal: {local_id}")

                elif trigger == "END":
                    log.info(f"🔴 Perintah END Diterima.
Menghentikan Pengambilan Gambar Sementara.")
                    with state_lock:
                        active_session =
shared_state["session_id"]
                        was_cloud_active =
shared_state["is_cloud_active"]
                        shared_state["is_scanning"] = False
                        shared_state["is_cloud_active"] =
False
                        shared_state["session_id"] =
f"LOCAL_IDLE_{datetime.now().strftime('%Y%m%d_%H%M%S')}"

                    if was_cloud_active and client:
                        try:
                            client.end_session(active_session)
                        except Exception:
                            pass

                elif trigger in ["PAUSED", "RESUMED"]:
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
                    with state_lock:
                        shared_state["is_scanning"] = (trigger
== "RESUMED")
                    log.info(f"ℹ️ Status Mekanik Rel Berubah:
{trigger}")

            time.sleep(0.02)

        except Exception as e:
            log.error(f"Kesalahan Fatal pada Worker: {e}")
            with state_lock:
                shared_state["esp_hardware_ready"] = False
                shared_state["esp_handshake_done"] = False

            if ser:
                ser.close()
                ser = None
            time.sleep(2)
```

Lampiran 11. Source Code Skrip Raspberry Pi Lanjutan

```
import json
import logging
import threading
import time
from clients.base import BaseCloudClient

log = logging.getLogger("LokataniEdge.MQTT")

class MQTTClient(BaseCloudClient):
    def __init__(self, host, broker_url_arg, api_key):
        import paho.mqtt.client as paho_mqtt
        self._paho = paho_mqtt
        self.api_key = api_key

        broker_url = broker_url_arg or f"mqtt://{host}:1883"
        clean_url = broker_url.replace("mqtt://", "")
        broker_host = clean_url.split(":")[0]
        broker_port = int(clean_url.split(":")[-1]) if ":" in
clean_url else 1883

        self._client = paho_mqtt.Client(
```

```
            client_id=f"lokatani-rpiw-{int(time.time())}",
            protocol=paho_mqtt.MQTTv311,
        )
        self._responses = {}
        self._events = {}
        self._lock = threading.Lock()

        self._client.on_connect = self._on_connect
        self._client.on_disconnect = self._on_disconnect
        self._client.on_message = self._on_message

        self._client.connect(broker_host, broker_port, 60)
        self._client.loop_start()

    def _on_connect(self, client, userdata, flags, rc):
        if rc == 0:
            log.info(f"🔌 [MQTT Connected] Berhasil
terhubung/reconnect ke broker.")
            self._client.subscribe([
                ("lokatani/detect/down", 0),
                ("lokatani/session/ack", 1),
            ])
        else:
            log.error(f"❌ [MQTT Connect Error] Gagal
terhubung, kode hasil: {rc}")

    def _on_disconnect(self, client, userdata, rc):
        if rc != 0:
            log.warning(f"⚠️ [MQTT Disconnected] Koneksi
terputus tak wajar. Auto-reconnect aktif...")
        else:
            log.info(f"🔌 [MQTT Disconnected] Klien ditutup
secara normal.")

    def _on_message(self, client, userdata, msg):
        try:
            payload = json.loads(msg.payload.decode())
            action = payload.get("action", msg.topic)
            with self._lock:
                self._responses[action] = payload
                if action in self._events:
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
                    self._events[action].set()
        except Exception as e:
            log.error(f"⚠️ [MQTT Parse Error] Gagal memetakan
payload masuk: {e}")

    def _wait_for(self, action, timeout=10):
        event = threading.Event()
        with self._lock:
            self._events[action] = event
        if not event.wait(timeout):
            raise TimeoutError(f"⏳ [MQTT Timeout] Menunggu
ACK untuk aksi: {action}")
        with self._lock:
            return self._responses.pop(action, None)

    def start_session(self) -> str:
        try:
            if not self._client.is_connected():
                raise ConnectionError("Broker sedang luring
(Offline).")
            payload = {"api_key": self.api_key, "protokol":
"MQTT"}
            self._client.publish("lokatani/session/start",
json.dumps(payload), qos=1)
            resp = self._wait_for("session_started")
            return resp["data"]["sessionId"]
        except Exception as e:
            log.error(f"❌ [MQTT Session] Gagal
mempublikasikan start session: {e}")
            return None

    def send_detection(self, session_id: str, payload_data:
dict) -> dict:
        try:
            if not self._client.is_connected():
                raise ConnectionError("MQTT Broker Offline
(Menunggu pemulihan otomatis...)")

            payload = {
                "api_key": self.api_key,
                "timestamp": payload_data.get("timestamp"),
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
                "image_base64": payload_data["image_base64"],
                "detections": payload_data["detections"],
                "scan_session_id": session_id,
                "min_confidence":
payload_data.get("min_confidence", 0.25),
            }

            msg_info =
self._client.publish("lokatani/detect/up",
json.dumps(payload), qos=0)
            if msg_info.rc != self._paho.MQTT_ERR_SUCCESS:
                raise RuntimeError(f"Paho MQTT Internal Error
Code: {msg_info.rc}")

            return {"latency_ms": "N/A (QoS 0)"}
        except Exception as e:
            raise Exception(f"❌ [MQTT Detect] {e}")

    def end_session(self, session_id: str) -> dict:
        try:
            if not self._client.is_connected():
                raise ConnectionError("Broker sedang luring
(Offline).")
            payload = {"api_key": self.api_key, "session_id":
session_id}
            self._client.publish("lokatani/session/end",
json.dumps(payload), qos=1)
            resp = self._wait_for("session_ended")
            return resp.get("data", {})
        except Exception as e:
            log.error(f"❌ [MQTT Session] Gagal
mempublikasikan end session: {e}")
            return None

    def close(self):
        self._client.loop_stop()
        self._client.disconnect()
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Lampiran 12. Source Code Skrip ESP8266

```
const int pinStep        = 12;
const int pinDir         = 13;

const int pinLimitMaju   = 5;
const int pinLimitMundur = 4;

const int JEDA_SPEED_MOTOR = 6000;
const unsigned long TIMEOUT_S_REL = 90000;
const unsigned long TIMEOUT_RASPI = 5000;
unsigned long waktuKomunikasiTerakhir = 0;

bool sistemAktif = false;
bool statusPause = false;
char arahSiklus = '1';

unsigned long waktuMulaiSiklus = 0;
unsigned long waktuTerpause = 0;

void setup() {
  Serial.begin(115200);
  delay(500);
  Serial.flush();
  while (!Serial) { ; }

  pinMode(pinStep, OUTPUT);
  pinMode(pinDir, OUTPUT);

  pinMode(pinLimitMaju, INPUT_PULLUP);
  pinMode(pinLimitMundur, INPUT_PULLUP);
  waktuKomunikasiTerakhir = millis();

  Serial.println("\nSYSTEM_READY");
  Serial.flush();
}

void loop() {
  if (Serial.available() > 0) {
    waktuKomunikasiTerakhir = millis();
    char inputUser = Serial.read();
    if (inputUser == '\n' || inputUser == '\r') return;
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
    if (inputUser == 'P') {
      Serial.println("SYSTEM_READY");
      Serial.flush();
    }
    else if ((inputUser == '1' || inputUser == '2') &&
!sistemAktif && !statusPause) {
      arahSiklus = inputUser;
      waktuKomunikasiTerakhir = millis();

      Serial.println("START");
      Serial.flush();

      bool cekMaju  = (arahSiklus == '1' &&
digitalRead(pinLimitMaju) == LOW);
      bool cekMundur = (arahSiklus == '2' &&
digitalRead(pinLimitMundur) == LOW);

      if (cekMaju || cekMundur) {
        Serial.println("END");
        Serial.flush();

        delay(2000);

        arahSiklus = (arahSiklus == '1') ? '2' : '1';
        waktuKomunikasiTerakhir = millis();

        Serial.println("START");
        Serial.flush();
      }

      sistemAktif = true;
      waktuMulaiSiklus = millis();
    }
    else if (inputUser == '3' && sistemAktif && !statusPause)
{
      sistemAktif = false;
      statusPause = true;
      waktuTerpause = millis() - waktuMulaiSiklus;
      Serial.println("PAUSED");
      Serial.flush();
```

```
    }
    else if (inputUser == '4' && !sistemAktif && statusPause)
{
      statusPause = false;
      sistemAktif = true;
      waktuMulaiSiklus = millis() - waktuTerpause;
      waktuKomunikasiTerakhir = millis();

      Serial.println("RESUMED");
      Serial.flush();
    }

    while(Serial.available() > 0) { Serial.read(); }
  }

  if (sistemAktif) {

    if (arahSiklus == '1') {
      digitalWrite(pinDir, HIGH);
    } else if (arahSiklus == '2') {
      digitalWrite(pinDir, LOW);
    }

    while (true) {
      if (millis() - waktuMulaiSiklus > TIMEOUT_S_REL) {
        sistemAktif = false;
        statusPause = false;
        Serial.println("ERROR_LIMIT_SWITCH_TIMEOUT");
        Serial.flush();
        break;
      }

      if (millis() - waktuKomunikasiTerakhir > TIMEOUT_RASPI)
{
        sistemAktif = false;
        statusPause = false;
        Serial.println("ERROR_RASPI_DISCONNECTED_WATCHDOG");
        Serial.flush();
        break;
      }
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
      bool tabrakanMaju   = (arahSiklus == '1' &&
digitalRead(pinLimitMaju) == LOW);
      bool tabrakanMundur = (arahSiklus == '2' &&
digitalRead(pinLimitMundur) == LOW);

      if (tabrakanMaju || tabrakanMundur) {
        Serial.println("END");
        Serial.flush();

        for (int d = 0; d < 70; d++) {
          delay(100);
          if (Serial.available() > 0 && Serial.peek() == 'P')
{
            Serial.read();
            waktuKomunikasiTerakhir = millis();
            Serial.println("SYSTEM_READY");
            Serial.flush();
          }
        }

        waktuKomunikasiTerakhir = millis();

        arahSiklus = (arahSiklus == '1') ? '2' : '1';
        digitalWrite(pinDir, (arahSiklus == '1') ? HIGH :
LOW);
        delayMicroseconds(10);

        for (int i = 0; i < 200; i++) {
          digitalWrite(pinStep, HIGH);
          delayMicroseconds(JEDA_SPEED_MOTOR);
          digitalWrite(pinStep, LOW);
          delayMicroseconds(JEDA_SPEED_MOTOR);
          waktuKomunikasiTerakhir = millis();
          yield();
        }

        Serial.println("START");
        Serial.flush();

        waktuMulaiSiklus = millis();
        break;
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
    }
    if (Serial.available() > 0) {
      char peekChar = Serial.peek();
      if (peekChar == '3') {
        Serial.read();
        sistemAktif = false;
        statusPause = true;
        waktuTerpause = millis() - waktuMulaiSiklus;
        Serial.println("PAUSED");
        Serial.flush();
        break;
      }
      else if (peekChar == 'P') {
        Serial.read();
        waktuKomunikasiTerakhir = millis();
        Serial.println("SYSTEM_READY");
        Serial.flush();
      }
      else {
        Serial.read();
      }
    }

    digitalWrite(pinStep, HIGH);
    delayMicroseconds(JEDA_SPEED_MOTOR);
    digitalWrite(pinStep, LOW);
    delayMicroseconds(JEDA_SPEED_MOTOR);

    yield();
  }
 }
}
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Lampiran 13. Dokumentasi Hasil Inferensi Model YOLO

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Audit Trail

DigiSigner Document ID: 0c9c7c0f-d703-4caa-90cd-d56f0e8ac92f

Signer

Email: alfarizki.nurachman.tik22@mhsw.pnj.ac.id
IP Address: 118.99.112.63

Signature

Event	User	Time	IP Address
Upload document	alfarizkin9286@gmail.com	07/24/2026 8:16:44PM EDT	118.99.112.63
Send for signing	alfarizkin9286@gmail.com	07/24/2026 8:16:51PM EDT	118.99.112.63
Open document	alfarizki.nurachman.tik22@mhsw.pnj.ac.id	07/24/2026 8:17:03PM EDT	118.99.112.63
Sign document	alfarizki.nurachman.tik22@mhsw.pnj.ac.id	07/24/2026 8:17:20PM EDT	118.99.112.63
Close document	alfarizki.nurachman.tik22@mhsw.pnj.ac.id	07/24/2026 8:17:20PM EDT	118.99.112.63

Audit Trail

DigiSigner Document ID: 462d379f-f262-478e-acf3-a8cdf1211ca0

Signer

IP Address: 103.36.14.70

Signature

Event	User	Time	IP Address
Upload document	alfarizkin9286@gmail.com	07/21/2026 12:24:48AM EDT	118.99.112.63
Send for signing	alfarizkin9286@gmail.com	07/21/2026 12:25:16AM EDT	118.99.112.63
Open document	unknown	07/21/2026 2:18:19AM EDT	103.36.14.70
Sign document	unknown	07/21/2026 2:18:41AM EDT	103.36.14.70
Close document	unknown	07/21/2026 2:18:41AM EDT	103.36.14.70
Open document	alfarizkin9286@gmail.com	07/21/2026 8:07:24AM EDT	118.99.112.63
Close document	alfarizkin9286@gmail.com	07/21/2026 8:07:29AM EDT	118.99.112.63
Download document	alfarizkin9286@gmail.com	07/21/2026 11:57:21PM EDT	182.0.207.42