Jurnal Minfo Polgan (JMP)
Volume 15, Nomor 2, Mei-Agustus 2026
DOI : https://doi.org/10.33395/jmp.v15i2.16479
e-ISSN : 2797-3298
p-ISSN : 2089-9424

# Implementasi Koreksi Posisi Otomatis Berbasis YOLOv8 dan FINS pada Trainer PLC Plant XYZ

**[1] Yuliadi Erdani, [2]Hendy Rudiansyah, [3*]Aurilia Lysandra Rosidin, [4]Sarosa Kasrena Abadi, [5]Abdur Rohman Harits Martawireja, [6]Abyanuddin Salam, [7]Rizqi Aji Pratama**
[1*,2,3,4,5]Teknik Otomasi Manufaktur dan Mekatronika, Politeknik Manufaktur Bandung, Bandung, Indonesia
[6]School of Science Computing Emerging Technologies, Swinburne University of Technology, Hawthorn, Australia
*Korespondensi: aurilyandin@gmail.com

**Submit** : 09 Jun 2026 | **Diterima** : 02 Jul 2026 | **Terbit** : 28 Agust 2026

***ABSTRACT***

*The pneumatic manipulator on the Plant XYZ trainer operates sequentially without visual feedback, making it rigid and unable to accommodate variation in object position or type. This study designs and implements an intelligent control system that integrates Computer Vision and Machine Learning into the Plant XYZ PLC trainer. A Raspberry Pi 4 runs a YOLOv8 model to detect and classify circular and rectangular workpieces in real-time, while OpenCV handles image pre-processing. Classification results and work zones are transmitted to an Omron CP1H PLC through the FINS UDP/TCP protocol using a bit-based handshaking mechanism on the Work Area rather than numeric coordinate transfer, enabling the PLC to execute Ladder logic for a three-axis pneumatic manipulator. The system was developed using the Waterfall method and validated through both simulation and ten sessions of actual field testing. The YOLOv8 model achieved a perfect mAP50 of 100 percent with 99.1 percent precision and 100 percent recall on the validation set, with no misclassification across 283 actual detection cycles. FINS communication proved reliable with a 96.5 percent success rate over 1,662 commands, while the automatic position-correction mechanism, the primary objective of this study, was validated at a 94.6 percent success rate over 111 real correction operations. These results confirm that integrating computer vision and PLC control successfully transforms the Plant XYZ pick-and-place process from a static system into an adaptive one capable of detecting, classifying, and correcting workpiece placement autonomously.*

***Keywords****: automatic position correction; Computer Vision; FINS UDP/TCP; Raspberry Pi 4; YOLOv8.*

**ABSTRAK**

Manipulator pneumatik pada trainer Plant XYZ bekerja secara sekuensial tanpa umpan balik visual sehingga bersifat kaku dan gagal mengakomodasi variasi posisi maupun jenis objek. Penelitian ini merancang dan mengimplementasikan sistem kendali cerdas yang mengintegrasikan Computer Vision dan Machine Learning pada trainer PLC Plant XYZ. Raspberry Pi 4 menjalankan model YOLOv8 untuk mendeteksi dan mengklasifikasikan objek bulat dan kotak secara real-time, dibantu OpenCV untuk pra-pemrosesan citra. Hasil klasifikasi dan zona kerja ditransmisikan ke PLC Omron CP1H melalui protokol FINS UDP/TCP menggunakan mekanisme handshaking berbasis bit pada Work Area sebagai pengganti transfer koordinat numerik, sehingga PLC dapat mengeksekusi logika Ladder untuk menggerakkan manipulator pneumatik tiga sumbu. Sistem dikembangkan melalui metode Waterfall dan divalidasi melalui pengujian simulasi serta sepuluh sesi pengujian aktual lapangan. Model YOLOv8 mencapai mAP50 sempurna 100 persen dengan precision 99,1 persen dan recall 100 persen pada

@2026 The Author Published By Politeknik Ganesha Medan
This is an open access article under the CC BY licence (http://creativecommons.org/licenses/by/4.0/). which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properlycited.

validation set, serta tanpa misklasifikasi pada 283 siklus deteksi aktual. Komunikasi FINS terbukti andal dengan tingkat keberhasilan 96,5 persen dari 1.662 perintah, sementara mekanisme koreksi posisi otomatis sebagai tujuan utama penelitian berhasil divalidasi dengan tingkat keberhasilan 94,6 persen dari 111 operasi koreksi nyata. Hasil ini membuktikan bahwa integrasi visi komputer dan kendali PLC berhasil mentransformasi proses pick-and-place Plant XYZ dari sistem statis menjadi sistem adaptif yang mampu mendeteksi, mengklasifikasikan, dan mengoreksi penempatan benda kerja secara mandiri.

**Kata Kunci**: Computer Vision; FINS UDP/TCP; koreksi posisi otomatis; Raspberry Pi 4; YOLOv8.

## PENDAHULUAN

Revolusi Industri 4.0 menandai transisi paradigmatik dalam sektor manufaktur, yaitu pergeseran dari otomatisasi kaku berbasis urutan tetap menuju sistem siber-fisik yang adaptif dan mampu mengambil keputusan secara mandiri. Dalam kerangka ini, integrasi kecerdasan buatan ke dalam ekosistem Programmable Logic Controller (PLC) telah menjadi pendekatan strategis untuk memangkas latensi pengambilan keputusan sekaligus meningkatkan efisiensi dan fleksibilitas produksi (Dhameliya, 2023). Salah satu aktivitas manufaktur yang paling banyak menuntut fleksibilitas adalah proses *pick-and-place*. Meskipun sistem *pick-and-place* konvensional berbasis aktuator pneumatik menawarkan reliabilitas tinggi serta determinisme yang baik untuk tugas repetitif, sistem ini secara inheren bersifat buta identitas, yakni tidak memiliki kemampuan sensorik untuk mengenali karakteristik fisik objek dan hanya mengandalkan umpan balik mekanis dari sensor proximity. Akibatnya, sistem mengeksekusi urutan gerak yang identik untuk setiap benda tanpa mempertimbangkan variasi jenis maupun posisi, sehingga kurang efektif ketika dihadapkan pada variabilitas produk (Maslanka et al., 2025).

Trainer PLC Plant XYZ yang menjadi objek penelitian ini merepresentasikan keterbatasan tersebut secara konkret. Manipulator pneumatik tiga sumbu pada platform ini dikendalikan oleh logika sekuensial murni (*step sequence control)* sehingga tidak mampu beradaptasi ketika sebuah benda kerja ditempatkan pada zona yang keliru. Untuk mengatasi rigiditas ini, penelitian mengusulkan integrasi Computer Vision berbasis Machine Learning dengan kontroler PLC, di mana kamera difungsikan sebagai sensor cerdas yang tidak hanya mendeteksi keberadaan objek tetapi juga mengklasifikasikan jenisnya serta menentukan zona kerjanya secara real-time. Algoritma *You Only Look Once* (YOLO) dipilih sebagai inti deteksi karena pendekatan satu tahap yang memprediksi bounding box dan probabilitas kelas secara simultan, sehingga menghasilkan keseimbangan optimal antara kecepatan inferensi dan akurasi yang krusial bagi lingkungan industri (Casas et al., 2024).

Sejumlah penelitian terdahulu telah membuktikan kelayakan pendekatan ini, namun masih meninggalkan kesenjangan yang belum terjawab. Almtireen et al. (2025) membuktikan bahwa arsitektur visi berbasis kecerdasan buatan yang terhubung ke PLC mampu memilah limbah secara otomatis dengan presisi tinggi, sementara Greco et al. (2024) memvalidasi kapabilitas Raspberry Pi 4 sebagai perangkat edge computing untuk menjalankan algoritma visi pada tugas depalletizing dengan biaya jauh lebih rendah dibanding PC industri. Pada sisi komunikasi, Kusumengema et al. (2022) menunjukkan bahwa protokol Factory Interface Network Service (FINS) mampu mentransfer data deteksi dari Raspberry Pi ke PLC dengan latensi sangat rendah. Namun, mayoritas studi tersebut berhenti pada tahap klasifikasi dan penempatan satu arah, serta umumnya mengandalkan transfer koordinat numerik yang menuntut kalibrasi presisi dan rentan terhadap kesalahan skala. Belum banyak studi yang secara eksplisit menutup loop kendali dengan mekanisme koreksi posisi otomatis, yaitu

kemampuan sistem untuk mendeteksi kesalahan penempatan dan secara mandiri memindahkan benda ke lokasi yang benar, serta memvalidasinya pada kondisi operasional fisik yang sesungguhnya.

Penelitian ini mengisi kesenjangan tersebut dengan mengembangkan arsitektur komunikasi yang andal namun tetap rendah biaya, yang menerapkan handshaking berbasis bit pada *Work Area* PLC sebagai pengganti transfer koordinat numerik. Perbedaan utama penelitian ini dibanding studi sebelumnya terletak pada penerapan mekanisme koreksi posisi tertutup yang divalidasi secara empiris pada kondisi lapangan nyata, bukan sekadar simulasi. Berdasarkan latar belakang tersebut, tujuan penelitian ini dirumuskan menjadi tiga, yaitu: (1) merancang arsitektur komunikasi data yang reliabel dan berlatensi rendah antara Raspberry Pi 4 dan PLC Omron CP1H; (2) mengoptimalkan performa algoritma YOLOv8 untuk mendeteksi serta mengklasifikasikan objek kotak dan bulat pada lingkungan Plant XYZ; dan (3) merancang serta memvalidasi logika Ladder Diagram adaptif yang mampu mengoreksi penempatan benda kerja secara mandiri. Kontribusi penelitian mencakup dua hal: arsitektur komunikasi FINS berbasis bit yang ringan dan tahan terhadap kesalahan skala, serta bukti empiris kuantitatif mengenai kinerja koreksi posisi otomatis pada sepuluh sesi pengujian lapangan.

## METODE PENELITIAN

### Studi Penelitian Terdahulu

Perancangan metode pada penelitian ini berlandaskan sejumlah studi terdahulu yang relevan. Casas et al. (2024) melakukan studi komparatif YOLOv5 dan YOLOv8 dan menyimpulkan bahwa YOLOv8 dengan struktur backbone C2f serta mekanisme anchor-free menawarkan presisi dan kecepatan inferensi lebih unggul dibanding pendahulunya, sehingga menjadi dasar pemilihan arsitektur deteksi pada penelitian ini. Pamungkas et al. (2025) membangun sistem sortir otomatis berbasis YOLOv5 dan PLC Outseal dengan akurasi rata-rata 88,26 persen, namun menyoroti sensitivitas sistem terhadap variabel pencahayaan, yang menjustifikasi transisi ke arsitektur yang lebih mutakhir seperti YOLOv8 sekaligus pengondisian pencahayaan terkontrol.

Pada aspek komunikasi, Kusumengema et al. (2022) membuktikan bahwa integrasi Raspberry Pi dengan PLC melalui protokol FINS mampu mentransfer data deteksi ke area memori dengan latensi sangat rendah, yang menjadi acuan pemilihan protokol komunikasi. Ridwan et al. (2025) menghadirkan arsitektur pemilahan SmartEdu Conveyor yang menyinergikan Raspberry Pi 4 dengan YOLOv8 dan mencapai mean Average Precision 0,993 dalam klasifikasi objek geometris, menjadi preseden integrasi visi ke perangkat SmartEdu. Untuk transformasi koordinat, Song et al. (2021) memformalkan konversi piksel ke koordinat dunia menggunakan model pinhole camera yang diadopsi pada penelitian ini. Berbeda dari studi-studi tersebut yang umumnya berhenti pada klasifikasi dan penempatan satu arah, penelitian ini melakukan modifikasi dengan menambahkan mekanisme koreksi posisi tertutup berbasis handshaking bit yang divalidasi pada kondisi lapangan nyata.

### Tahapan Penelitian

Perancangan sistem mengadopsi kerangka kerja Waterfall, yaitu pendekatan pengembangan sekuensial yang menuntut penyelesaian tuntas pada setiap fase sebelum beralih ke fase berikutnya. Model ini dipilih secara strategis untuk memitigasi risiko integrasi yang kompleks antara domain perangkat keras dan perangkat lunak, mengingat sistem yang dikembangkan menggabungkan visi komputer, komunikasi jaringan, dan kendali aktuator fisik. Penelitian terbagi ke dalam empat tahapan utama. Tahap analisis kebutuhan difokuskan pada diagnostik terhadap infrastruktur Plant XYZ yang ada, yang mengungkap defisiensi utama berupa manipulator pneumatik yang beroperasi secara buta karena hanya bergantung pada limit switch; pada fase ini

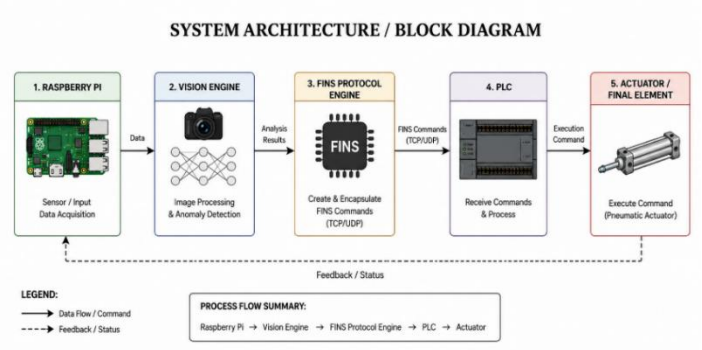
parameter krusial seperti resolusi kamera, beban komputasi Raspberry Pi 4, dan protokol komunikasi FINS didefinisikan. Tahap desain sistem menghasilkan cetak biru berupa diagram arsitektur komunikasi, perancangan peta memori, diagram alir penyelarasan inferensi dengan logika PLC, serta model matematis transformasi koordinat. Tahap implementasi merealisasikan rancangan menjadi sistem fungsional pada dua domain paralel, yaitu instalasi perangkat keras dan pengembangan pipeline kecerdasan buatan beserta logika kendali. Tahap pengujian melakukan validasi empiris secara kuantitatif, baik melalui simulasi maupun pengujian aktual lapangan. Seluruh data diperoleh secara otomatis melalui pencatatan log berlabel waktu yang dihasilkan sistem selama operasi, mencakup hasil inferensi per siklus, setiap transaksi FINS beserta waktu responsnya, dan setiap kejadian koreksi posisi.

**Perangkat Keras dan Lingkungan Pengujian**

Sistem dibangun di atas Raspberry Pi 4 sebagai unit pemrosesan utama yang dipilih karena keseimbangan antara performa komputasi dan efisiensi daya untuk menjalankan model deteksi ringan pada perangkat tertanam. Akuisisi citra dilakukan oleh webcam Logitech C270 beresolusi 720p yang dipasang pada posisi tegak lurus di ketinggian 40 cm dari permukaan tray, identik antara tahap pelatihan dan pengujian guna menjaga konsistensi distribusi data. Unit kendali menggunakan PLC Omron Sysmac CP1H yang beroperasi pada tegangan 24V DC. Karena PLC tipe ini tidak memiliki port Ethernet bawaan, ditambahkan modul ekspansi CP1W-CIF41 yang memungkinkan PLC memiliki alamat IP lokal dan berfungsi sebagai FINS Server. Aktuasi mekanik dilakukan oleh manipulator pneumatik tiga sumbu dengan silinder double-acting, solenoid valve 5/2 bertegangan 24V DC, dan vacuum ejector berbasis prinsip Venturi sebagai end-effector. Seluruh pengujian dilaksanakan pada kondisi pencahayaan buatan yang dipasang pada struktur trainer untuk menjaga konsistensi visual sesuai batasan sistem yang dirancang.

**Arsitektur Sistem dan Komunikasi FINS**

Sistem tersusun atas lima segmen yang saling terinterkoneksi secara serial: kamera Logitech C270 sebagai sensor visual, Raspberry Pi 4 sebagai unit pemrosesan cerdas yang menjalankan pra-pemrosesan OpenCV dan inferensi YOLOv8, jalur komunikasi FINS UDP/TCP melalui Ethernet, PLC Omron CP1H sebagai unit kendali, serta manipulator pneumatik tiga sumbu sebagai aktuator akhir. Dalam topologi ini Raspberry Pi berperan sebagai FINS Client dan PLC sebagai FINS Server melalui modul CP1W-CIF41. Arsitektur sistem secara keseluruhan ditunjukkan pada Gambar 1.




**Gambar 1**. Arsitektur sistem dan alur komunikasi FINS

Mekanisme pertukaran data dialokasikan pada area memori Work Area (W) dengan kode area FINS 0xB1 untuk operasi bit. Berbeda dengan pendekatan transfer koordinat numerik yang lazim, sistem ini menerapkan handshaking berbasis bit: Raspberry Pi menulis nilai ON pada bit jenis objek Trig_Bulat/Trig_Kotak, W0.06–W0.07 dan bit zona

tujuan W0.00–W0.05 hanya setelah hasil klasifikasi YOLOv8 dipastikan valid. PLC diprogram untuk mengeksekusi pergerakan silinder hanya apabila bit Safe_to_move W10.00 bernilai ON, yang menandakan kondisi mesin siap dan sensor Z0 berada di posisi atas. Setelah satu *fase sequence Pick atau Place* selesai dieksekusi oleh ladder, PLC men-SET bit Seq_Done W10.01 sebagai sinyal kepada Raspberry Pi untuk melanjutkan ke fase berikutnya, lalu skrip Python meresetnya kembali. Apabila ladder mendeteksi kegagalan, misalnya batas waktu sensor terlampaui, bit Seq_Error (W10.02) diaktifkan agar sistem tidak mengalami deadlock. Pendekatan berbasis bit ini menghilangkan kebutuhan scaling nilai desimal ke integer sekaligus menjadikan komunikasi lebih ringan dan tahan terhadap kesalahan skala.

**Pelatihan Model YOLOv8**

Model dilatih untuk mengenali dua kelas objek, yaitu benda kerja berbentuk bulat dan kotak dengan variasi warna hitam dan merah. Dataset citra dikumpulkan langsung dari lingkungan Plant XYZ agar mewakili kondisi operasional nyata, kemudian seluruh alur kerja mulai dari pelabelan, augmentasi, hingga pelatihan dieksekusi secara terpusat menggunakan platform Roboflow beserta layanan Roboflow Train. Augmentasi data diterapkan untuk memperkaya variasi sampel dan meningkatkan ketahanan model terhadap variasi pencahayaan serta orientasi penempatan. Pendekatan terpadu ini dipilih karena efisien dan menghindari pembebanan komputasi pelatihan yang berat pada Raspberry Pi. Model keluaran terbaik kemudian diunduh dan dipindahkan ke lingkungan lokal Raspberry Pi untuk digunakan pada tahap inferensi melalui kerangka kerja Ultralytics YOLOv8.

**Penetapan Zona dan Mekanisme Koreksi Posisi**

Area kerja Plant XYZ terbagi menjadi enam zona (A, B, C, D, P1, P2), dengan Zona A ditetapkan sebagai home objek Bulat dan Zona C sebagai home objek Kotak, sebagaimana dipetakan pada Gambar 2. Logika koreksi bekerja secara tertutup: untuk setiap deteksi, sistem membandingkan zona tempat objek terdeteksi dengan zona home yang seharusnya berdasarkan hasil klasifikasi. Apabila keduanya cocok, tidak ada tindakan yang diambil. Apabila berbeda, sistem memicu satu siklus *pick-and-place* korektif penuh, yaitu mengangkat objek dari zona keliru, melakukan traverse menuju zona home-nya, lalu melepaskannya. Setiap kejadian koreksi beserta status keberhasilan dan durasinya dicatat secara otomatis. Mekanisme inilah yang mentransformasi platform dari pemilah loop-terbuka menjadi pengoreksi adaptif berbasis persepsi.




**Gambar 2.** Pemetaan enam zona kerja dan penetapan zona home

Logika kendali pada sisi Raspberry Pi disusun ke dalam empat langkah sekuensial yang dikoordinasikan melalui bit handshake. Pada langkah *pre-check*, sistem memeriksa status Safe_to_move sebelum mengaktifkan bit jenis objek dan bit zona asal untuk memulai fase *Pick*. Pada langkah positioning dan pick, PLC mengarahkan silinder X dan Y ke zona asal yang diverifikasi melalui sensor posisi, lalu menurunkan silinder Z dan mengaktifkan vakum hingga ladder menyalakan Seq_Done. Pada langkah sorting dan place, Raspberry Pi mereset Seq_Done dan mengaktifkan bit zona tujuan home sesuai

klasifikasi, lalu PLC mengarahkan manipulator ke zona tersebut. Pada langkah placing dan reset, ladder menonaktifkan vakum untuk melepaskan objek dan menyalakan bit konfirmasi pelepasan, kemudian seluruh bit trigger direset untuk menandakan kesiapan siklus berikutnya.

**Transformasi Koordinat**

Keluaran tahap deteksi berupa koordinat piksel (u, v) bersifat relatif dan tidak dapat langsung digunakan untuk aktuasi. Oleh karena itu, dilakukan transformasi ke koordinat fisik berdasarkan model pinhole camera (Song et al., 2021), yang menghubungkan titik spasial tiga dimensi dengan proyeksinya pada bidang citra melalui parameter intrinsik dan ekstrinsik kamera. Hubungan proyeksi dasar antara titik pada kamera dan bidang citra dinyatakan pada Persamaan (1).

$$u = \frac{f \cdot X_c}{Z_c} \qquad v = \frac{f \cdot Y_c}{Z_c} \tag{1}$$

Panjang fokus fisik (f) diestimasi dari hubungan trigonometris antara diagonal sensor (d) dan medan pandang diagonal kamera (dFOV), sebagaimana dinyatakan pada Persamaan (2), dengan diagonal sensor diperoleh melalui Teorema Pythagoras dari lebar (w) dan tinggi (h) sensor.

$$f = \frac{d}{2\tan\left(\frac{dFOV}{2}\right)} \qquad d = \sqrt{w^2 + h^2} \tag{2}$$

Agar panjang fokus dapat digunakan dalam komputasi berbasis piksel, nilai fisik tersebut dikonversi ke satuan piksel melalui parameter pixel pitch, yaitu ukuran fisik satu piksel pada sensor, seperti pada Persamaan (3).

$$Pixel\ Pitch = \frac{Sensor\ Size}{Image\ Resolution} \qquad f_{pixel} = \frac{f_{mm}}{Pixel\ Pitch} \tag{3}$$

Untuk Logitech C270 dengan dFOV 60°, lebar sensor 3,58 mm, dan tinggi 2,02 mm, perhitungan menghasilkan diagonal sensor sekitar 4,0 mm, panjang fokus fisik sekitar 3,47 mm, dan panjang fokus digital sekitar 694 piksel; nilai inilah yang menginisialisasi matriks intrinsik kamera. Selanjutnya, konversi koordinat piksel ke satuan milimeter dilakukan menggunakan titik tengah offset (C) dan skala rasio milimeter per piksel (S) yang diukur pada meja kerja, sebagaimana pada Persamaan (4).

$$X_{mm} = (X_{pixel} - C_x) \cdot S_x \qquad Y_{mm} = (Y_{pixel} - C_y) \cdot S_y \tag{4}$$

Pada implementasi akhir, koordinat milimeter digunakan untuk pencatatan posisi, sedangkan penentuan arah gerak manipulator didasarkan pada zona kerja hasil klasifikasi kuadran piksel, sehingga sistem tidak bergantung pada presisi koordinat numerik untuk menjalankan aktuasi.

**Metrik Kinerja dan Protokol Evaluasi**

Evaluasi sistem dilakukan terhadap enam metrik kuantitatif. Pertama, kecepatan deteksi (FPS) diukur sebagai jumlah frame yang berhasil diproses model per detik pada setiap siklus, mencerminkan throughput visi pada beban komputasi nyata. Kedua,

Confidence Score adalah tingkat keyakinan model dalam mengklasifikasikan objek per deteksi, dengan ambang validasi ditetapkan di atas 50 persen dan target ideal 90 persen. Ketiga, tingkat keberhasilan FINS dan keempat, tingkat keberhasilan koreksi posisi keduanya dihitung sebagai rasio jumlah operasi sukses terhadap total operasi yang dilakukan, sebagaimana dirumuskan secara umum pada Persamaan (5).

$$Success\ Rate = \frac{N_{sukses}}{N_{total}} \times 100\% \tag{5}$$

Kelima, waktu respons FINS adalah selisih waktu antara pengiriman perintah dan penerimaan acknowledgement dari PLC, dilaporkan dalam nilai minimum, maksimum, dan rata-rata per sesi. Keenam, durasi koreksi adalah waktu yang dibutuhkan untuk menyelesaikan satu siklus pick-and-place korektif dari zona asal ke zona home. Selain itu, akurasi klasifikasi dievaluasi melalui mAP50, precision, dan recall pada validation set, di mana precision dan recall dihitung berdasarkan Persamaan (6), lalu diverifikasi terhadap jumlah misklasifikasi pada kondisi aktual.

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN} \tag{5}$$

dengan TP (true positive) menyatakan deteksi benar, FP (false positive) deteksi keliru, dan FN (false negative) objek yang terlewat. Seluruh metrik diukur pada dua kondisi, yaitu pengujian simulasi dan pengujian aktual lapangan, untuk mengkarakterisasi kesenjangan simulasi-ke-realitas.

## HASIL DAN PEMBAHASAN

Pengujian aktual lapangan dilaksanakan dalam sepuluh sesi yang menempatkan seluruh rantai sistem, dari akuisisi citra hingga eksekusi mekanik, dalam satu siklus terintegrasi. Sistem merekam total 283 siklus deteksi, 1.662 perintah FINS, dan 111 operasi koreksi posisi. Setiap sesi dirancang dengan variasi komposisi benda yang berbeda untuk menguji adaptasi sistem, sebagaimana dirangkum pada Tabel 1.

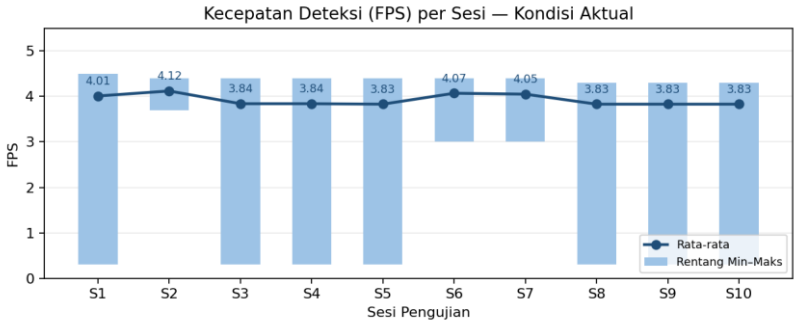
**Tabel 1.** Profil sesi pengujian aktual lapangan

| Sesi | Total Deteksi | Bulat | Kotak | Koreksi Diperlukan |
|---|---|---|---|---|
| 1 | 21 | 14 | 7 | 6 |
| 2 | 43 | 19 | 24 | 12 |
| 3 | 17 | 17 | - | 8 |
| 4 | 17 | 17 | - | 8 |
| 5 | 28 | 28 | - | 13 |
| 6 | 25 | 5 | 20 | 14 |
| 7 | 28 | 7 | 21 | 15 |
| 8 | 28 | 20 | 8 | 9 |
| 9 | 35 | 26 | 9 | 12 |
| 10 | 41 | 30 | 11 | 14 |
| **Total** | **283** | **183** | **100** | **111** |

### Kecepatan Deteksi pada Kondisi Aktual

Hasil pengukuran menunjukkan FPS rata-rata keseluruhan sebesar 3,93 dengan nilai maksimum konsisten 4,3 – 4,5 di seluruh sesi (Gambar 3). Pada kondisi aktual, *frame-*

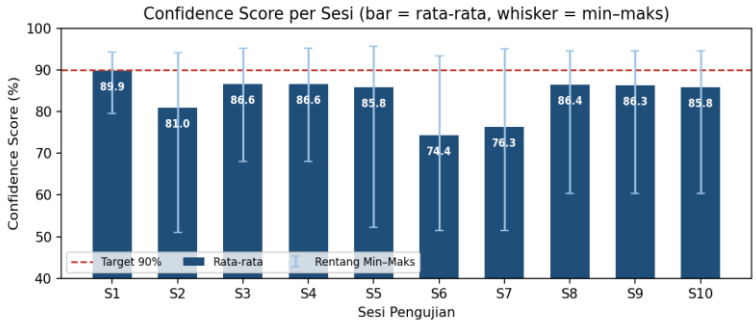
*skip* dinonaktifkan agar tidak ada deteksi yang terlewat, berbeda dengan simulasi. Kemunculan FPS sangat rendah 0,3 bersifat sesaat, terjadi ketika thread komunikasi FINS dan thread inferensi YOLOv8 berebut sumber daya CPU. Meski lebih rendah dari simulasi, FPS ini tidak mengganggu operasional karena satu siklus mekanik pneumatik memerlukan rata - rata 23 detik; dalam rentang itu sistem mampu menghasilkan sekitar 90 frame inspeksi, jauh melampaui kebutuhan satu koordinat valid per siklus.




**Gambar 3.** Kecepatan deteksi (FPS) rata-rata dan rentang min–maks per sesi

**Confidence Score pada Kondisi Aktual**

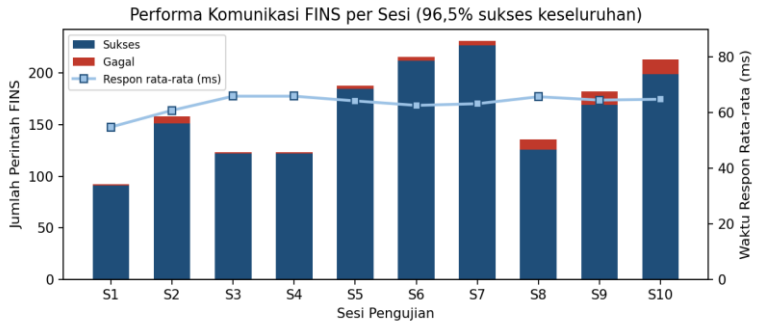
Rata-rata confidence score aktual keseluruhan mencapai 83,9 persen (Gambar 4). Terlihat pola konsisten: sesi yang didominasi objek Kotak Sesi 6: 74,4 persen dan Sesi 7: 76,3 persen menghasilkan confidence lebih rendah dibanding sesi dominan Bulat Sesi 1: 89,9 persen, Sesi 3: 86,6 persen. Fenomena ini berkaitan dengan karakteristik geometri: bentuk bulat memiliki kurva sisi seragam yang mudah dibedakan dari latar tray, sedangkan sudut dan sisi lurus objek Kotak lebih rentan terhadap ambiguitas tepi akibat variasi bayangan dan orientasi. Nilai confidence tertinggi 95,6 persen konsisten dicapai pada objek Bulat di zona standar, mengonfirmasi model bekerja paling optimal ketika objek berada dalam distribusi posisi serupa data pelatihan.




**Gambar 4.** Confidence score per sesi berdasarkan komposisi objek dominan

**Performa Komunikasi FINS**

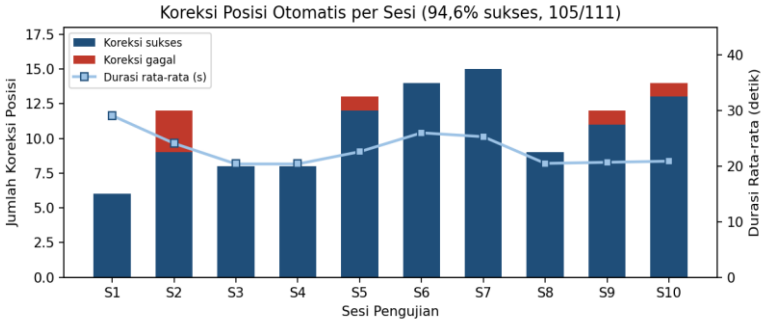
Dari 1.662 perintah FINS, sebanyak 1.604 berhasil dengan tingkat keberhasilan keseluruhan 96,5 persen (Gambar 5). Waktu respons rata-rata berada di rentang 54,8 – 65,9 ms antar sesi dengan nilai minimum sangat konsisten di 50,3 – 50,4 ms, menunjukkan baseline latensi jaringan lokal yang stabil. Sesi 8, 9, dan 10 mencatat kegagalan lebih tinggi (10, 13, dan 14), kemungkinan akibat akumulasi beban termal pada Raspberry Pi 4 setelah beroperasi lebih dari dua jam yang memicu CPU throttling. Meski demikian, mekanisme handshaking dan retry berhasil menangani seluruh kegagalan sesaat tanpa menimbulkan kegagalan mekanik fisik pada Plant XYZ.




**Gambar 5.** Keberhasilan dan waktu respons komunikasi FINS per sesi

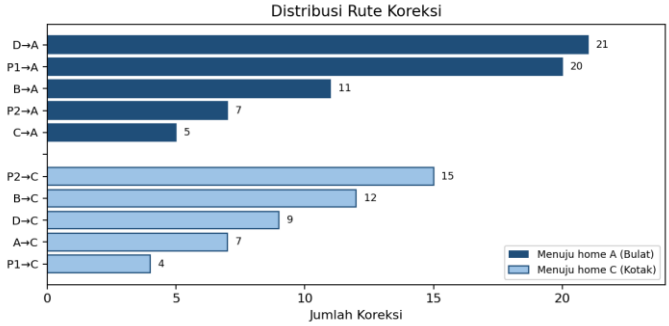
**Akurasi Koreksi Posisi Otomatis**

Sebagai validasi paling fundamental terhadap tujuan utama penelitian, sistem mencatat 111 kejadian koreksi posisi, dengan 105 berhasil dan 6 gagal, menghasilkan tingkat keberhasilan 94,6 persen (Gambar 6). Durasi rata-rata satu koreksi sukses adalah 23,0 detik dengan rentang 15,2 hingga 41,1 detik, mencerminkan jarak tempuh mekanik antara zona asal dan zona tujuan. Koreksi terpanjang terjadi pada Sesi 6 dan 7 yang melibatkan perpindahan lintas zona berjauhan.




**Gambar 6.** Keberhasilan dan durasi koreksi posisi otomatis per sesi

Distribusi rute koreksi (Gambar 7) menegaskan konsistensi logika zona home. Untuk objek Bulat, rute terbanyak adalah D→A (21), P1→A (20), B→A (11), P2→A (7), dan C→A (5), seluruhnya mengarah ke Zona A. Untuk objek Kotak, rute dominan adalah P2→C (15), B→C (12), D→C (9), A→C (7), dan P1→C (4), seluruhnya mengarah ke Zona C. Tidak ada satu pun kasus objek dikirim ke zona yang salah, membuktikan sistem konsisten mengidentifikasi zona tujuan berdasarkan klasifikasi jenis objek. Keenam kegagalan bersifat sporadis dan dapat ditelusuri penyebabnya secara spesifik pada lapisan mekanis, bukan kegagalan sistemik rancangan arsitektur.




**Gambar 7.** Distribusi rute koreksi menuju zona home (A untuk Bulat, C untuk Kotak)

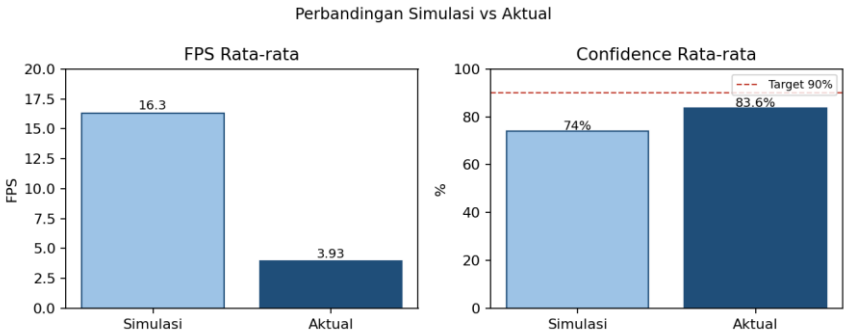
**Perbandingan Simulasi dan Aktual**

Tabel 2 menyandingkan parameter kinerja utama pada kedua kondisi. Perbedaan FPS 16,3 simulasi vs 3,93 aktual merupakan konsekuensi langsung penonaktifan *frame-skip*

pada mode aktual, bukan penurunan kualitas. Menariknya, confidence score aktual 83,9 persen justru melampaui simulasi 74 persen, mengindikasikan kondisi pencahayaan dan penempatan nyata lebih terkontrol dari yang diasumsikan pada skenario simulasi (Gambar 8). Latensi FINS meningkat 63,5 ms vs di bawah 10 ms karena pada kondisi aktual komunikasi berjalan paralel dengan inferensi dan akuisisi frame. Satu parameter konsisten sempurna pada kedua kondisi adalah akurasi klasifikasi 100 persen, di mana tidak ada misklasifikasi pada 283 siklus, selaras dengan mAP50 sempurna saat pelatihan.

**Tabel 2.** Perbandingan pengujian simulasi vs aktual lapangan

| Parameter | Simulasi | Aktual Lapangan |
|---|---|---|
| Total siklus deteksi | 26 | 283 |
| FPS rata-rata | 16,3 | 3,93 |
| FPS minimum / maksimum | 14,0 / 20,5 | 0,3 / 4,5 |
| Confidence rata-rata | 74% | 83,9% |
| Confidence min / maks | 50% / 91% | 51,0% / 95,6% |
| Latensi FINS rata-rata | < 10 ms | 63,5 ms |
| Latensi FINS maksimum | 73 ms | 194,5 ms |
| Tingkat keberhasilan FINS | - | 96,5% (1.604/1.662) |
| Akurasi klasifikasi | 100% | 100% (283/283) |
| Tingkat keberhasilan koreksi | - | 94,6% (105/111) |
| Durasi rata-rata per koreksi | - | 23,0 detik |




**Gambar 8.** Perbandingan FPS dan confidence rata-rata: simulasi vs aktual

Sebagai batasan studi, pengujian dibatasi pada dua kelas geometri Bulat dan Kotak dengan dua warna pada enam zona kerja, dan parameter intrinsik kamera diestimasi dari datasheet tanpa kalibrasi papan catur pada ketinggian aktual. Akumulasi beban termal pada operasi panjang juga belum dikarakterisasi secara khusus dan menjadi ancaman validitas pada uji ketahanan durasi panjang.

## KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan sistem kendali cerdas berbasis Computer Vision dan Machine Learning pada trainer PLC Plant XYZ yang mampu mengidentifikasi jenis dan posisi benda kerja secara otomatis serta mengeksekusi koreksi penempatan melalui protokol FINS UDP/TCP ke PLC Omron CP1H, di mana model YOLOv8 mencapai mAP50 sempurna 100 persen dengan precision 99,1 persen dan recall 100 persen serta tanpa misklasifikasi pada 283 siklus deteksi aktual, arsitektur komunikasi FINS berbasis bit dengan handshaking Safe_to_move dan Seq_Done terbukti andal dengan keberhasilan 96,5 persen dari 1.662 perintah, dan mekanisme koreksi posisi otomatis sebagai tujuan utama berhasil

divalidasi dengan tingkat keberhasilan 94,6 persen dari 111 operasi nyata pada durasi rata-rata 23,0 detik, sehingga secara keseluruhan sistem berhasil mentransformasi Plant XYZ dari sistem pick-and-place loop-terbuka konvensional menjadi sistem adaptif berbasis visi yang mampu mendeteksi, mengklasifikasikan, dan mengoreksi penempatan benda kerja secara mandiri, dengan arah pengembangan berikutnya meliputi peningkatan konsistensi confidence objek Kotak melalui augmentasi dataset, integrasi akselerator inferensi untuk menaikkan FPS, serta pengembangan koreksi multi-objek berbasis antrian.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Program Studi Mekatronika, Jurusan Teknik Otomasi Manufaktur dan Mekatronika, Politeknik Manufaktur Bandung, atas akses fasilitas Plant XYZ Trainer dan sumber daya laboratorium, serta kepada para pembimbing atas bimbingan dan telaah kritis sepanjang penelitian.

## REFERENSI

Almtireen, N., et al. (2025). PLC-controlled intelligent conveyor system with AI-enhanced vision for efficient waste sorting. Applied Sciences, 15(3), 1550. https://doi.org/10.3390/app15031550

Casas, E., Ramos, L., Romero, C., & Rivas-Echeverría, F. (2024). A comparative study of YOLOv5 and YOLOv8 for corrosion segmentation tasks in metal surfaces. Array, 22, 100351. https:// doi.org/10.1016/j.array.2024.100351

Alahdal, R., Chen, W., & Liang, F. (2023). YOLOv8 backbone optimization with C2f modules for precision manufacturing quality inspection. IEEE Access, 11, 56123–56135. https://doi.org/10.1109/ACCESS.2023.3274122

Greco, D., Fasihiany, M., Ranjbar, A. V., Masulli, F., Rovetta, S., & Cabri, A. (2024). Computer vision algorithms on a Raspberry Pi 4 for automated depalletizing. Algorithms, 17(8), 363. https://doi.org/10.3390/a17080363

Alahdal, S., Al-Rahmi, A., & Al-Adhaileh, H. (2023). Performance analysis of YOLOv8 for real-time industrial object detection. Sensors, 23(10), 4782. https://doi.org/10.3390/s23104782

Kusuma, E. D., Aziz, T. F., & Dewanto, W. (2024). Implementasi Modbus TCP dalam pengiriman data deteksi objek ke PLC Schneider M221. Journal of Applied Electrical Engineering, 8(1), 9–16. https://doi.org/10.30871/jaee.v8i1.7334

Maslanka, M., Jancarczyk, D., & Rysinski, J. (2025). Integration of machine vision and PLC-based control for scalable quality inspection in Industry 4.0. Sensors, 25(20), 6383. https://doi.org/10.3390/s25206383

Maulana, G. G., Mada, R., & Purba, R. R. (2022). Automation storage system based on SCADA using PLC CP1H and CP1L. Jurnal Rekayasa Elektrika, 18(3), 144–151. https://doi.org/10.17529/jre.v18i3.26363

Nguyen, V.-T., Nguyen, P.-T., Su, S.-F., Tan, P. X., & Bui, T.-L. (2025). Vision-based pick and place control system for industrial robots using an eye-in-hand camera. IEEE Access, 13, 25127–25140. https://doi.org/10.1109/ACCESS.2025.3536496

Nugroho, B., et al. (2025). Implementation of sparrow pest detection using YOLOv8 method on Raspberry Pi and Google Coral USB Accelerator. Sebatik, 29(1), 129–135. https://doi.org/10.46984/sebatik.v29i1.2558

Pamungkas, D. S., Saputra, S., & Pelmelay, A. A. (2025). Rancang bangun dan evaluasi sistem sortir otomatis barang dengan metode deteksi objek YOLO v5 dan kendali PLC Outseal. Journal of Applied Computer Science and Technology, 6(1), 57–66. https://doi.org/10.52158/jacost.v6i1.1165

Ridwan, Erdani, Y., Abadi, S. C., Hidayat, M. D. A., Harits, A. R., & Pratama, R. A. (2025). Developing sorting algorithm for SmartEdu Conveyor using computer vision technology. INFORMATIK Journal, 21(3), 207–213. https://doi.org/10.52958/iftk.v21i3.12434

Song, Q., Li, S., Bai, Q., Yang, J., Zhang, X., Li, Z., & Duan, Z. (2021). Object detection method for grasping robot based on improved YOLOv5. Micromachines, 12(11), 1273 https://doi.org/10.3390/mi12111273

Sutisna, T., Raharja, A. R., Solihin, Hariyadi, E., & Putra, V. H. C. (2024). Penggunaan computer vision untuk menghitung jumlah kendaraan dengan metode SSD. Innovative: Journal of Social Science Research, 4(2), 6060–6067. https://doi.org/10.31004/innovative.v4i2.10071

Zaini, S. N. A. M., Aras, M. H. M., Rashid, M. Z. A., & Hashim, N. M. Z. (2024). Automation of sustainable industrial machine using computer vision and machine learning: A review. Journal of Robotics and Control, 5(2), 341–356. https://doi.org/10.18196/jrc.v5i2.21151

Khan, A. M., Hussain, I., & Ahmad, M. (2022). Logitech C270 performance evaluation for real-time machine vision in embedded systems. Measurement, 190, 110694. https://doi.org/10.1016/j.measurement.2022.110694