

# Sistem Pengenalan dan Pengelompokan Bentuk Objek Secara Otomatis Menggunakan Robot Lengan 3-DOF

Muhammad Ryan Pratama Nugroho  
Departemen Teknik Elektro  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
ryanpratamanugroho@gmail.com

**Abstrak** — Sistem deteksi objek memiliki peran krusial dalam berbagai aplikasi otomasi. Namun, mengimplementasikan algoritma deep learning yang kompleks pada perangkat embedded dengan sumber daya terbatas menjadi tantangan. Oleh karena itu, penelitian ini berfokus pada pengembangan sistem pengenalan dan pengelompokan bentuk objek secara otomatis menggunakan Raspberry Pi 5 dan robot lengan 3-DOF. Tujuan utamanya adalah mengimplementasikan pengenalan objek pada Raspberry Pi, menganalisis pengaruh jumlah epoch terhadap akurasi sistem, serta mengintegrasikan sistem pengenalan objek dengan robot lengan 3-DOF untuk pengelompokan otomatis. Penelitian ini dibatasi pada penggunaan Raspberry Pi 5 sebagai unit kontrol utama, penerapan algoritma YOLOv8 untuk deteksi objek, pengenalan tiga bentuk geometri sederhana (segitiga, persegi, lingkaran).

Diharapkan sistem ini mampu menunjukkan kinerja deteksi objek yang efisien dan akurat pada perangkat embedded Raspberry Pi. Keberhasilan penelitian akan diukur secara kuantitatif berdasarkan akurasi pengenalan objek dan kemampuan robot lengan dalam mengelompokkan objek berdasarkan bentuknya. Hasil penelitian ini diharapkan memberikan kontribusi signifikan bagi berbagai bidang, termasuk industri otomasi, pertanian cerdas, sistem keamanan, dan sebagai platform edukasi untuk studi visi komputer dan robotika, dengan menawarkan solusi yang ekonomis, portabel, dan efisien.

**Kata kunci**— *pengenalan objek, YOLOv8, Raspberry Pi, robot lengan 3-DOF, klasifikasi bentuk*

## I. PENDAHULUAN

Teknologi Sistem deteksi objek memiliki beragam aplikasi dalam kehidupan sehari-hari, mencakup aspek keamanan, otomasi industri, transportasi otonom, serta sistem pembelajaran interaktif. Teknologi ini memungkinkan mesin untuk mengenali dan melacak objek dalam lingkungan sekitarnya melalui pengolahan citra digital. Kemampuan ini menjadi kunci dalam mendorong efisiensi dan akurasi dalam berbagai proses otomatisasi [1].

Seiring dengan kemajuan teknologi, pemanfaatan perangkat komputasi kecil dan efisien seperti Raspberry Pi memungkinkan pengembangan sistem deteksi objek yang lebih ekonomis dan mudah diterapkan. Raspberry Pi merupakan komputer papan tunggal (single-board computer) yang mendukung berbagai sistem operasi berbasis Linux dan mampu menjalankan aplikasi-aplikasi pemrosesan gambar dengan library seperti OpenCV, TensorFlow Lite, dan PyTorch [2], [3]. Dengan harga yang terjangkau, ringan, dan hemat energi, sehingga membuka peluang besar dalam pengembangan sistem deteksi objek yang ekonomis, portabel, dan dapat digunakan dalam berbagai skenario nyata [3], [4].

Pengenalan objek adalah salah satu cabang dari visi komputer di mana robot dapat mendeteksi objek di suatu lingkungan menggunakan kamera atau sensor yang mampu mengekstrak gambar lingkungan sekitar robot [1]. Implementasi sistem deteksi objek menggunakan Raspberry Pi umumnya melibatkan kamera sebagai sensor masukan untuk

menangkap citra atau video, kemudian citra tersebut diproses untuk dilakukan ekstraksi fitur dan klasifikasi objek. Beberapa penelitian telah berhasil mengembangkan sistem seperti media pembelajaran interaktif berbasis deteksi bentuk dan warna [5], robot penyortir objek secara otomatis berdasarkan bentuk dan warna [6], serta sistem pendeteksi objek 3D berbasis YOLOv8 dengan tingkat presisi tinggi [7].

Meski demikian, terdapat sejumlah tantangan dalam mengimplementasikan algoritma deteksi objek di Raspberry Pi. Keterbatasan sumber daya perangkat keras seperti kapasitas memori dan kecepatan pemrosesan dapat menjadi hambatan saat menjalankan model deep learning yang kompleks. Oleh karena itu, diperlukan model dan pendekatan yang telah dioptimasi, seperti TensorFlow Lite atau versi ringan dari YOLO, yang secara khusus dirancang untuk perangkat edge atau embedded (tertanam) [8], [9].

Penelitian mengenai robot lengan untuk otomasi pengemasan telah banyak dilakukan. Salah satunya adalah penelitian oleh Petrina Tsaubantani (2024) yang berjudul “Pengembangan Robot Lengan 3-DOF untuk Otomatisasi Pengemasan Berdasarkan Deteksi Warna dengan Sensor PixyCam”. Penelitian tersebut menggunakan PixyCam untuk mendeteksi objek berdasarkan warna dan bentuk sederhana. Meskipun sistem ini berhasil melakukan pengenalan objek secara otomatis, penelitian tersebut masih memiliki keterbatasan signifikan. Pertama, PixyCam hanya mampu mengenali fitur visual sederhana, sehingga kinerjanya menurun pada kondisi pencahayaan yang berbeda atau variasi objek yang kompleks. Kedua, sistem hanya mampu membedakan dua bentuk objek saja, yaitu persegi panjang dan persegi, sehingga fleksibilitasnya terbatas untuk aplikasi industri yang memerlukan klasifikasi bentuk lebih beragam.

Keterbatasan ini menunjukkan adanya kebutuhan akan sistem yang lebih adaptif dan akurat. Oleh karena itu, penelitian ini mengusulkan pengembangan sistem pengenalan dan pengelompokan bentuk objek menggunakan algoritma YOLOv8 yang terintegrasi dengan robot lengan 3-DOF. Dengan kemampuan deep learning, YOLOv8 memungkinkan pengenalan berbagai bentuk objek dengan akurasi tinggi secara real-time, sehingga diharapkan mampu mengatasi keterbatasan penelitian sebelumnya yang masih bergantung pada PixyCam dan klasifikasi dua bentuk objek saja.

Berdasarkan latar belakang tersebut, penelitian ini akan berfokus pada implementasi sistem pengenalan objek menggunakan algoritma YOLOv8 yang terintegrasi dengan robot lengan 3-DOF dan menggunakan Raspberry Pi sebagai komputer utama. Dengan pendekatan ini, diharapkan dapat dihasilkan sistem deteksi objek yang efisien, akurat, serta layak secara ekonomis untuk diaplikasikan pada skala kecil maupun menengah.

## II. KAJIAN TEORI

Menyajikan dan menjelaskan teori-teori yang berkaitan dengan variabel-variabel penelitian. Poin subjudul ditulis dalam abjad.

### A. Kecerdasan Buatan dan Pembelajaran Mesin

Artificial Intelligence (AI) merupakan cabang ilmu komputer yang bertujuan menciptakan sistem yang mampu

meniru kecerdasan manusia, seperti berpikir, belajar, dan mengambil keputusan. Salah satu turunan penting dari AI adalah *Machine Learning (ML)*, yang memungkinkan sistem komputer belajar dari data tanpa diprogram secara eksplisit. Dalam konteks deteksi objek, ML digunakan untuk mengenali pola visual dari gambar dan mengkategorikannya ke dalam kelas tertentu.

*Supervised learning*, sebagai pendekatan umum dalam ML, melibatkan pelatihan model menggunakan dataset berlabel. Model kemudian belajar untuk mengasosiasikan fitur input (misalnya piksel gambar) dengan label yang sesuai (misalnya “lingkaran” atau “segitiga”). Setelah pelatihan selesai, model dapat digunakan untuk mengklasifikasi data baru yang belum pernah dilihat sebelumnya.

Lebih lanjut, *Deep Learning (DL)* merupakan subbidang dari ML yang menggunakan jaringan saraf tiruan bertingkat (*deep neural networks*) untuk melakukan pembelajaran representasi data yang kompleks. Salah satu arsitektur DL yang paling efektif untuk pemrosesan gambar adalah *Convolutional Neural Network (CNN)*.

#### B. Convolutional Neural Network (CNN)

CNN dirancang untuk bekerja pada data berbentuk grid, seperti gambar. CNN terdiri atas beberapa lapisan (*layers*), seperti lapisan konvolusi, aktivasi (ReLU), *pooling*, dan *fully connected layers*. Lapisan konvolusi bertugas mengekstrak fitur lokal dari gambar, seperti tepi dan bentuk dasar. *Pooling* bertugas mereduksi dimensi data untuk efisiensi komputasi, sedangkan *fully connected layers* menghasilkan prediksi klasifikasi.

Keunggulan CNN adalah kemampuannya dalam *feature extraction* otomatis. Ini menjadikannya pilihan utama untuk tugas-tugas pengenalan objek, pengenalan wajah, dan deteksi objek real-time. CNN telah menjadi fondasi bagi algoritma- algoritma deteksi objek modern, termasuk YOLO, SSD, dan Faster R-CNN.

#### C. Algoritma Deteksi Objek: YOLO

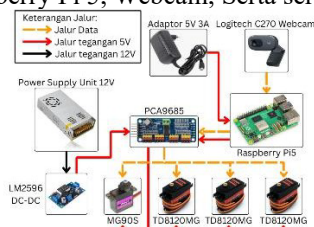
*You Only Look Once (YOLO)* adalah salah satu algoritma deteksi objek yang terkenal karena kecepatannya dan efisiensinya. Tidak seperti algoritma deteksi dua tahap seperti Faster R-CNN yang melakukan *region proposal* terlebih dahulu, YOLO melakukan prediksi *bounding box* dan klasifikasi objek secara langsung dalam satu langkah (*one-stage detector*). Ini menjadikannya sangat cocok untuk aplikasi real-time.

YOLO membagi citra input menjadi grid, kemudian setiap grid bertugas memprediksi beberapa *bounding box* dan probabilitas kelas dari objek di dalam grid tersebut. YOLOv8, sebagai versi terbaru, membawa berbagai peningkatan performa, seperti arsitektur model yang lebih ringan, penggunaan *anchor-free detection*, dan integrasi dengan pustaka *Ultralytics* yang memudahkan pelatihan serta inferensi di berbagai platform, termasuk Raspberry Pi.

### III. HASIL DAN PEMBAHASAN

#### A. Rancangan Sistem

Pada gambar di bawah ini menjelaskan tentang perancangan sistem yang terdiri dari beberapa komponen inti yaitu Raspberry Pi 5, Webcam, Serta servo-servo



GAMBAR 1

Desain perangkat keras

#### B. Sub Sistem Pemrosesan Citra

Pada Sub sistem pemrosesan citra merupakan komponen utama dalam sistem pengenalan objek. Perangkat keras yang digunakan pada sub sistem ini meliputi kamera Logitech C270 sebagai sensor input dan Raspberry Pi 5 sebagai pusat pemrosesan. Kamera bertugas untuk menangkap citra dari lingkungan kerja (*workspace*), baik berupa gambar diam maupun video secara real-time. Citra yang ditangkap kemudian dikirim ke Raspberry Pi untuk diproses lebih lanjut.

Raspberry Pi akan menjalankan algoritma deteksi objek YOLOv8 (*You Only Look Once* versi 8) yang telah dilatih sebelumnya menggunakan dataset bentuk-bentuk objek. YOLOv8 dipilih karena memiliki kemampuan deteksi yang cepat dan akurat, serta efisien untuk dijalankan pada perangkat embedded seperti Raspberry Pi. Model YOLO akan menghasilkan *bounding box* dan label klasifikasi dari objek-objek yang terdeteksi dalam citra.

Untuk implementasi, digunakan bahasa pemrograman Python karena kompatibel dengan berbagai pustaka open-source, serta mudah digunakan dalam platform seperti Raspberry Pi. Proses pengolahan citra dilakukan dengan bantuan pustaka OpenCV (*OpenSource Computer Vision Library*), yang mendukung manipulasi citra dan integrasi dengan YOLO.

#### C. Sub Sistem Lengan Robot

Pada Sub sistem ini berfungsi untuk menangani respons fisik terhadap hasil deteksi objek dari sistem pengolahan citra. Perangkat keras yang terlibat adalah servo motor TD8120MG dan Raspberry Pi 5. Lengan robot akan bergerak sesuai instruksi endpoint yang telah ditentukan berdasarkan lokasi objek yang terdeteksi oleh sistem visi.

Servo motor TD8120MG dipilih karena kemampuannya menghasilkan torsi tinggi serta mendukung kendali posisi presisi melalui komunikasi serial digital. Servo ini mampu merespons perintah yang diberikan oleh Raspberry Pi untuk melakukan gerakan seperti mengambil, memindahkan, atau menyusun objek berdasarkan klasifikasi yang telah ditentukan.

Raspberry Pi tidak hanya berfungsi sebagai pengontrol pengolahan citra, tetapi juga sebagai controller utama sistem aktuator, yang mengatur komunikasi dengan servo menggunakan protokol digital. Dalam sistem ini, endpoint posisi untuk setiap jenis objek akan dikodekan dalam skrip Python, dan akan dieksekusi secara otomatis setelah deteksi objek berhasil dilakukan.

### IV. HASIL DAN PEMBAHASAN

Data akurasi diperoleh melalui pengujian sistem setelah model YOLOv8 dilatih dengan jumlah epoch yang berbeda. Pengujian dilakukan secara konsisten terhadap dataset uji yang sama untuk menjaga validitas hasil. Objek yang diuji meliputi tiga bentuk geometri: segitiga, persegi, dan lingkaran. Setiap konfigurasi epoch diuji sebanyak beberapa kali (*batch*) untuk memastikan kestabilan performa model.

Pada konfigurasi epoch 20, akurasi deteksi pada sebagian bentuk masih cukup bervariasi dan cenderung lebih rendah. Ketika model dilatih hingga 60 epoch, terlihat adanya peningkatan konsistensi dan akurasi pada ketiga bentuk. Untuk menguji stabilitas, dilakukan dua batch pengujian terpisah pada epoch 60. Selanjutnya, epoch 100 digunakan sebagai pelatihan lanjutan untuk mengetahui apakah peningkatan epoch secara signifikan terus berbanding lurus dengan peningkatan akurasi, atau justru mengalami stagnasi.

#### A. Hasil Analisis Statistik

Untuk memberikan pemahaman lebih lanjut mengenai

kinerja sistem, dilakukan analisis statistik sederhana terhadap hasil pengujian. Perhitungan mencakup rata-rata akurasi (mean), akurasi minimum, dan akurasi maksimum untuk masing-masing bentuk objek pada setiap kelompok epoch. Hasil analisis ini disajikan dalam bentuk tabel dan grafik untuk memudahkan interpretasi.

Hasil menunjukkan bahwa peningkatan jumlah epoch dari 20 ke 60 secara umum menghasilkan lonjakan akurasi yang signifikan, terutama untuk bentuk persegi dan lingkaran. Nilai rata-rata meningkat dan rentang variasi hasil antar percobaan menjadi lebih sempit, menandakan model semakin stabil. Namun, ketika epoch dinaikkan menjadi 100, peningkatan akurasi mulai melambat atau bahkan mendatar. Ini mengindikasikan adanya titik optimal pelatihan, di mana penambahan epoch tidak lagi memberikan perbaikan signifikan, dan justru bisa mengarah pada overfitting jika tidak diatur dengan baik.

Selain itu, nilai akurasi minimum dan maksimum untuk epoch 60 dan 100 lebih tinggi dibandingkan epoch 20, yang menunjukkan bahwa sistem menjadi lebih andal setelah pelatihan lebih dalam. Hal ini memperkuat dugaan bahwa jumlah epoch memiliki peran penting dalam meningkatkan akurasi pengenalan objek, namun tetap perlu disesuaikan dengan kompleksitas dataset dan keterbatasan perangkat keras.

TABEL 1  
Ringkasan Akurasi Rata-rata, Minimum, dan Maksimum.

Epoch	Objek	Rata-rata Akurasi (%)	Akurasi Minimum (%)	Akurasi Maksimum (%)
20	Persegi	79.5%	73	88
	Segitiga	88.6%	83	97
	Lingkaran	91.8%	89	96
60	Persegi	85.0%	78	94
	Segitiga	93.5%	91	98
	Lingkaran	93.4%	91	97
100	Persegi	90.5%	88	94
	Segitiga	92.0%	89	96
	Lingkaran	91.9%	89	97

## B. Pembahasan Hasil

Berdasarkan analisis data pada Tabel 1, dapat diamati beberapa tren penting mengenai pengaruh jumlah epoch terhadap akurasi sistem pengenalan objek:

1. Peningkatan Akurasi dengan Penambahan Epoch: Secara umum, terjadi peningkatan rata-rata akurasi pada semua jenis objek ketika jumlah epoch ditingkatkan dari 20, 60, hingga 100. Fenomena ini konsisten dengan prinsip dasar machine learning di mana model membutuhkan iterasi pelatihan yang cukup untuk mengoptimalkan parameter internalnya (bobot dan bias) agar dapat mempelajari pola-pola kompleks dalam data. Pada epoch awal (misalnya 20), model mungkin masih dalam tahap underfitting, di mana ia belum sepenuhnya menangkap hubungan antara fitur input (gambar objek) dan label output (bentuk objek). Dengan lebih banyak epoch, model memiliki kesempatan lebih besar untuk melakukan gradient descent dan meminimalkan fungsi loss, sehingga konvergensi menuju solusi yang lebih optimal dapat tercapai.

Untuk objek Persegi, rata-rata akurasi menunjukkan peningkatan yang stabil dari 79.5% (epoch 20) menjadi 85.0% (epoch 60) dan kemudian mencapai 90.6% (epoch 100). Peningkatan ini menunjukkan bahwa objek persegi, yang mungkin memiliki variasi orientasi atau deformasi yang lebih menantang pada epoch awal, dapat dikenali dengan lebih baik setelah model dilatih lebih lama. Rentang akurasi pada epoch 20 (73-88%) juga lebih lebar

dibandingkan epoch

60 (78-94%) dan epoch 100 (88-94%), mengindikasikan bahwa pelatihan yang lebih panjang tidak hanya meningkatkan rata-rata tetapi juga mengurangi variabilitas kinerja, menjadikan deteksi lebih konsisten.

Objek Segitiga juga menunjukkan peningkatan rata-rata akurasi yang signifikan dari 88.6% (epoch 20) menjadi 93.5% (epoch 60). Pada epoch 100, akurasi sedikit menurun menjadi 92.4%, namun tetap secara substansial lebih tinggi daripada epoch 20. Variasi akurasi untuk segitiga pada epoch 20 (83-97%) juga lebih besar dibandingkan epoch 60 (91-98%) dan epoch 100 (89-96%). Ini mengindikasikan bahwa model menjadi lebih stabil dan akurat dalam mendeteksi segitiga setelah pelatihan yang lebih intensif, meskipun ada potensi plateau atau sedikit fluktuasi pada epoch yang sangat tinggi.

Untuk objek Lingkaran, rata-rata akurasi menunjukkan peningkatan dari 91.8% (epoch 20) menjadi 93.4% (epoch 60). Pada epoch 100, akurasi sedikit menurun menjadi 92.4%, namun tetap lebih tinggi dari epoch 20. Peningkatan ini konsisten, meskipun objek lingkaran sudah memiliki akurasi yang relatif tinggi sejak epoch 20. Hal ini mungkin disebabkan oleh karakteristik geometris lingkaran yang lebih sederhana dan lebih mudah diekstraksi fiturnya oleh CNN dibandingkan bentuk poligon seperti persegi atau segitiga. Meskipun demikian, penambahan epoch tetap memberikan sedikit peningkatan atau setidaknya mempertahankan akurasi tinggi yang telah dicapai.

2. Variabilitas Akurasi pada Epoch yang Sama dan Konsistensi Peningkatan Akurasi: Meskipun ada peningkatan rata-rata, terlihat adanya variasi akurasi (rentang antara minimum dan maksimum) dalam satu kelompok epoch yang sama. Hal ini wajar dalam proses pelatihan machine learning dan dapat dipengaruhi oleh faktor-faktor seperti batch data yang diproses, inisialisasi bobot, atau kondisi lingkungan pengujian. Namun, yang lebih penting adalah konsistensi peningkatan akurasi rata-rata dari epoch 20 ke epoch 60 dan epoch 100 pada ketiga jenis objek. Ini mendukung hipotesis bahwa jumlah epoch memiliki pengaruh positif terhadap akurasi sistem pengenalan objek. Semakin banyak epoch, model memiliki lebih banyak kesempatan untuk menyesuaikan bobotnya dan mengurangi error, sehingga menghasilkan deteksi yang lebih akurat.

3. Perbandingan Kinerja Antar Objek: Objek Lingkaran cenderung memiliki akurasi tertinggi di epoch 20 dan 60, dan tetap tinggi pada epoch 100, diikuti oleh Segitiga, dan kemudian Persegi. Urutan kinerja ini dapat dijelaskan oleh kompleksitas bentuk geometris dan variasi yang mungkin terjadi dalam dataset. Lingkaran adalah bentuk yang paling simetris dan memiliki fitur yang paling konsisten, sehingga lebih mudah dikenali oleh model. Segitiga memiliki tiga sisi dan tiga sudut, yang mungkin sedikit lebih bervariasi dalam representasi pikselnya. Sementara itu, persegi, meskipun terlihat sederhana, dapat memiliki variasi orientasi yang signifikan (misalnya, persegi yang diputar 45 derajat akan terlihat seperti belah ketupat), yang mungkin memerlukan model untuk mempelajari lebih banyak fitur untuk generalisasi yang baik. Ini menunjukkan bahwa robusta sistem terhadap variasi bentuk objek juga dipengaruhi oleh kompleksitas intrinsik dari bentuk itu sendiri.

4. Implikasi Terhadap Overfitting: Penting untuk dicatat bahwa meskipun peningkatan epoch umumnya meningkatkan akurasi, ada titik di mana penambahan epoch mungkin tidak lagi memberikan peningkatan akurasi yang signifikan pada data validasi atau bahkan dapat menyebabkan overfitting. Overfitting terjadi ketika model terlalu "menghafal" data pelatihan dan kehilangan kemampuannya

untuk menggeneralisasi ke data baru yang belum pernah dilihat. Dalam kasus ini, akurasi pada data pelatihan akan terus meningkat, tetapi akurasi pada data pengujian atau validasi akan stagnan atau bahkan menurun. Dari data yang tersedia, epoch 60 menunjukkan puncak akurasi rata-rata untuk segitiga dan lingkaran, dengan sedikit penurunan pada epoch 100. Ini bisa menjadi indikasi awal bahwa model mungkin mulai mendekati titik overfitting atau bahwa peningkatan epoch lebih lanjut tidak akan memberikan manfaat signifikan. Untuk pelatihan di masa depan, pemantauan akurasi validasi sangat penting untuk menentukan epoch optimal.

## V. KESIMPULAN

Hasil pengujian menunjukkan bahwa peningkatan jumlah epoch dari 20 ke 60 secara signifikan meningkatkan akurasi deteksi objek, yaitu sebesar 4,0% untuk persegi, 4,9% untuk segitiga, dan 1,6% untuk lingkaran. Namun, dari epoch 60 ke 100, peningkatan akurasi cenderung melambat, bahkan menurun untuk segitiga dan lingkaran. Puncak akurasi untuk persegi tercapai di epoch 100 (90,5%), sedangkan untuk segitiga dan lingkaran justru terbaik pada epoch 60. Stabilitas model juga meningkat, ditandai dengan menyempitnya rentang selisih akurasi antar percobaan. Secara keseluruhan, epoch 60 menjadi titik pelatihan yang optimal, di mana akurasi cukup tinggi tanpa indikasi overfitting. Temuan ini menggarisbawahi pentingnya penentuan jumlah epoch yang tepat untuk mencapai performa maksimal dengan efisiensi pelatihan yang baik.

## REFERENSI

- [1] L. K. Thakur et al., "Design and development of 5-dof robotic arm," 2023.
- [2] S. A. Kouritem et al., "A multi-objective optimization," Alexandria Eng. J., 2022.
- [3] A. Koksall et al., "Effect of annotation errors," CVPRW, 2020.
- [4] Z. Hua et al., "A Benchmark Review of YOLO," IEEE Access, 2025.
- [5] M. Hussain, "YOLO-v1 to YOLO-v8," Machines, 2023.
- [6] D. D. Aboyomi, "YOLO vs. SSD vs. Faster R-CNN," ITEJ, 2023.
- [7] J. A. Shariff, "Automated Object Sorting," 2022.
- [8] S. Patil, "Real time object detection," Electronicwings, 2017.
- [9] C. Rue and C. Uttraphan, "Shape Recognition using YOLOv8," EEEE, 2023.
- [10] H. Mulyawan et al., "Image Processing Secara Real Time," EEPIS, 2011.