

# Pengembangan Sistem Robot Arm dengan Integrasi Motion Capture dan Objek Deteksi untuk Penempatan Barang Otomatis

*Tsany Ammar Rasyid<sup>a</sup>, Ahmad Alfaruqi Haqinullah<sup>b</sup>, Dzaki Fajri Arrafi<sup>c</sup>, Said Abdurrahman<sup>d</sup>, Ardy Seto Priambodo<sup>e</sup>*

*<sup>a,b,c,d,e</sup> Department of Electrical & Electronics Engineering, Universitas Negeri Yogyakarta, Indonesia*

## ARTICLE INFO

### Article history:

Received 2 August 2024

Received in revised form

10 September 2024

Accepted 20 October 2024

Available online 31 November 2024

### Keywords:

Arm Robot

Motion Capture

Objek Deteksi

## ABSTRACT

Pada perkembangan teknologi robotika, khususnya robot arm 4 DOF, terdapat tantangan dalam mengintegrasikan teknologi motion capture dan deteksi objek. Tantangan tersebut dinamakan otomatisasi penempatan barang secara akurat. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem robot arm yang mampu merekam gerakan manusia, mendeteksi warna objek, dan menempatkan barang pada arah vertikal maupun horizontal secara presisi. Dalam pelaksanaannya, teknologi ini menggunakan gabungan antara mikrokontroler ESP32, algoritma motion capture berdasarkan interpolasi linier sebagai kontrol servo, hingga algoritma Canny Edge Detection untuk deteksi objek. Untuk meningkatkan keakuratan dari proses visualisasi data digunakan olah ruang warna HSV, sementara itu untuk menentukan gerakan robot diperlukan implementasi inverse kinematics. Pengujian sistem dilakukan dengan mengimplementasikan ciri motion capture dan deteksi objek melalui mekanisme pick and place. Robot dapat merekam dan mengulang gerak gerik dengan akurasi tinggi dan berhasil menyelesaikan tugas deteksi dan penempatan berdasarkan warna dengan tingkat keberhasilan 83,3% dalam uji coba enam percobaan. Namun, keberhasilannya terbatas oleh kondisi penerangan ruangan dan posisi objek yang berubah-ubah. Robot ini memiliki potensi yang sangat baik untuk industri otomatisasi, tetapi kemampuannya dapat ditingkatkan lagi dalam melakukan deteksi objek deteksi dan tingkat keandalan.

## 1 Pendahuluan

Perkembangan teknologi robotika telah mengalami kemajuan besar dalam era modern saat ini, terutama dalam penerapan robot arm, atau lengan robot, di berbagai bidang industri dan penelitian. Robot arm 4 DOF, atau Degrees of Freedom, adalah salah satu inovasi baru yang memungkinkan proses penempatan barang yang sangat tepat dilakukan secara otomatis. Robot arm ini dapat meningkatkan produktivitas dan efisiensi dalam berbagai aplikasi, mulai dari manufaktur hingga layanan kesehatan, karena mereka dapat meniru gerakan manusia dan berinteraksi dengan lingkungan sekitar.[1]

Integrasi teknologi pengambilan gerakan dan objek deteksi adalah komponen penting dalam pengembangan sistem robot arm. Dengan motion capture, data gerakan dapat diambil secara real-time. Ini dapat digunakan untuk mengontrol gerakan robot dengan lebih akurat. Teknologi ini mengumpulkan posisi dan gerakan objek melalui sensor dan kamera. Kemudian, data ini diubah menjadi data digital yang dapat diproses (Miller & Johnson, 2020). Di sisi lain, objek deteksi berfungsi untuk mengidentifikasi dan mengklasifikasikan objek di sekitar robot, sehingga robot dapat melakukan tindakan yang sesuai berdasarkan informasi yang diperoleh [2].

Meskipun teknologi ini menjanjikan, ada beberapa masalah saat memilih perangkat lunak dan perangkat keras yang tepat. Sebagai otak sistem robot arm, mikrokontroler ESP32 memberikan fleksibilitas dalam pengolahan data dan koneksi. Pemilihan motor servo yang tepat juga penting untuk mendapatkan kontrol posisi yang tepat. Motor servo mampu memberikan gerakan yang halus dan akurat, yang esensial dalam aplikasi robotika. [3]

## 2 Studi Literatur

Dalam perancangan sebuah robot ARM 4 DOF diperlukan beberapa aspek penting untuk dapat menciptakan suatu robot ARM 4 DOF yang cerdas dan adaptif. Keberhasilan dari pengimplementasian setiap fitur pada robot sangat bergantung pada pemilihan dan pengaturan dari perangkat lunak dan keras yang sesuai. Digunakannya ESP32 sebagai mikrokontroler memungkinkan untuk melakukan pengolahan data yang cepat dan memiliki koneksi yang sangat fleksibel. Motor Servo sebagai aktuator utama, digunakan sebagai penggerak lengan dari robot, perannya sangatlah penting untuk dapat menjalankan robot dengan baik. Motion capture dan object detection yang menjadi fitur utama yang dihadirkan pada robot digunakan

untuk mengamati dan meniru gerakan manusia dan untuk memahami dan berinteraksi pada lingkungan disekitarnya.

### 2.1 ESP32

Mikrokontroler ESP32 merupakan mikrokontroler SoC (System on Chip) terpadu dengan dilengkapi WiFi 802.11 b/g/n, Bluetooth versi 4.2, dan berbagai peripheral. ESP32 adalah chip yang cukup lengkap, terdapat prosesor, penyimpanan dan akses pada GPIO (General Purpose Input Output). ESP32 bisa digunakan untuk rangkaian pengganti pada Arduino, ESP32 memiliki kemampuan untuk mendukung terkoneksi ke WIFI secara langsung (Agus Wagyan, 2019). Adapun spesifikasi dari ESP32 adalah sebagai berikut: Board ini memiliki dua versi, yaitu 30 GPIO dan 36 GPIO. Keduanya memiliki fungsi yang sama tetapi versi yang 30 GPIO dipilih karena memiliki dua pin GND. Semua pin diberi label di bagian atas board sehingga mudah untuk dikenali. Board ini memiliki interface USB to UART yang mudah diprogram dengan program pengembangan aplikasi seperti Arduino IDE. Sumber daya board bisa diberikan melalui konektor micro USB [4].



Gambar 1. ESP32

### 2.2 Motor Servo

Motor Servo merupakan sebuah aktuator yang sering digunakan dalam sebuah robot karena motor servo memiliki kemampuan untuk dapat memberikan kontrol posisi yang presisi [6][7]. Motor servo sendiri adalah sebuah motor dengan sistem *closed feedback* [5]. Motor servo terdiri dari sebuah motor, gear, potensiometer, dan rangkaian kontrol. Batas setiap sudut pada putaran motor servo diatur menggunakan potensiometer, lalu untuk sudut dari sumbu servo diatur oleh lebar pulsa yang dikirim dari sinyal. Motor servo yang biasa digunakan dalam sebuah robot arm adalah servo dengan jenis MG996R dan MG90R.



Gambar 2. Servo MG996R

MG996R merupakan salah satu jenis servo yang memiliki torsi yang tinggi yang menggunakan mekanisme *gear* berupa logam, mekanisme *gear* logam ini memungkinkan servo untuk mengangkat beban yang relatif berat [7]. Servo jenis ini memiliki torsi hingga 11 kg/cm dengan tegangan input sebesar 6V. Jenis servo ini memiliki 2 jenis yang berbeda, yaitu yang memiliki putaran 180 derajat dan 360 derajat.



Gambar 3. Servo MG90R

MG90R merupakan sebuah *micro servo* yang juga memiliki *gear* berbahan logam. Jenis servo ini memiliki torsi sekitar 2,2 kg/cm dengan tegangan 6V. Servo MG90R ini cocok digunakan dalam robot arm 4 DOF pada bagian *gripper*. Ukurannya yang relatif kecil membuat jenis servo ini ideal digunakan pada sebuah robot arm.

### 2.3 Motion Capture

*Motion Capture* merupakan suatu proses penangkapan gerakan yang ada di lingkungan ke dalam komputer. Saat ini *Motion Capture* sudah menjadi teknik paling efektif untuk pembuatan animasi guna mempermudah dan mempercepat gerak karakter yang memiliki bentuk seperti manusia[8] Proses ini biasanya memanfaatkan kamera untuk mengamati gerakan dan posisi objek dalam ruang dua dimensi (2D) yang kemudian dikonversi ke tiga dimensi (3D) menggunakan pendekatan geometris, seperti kalibrasi kamera dan triangulasi[9].

Pada penelitian ini, jarak objek dihitung berdasarkan ukuran proyeksi objek dalam gambar (*bounding box*) menggunakan formula:

$$z = \frac{W_{\text{known}} \cdot f}{W_{\text{pixel}}}$$

Dengan  $z$  adalah jarak dari kamera,  $W_{\text{known}}$  adalah lebar sebenarnya dari objek,  $f$  adalah panjang fokus kamera (*focal length*), dan  $W_{\text{pixel}}$  adalah lebar dalam citra digital. Formula ini telah diterapkan oleh berbagai penelitian untuk aplikasi robot berbasis vision[10].

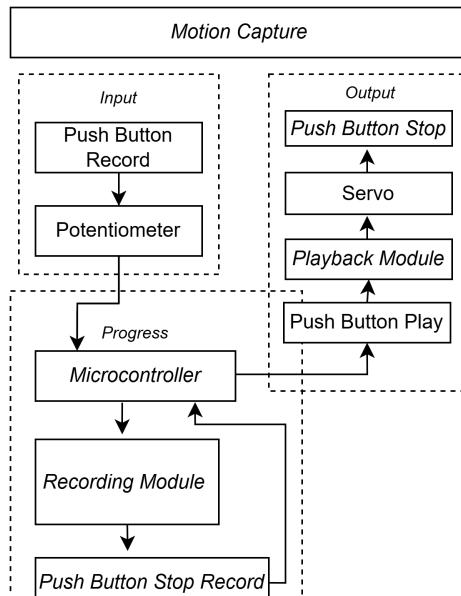
### 2.4 Objek Deteksi

Dekripsi objek (*object detection*) merupakan proses identifikasi dan klasifikasi objek tertentu dalam citra atau video. Salah satu teknik yang sering digunakan adalah dengan memanfaatkan metode *feature extraction* seperti tepi objek (*edges*) yang ditangkap melalui algoritma seperti *Canny Edge Detection*. Metode ini efektif untuk mendekripsi batas objek berbasis perbedaan intensitas pada gambar, sehingga cocok untuk mendekripsi objek berbentuk geometris seperti tabung. Metode deteksi tepi dapat diperkuat dengan analisis bentuk, seperti rasio aspek (*aspect ratio*) dan keserupaan dengan lingkaran (*circularity*). Studi oleh Gupta et al. (2021) menunjukkan bahwa deteksi objek berbasis tepi memberikan

hasil yang cukup presisi dalam lingkungan dengan pencahayaan yang baik dan kontras objek yang tinggi terhadap latar belakang (*background*). Untuk meningkatkan akurasi, sering digunakan pula segmentasi warna dalam ruang warna HSV (*Hue, Saturation, Value*) karena lebih tahan terhadap perubahan pencahayaan dibandingkan ruang warna RGB (Red, Green, Blue)[1].

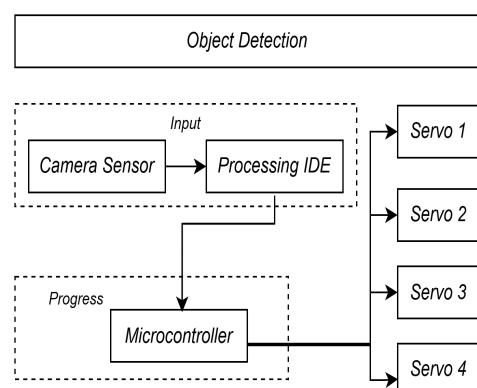
## 3 Metodologi

### 3.1 Blok Diagram



Gambar 4. Block Diagram Motion Record

Dari blok diagram diatas dengan tujuan merekam gerakan (*capture of motion*), sistem ini terdiri dari tiga komponen utama: input, progress, dan output. Di bagian input terdapat tombol perekaman *Push Button* untuk memulai perekaman dan *Potentiometer* yang berfungsi sebagai pengontrol input manual. Di bagian pergeseran, ada mikrokontroler yang berfungsi sebagai pengendali utama untuk mengolah data *input* dan mengirimkannya ke modul perekaman untuk menyimpan gerakan. *Push Button Stop Record* menghentikan perekaman, dan *Push Button Play* mengaktifkan modul pemutaran ulang pada bagian *output*, yang kemudian membaca data gerakan yang direkam dan meneruskan sinyal ke Servo sebagai penggerak. Selain itu, *Push Button Stop* memungkinkan sistem untuk menjalankan proses perekaman, penyimpanan, dan pemutaran ulang gerakan secara bersamaan.



Gambar 5. Block Diagram Object Detection

Sistem deteksi objek digambarkan dalam diagram ini. Sistem ini menggunakan platform pemrosesan untuk analisis, kamera untuk pengambilan data, dan mikrokontroler dan servo untuk melakukan pergerakan. Tujuan akhirnya adalah menggunakan data dari proses deteksi objek untuk mengontrol servo.

Pada diagram blok akan menjelaskan rancangan alur dari hardware yang dibuat yaitu:

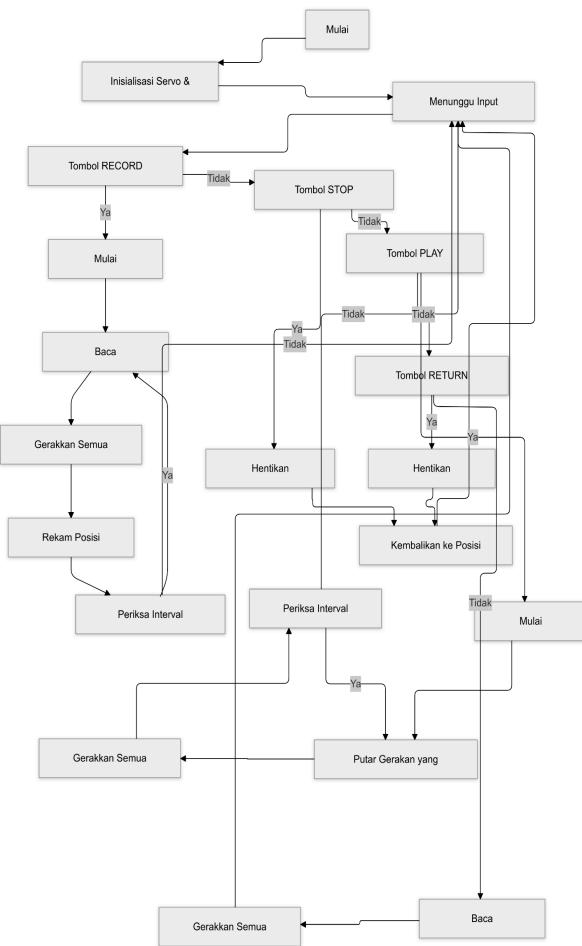
1. *Camera Sensor* berfungsi untuk melakukan analisis data visual, kamera menangkap gambar atau video dari lingkungan sekitar. Kamera ini sangat penting untuk mendeteksi objek berdasarkan fitur tertentu, seperti warna, bentuk, atau lokasi.
2. *Processing IDE (Integrated Development Environment)* berfungsi untuk mengolah data gambar menggunakan algoritma tertentu, seperti pengenalan pola atau segmentasi objek, sehingga hasil pemrosesan dapat dikirimkan ke mikrokontroler.
3. *Microcontroller* berfungsi sebagai pengendali utama, menerima instruksi dari sistem pemrosesan. Mikrokontroler mengumpulkan dan memproses data untuk menghasilkan sinyal kendali bagi servo motor.
4. Servo 1-4 berperan sebagai aktuator yang bergerak sesuai dengan hasil deteksi objek. Setiap servo memiliki tugas khusus yang dikendalikan oleh mikrokontroler berdasarkan instruksi yang diterima dari sistem pemrosesan.

Sistem ini menggunakan perekaman, pemutaran ulang gerakan, dan deteksi objek, seperti yang ditunjukkan pada blok diagram ini. Tombol perekaman dan potensiometer untuk kontrol manual adalah input. Mikrokontroler memproses data dan mengontrol modul perekaman serta servo motor. Gerakan yang direkam dikontrol oleh tombol play, stop, dan control servo dari output. Pada deteksi objek, data visual yang diproses untuk mengenali objek diambil dari kamera, dan berdasarkan hasil pemrosesannya, mikrokontroler menggerakkan servo. Semua komponen bekerja sama untuk membuat gerakan yang tepat dan responsif.

### 3.2 Flowchart Sistem

Pengembangan sistem yang menggunakan ESP32 untuk mendeteksi dan memantau objek 3D. Sistem ini dapat memproses gambar secara real-time dengan menggunakan metode pengolahan gambar seperti konversi ke ruang warna HSV, blurring, dan operasi morfologi. Ini terhubung ke kamera dan modul WiFi. Perangkat keras dihidupkan dan parameter pengolahan gambar diatur. Setelah itu, sistem menunggu pengguna mengambil gambar atau keluar dari aplikasi. Setelah objek ditemukan pada setiap frame, pembaruan posisi 3D dinamis dilakukan dengan pelacakan objek.

#### A. Flowchart Motion Capture

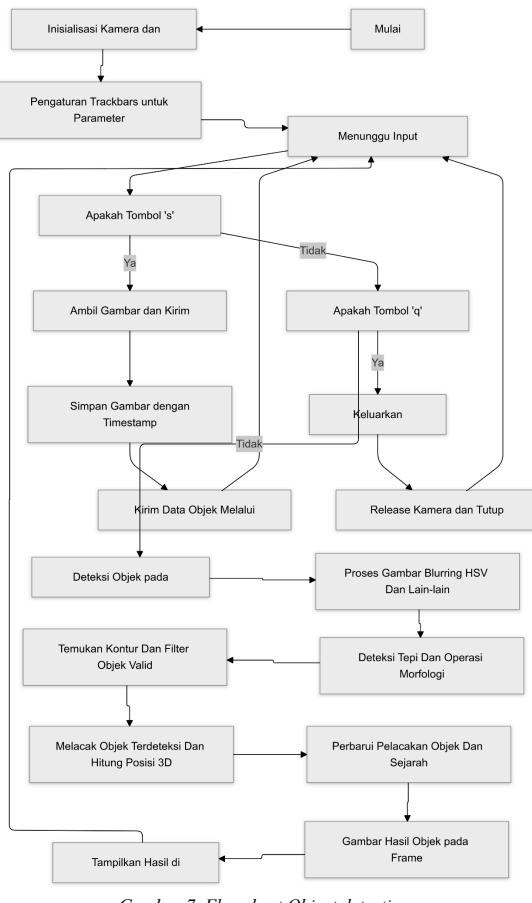


Gambar 6. Flowchart Motion Capture

Flowchart ini menggambarkan logika alur kerja untuk pengendalian servo motor menggunakan tombol dan potensiometer pada mode motion capture berbasis ESP32, Berikut adalah penjelasan langkah-langkahnya:

- Sistem dimulai dengan inisialisasi pada semua pin yang terhubung ke tombol dan servo. Servo siap menerima perintah, dan tombol siap membaca input pengguna.
- Sistem menunggu input dari tombol RECORD, STOP, PLAY, dan RETURN.
- Jika tombol RECORD ditekan, sistem mulai merekam posisi servo berdasarkan nilai potensiometer. Perekaman berlanjut hingga tombol STOP ditekan.
- Tombol STOP menghentikan perekaman dan mengembalikan servo ke posisi awal.
- Jika tombol PLAY ditekan, sistem memutar ulang posisi servo yang telah direkam sesuai interval yang ditentukan.
- Tombol RETURN menghentikan pemutaran dan mengembalikan servo ke posisi awal.
- Jika tidak ada tombol yang ditekan, sistem membaca potensiometer untuk menggerakkan servo secara manual.
- Setelah setiap proses selesai, sistem kembali menunggu input tombol berikutnya.

#### B. Flowchart Motion Capture

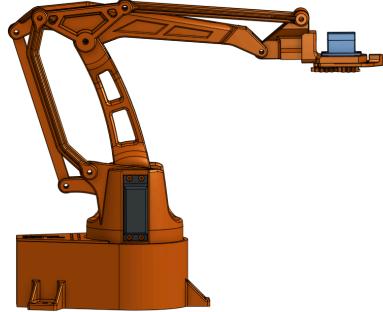


Flowchart ini menggambarkan logika alur kerja untuk Proses Sistem Deteksi dan Pelacakan Objek 3D. Proses dimulai dengan inisialisasi kamera dan WiFi (B) dan pengaturan trackbar untuk parameter pengolahan citra (C). Sistem menunggu input pengguna untuk mengambil gambar (tombol "s") dan mengirim data objek melalui WiFi (F-G-H). Jika tombol "q" ditekan, aplikasi akan keluar dan menutup WiFi dan kamera (I-J-K). Jika tidak, sistem melanjutkan untuk mendeteksi objek pada setiap frame (L) dan menggunakan HSV, blurring, dan morfologi. Sebelum kembali menunggu input pengguna untuk melanjutkan proses, posisi objek diperbarui dan hasil pelacakan ditampilkan pada layar (Q-R-S).

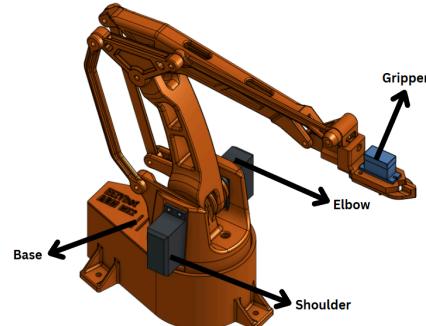
### 3.3 Perancangan Mekanik

Perancangan mekanik dari robot arm didapatkan dari situs Thingiverse, situs tersebut merupakan sebuah platform yang menyediakan berbagai desain 3D secara gratis, salah satunya adalah desain robot arm 4 DOF yang digunakan dalam penelitian. Desain yang digunakan "EAZY botRAM MK2" milik daGHIzmo (Thingiverse ID: 1454048), karya daGHIzmo dapat diakses melalui tautan <https://www.thingiverse.com/thing:1454048>. Desain ini dipublikasikan di bawah lisensi Creative Commons - Attribution - Share Alike (CC BY-SA), lisensi ini memungkinkan pengguna untuk menggunakan dan memodifikasi desain yang diberikan.

Digunakannya desain ini dilandasi dari fleksibilitasnya dalam menggunakan komponen servo (MG996R dan MG90R) yang digunakan. Selain itu setiap bagian dari robot dapat cetak secara terpisah menggunakan 3D printer, hal ini memudahkan dalam perakitan robot dan efisiensi waktu dalam mencetak robot. Desain Robot yang dapat dilihat pada Gambar 8 dan Gambar 9.



*Gambar 8. Robot arm design side view*



*Gambar 9. Robot arm design 4 DOF per degree*

Pada proyek ini, desain yang didapatkan sedikit dimodifikasi pada bagian Grippernya. Dimana Gripper mendapatkan modifikasi pada bagian panjang capitnya sebanyak 5 cm. Modifikasi dilakukan untuk mempermudah robot untuk mengangkat beban yang digunakan pada proyek ini, yaitu berupa sebuah tabung, hasil modifikasi dapat dilihat pada Gambar 10. Beban yang digunakan berupa sebuah tabung dengan ukuran Tinggi 7 cm dan Diameter 3,5 cm, dalam 2 warna yang berbeda, yaitu biru dan merah. Beban berupa sebuah tabung dapat dilihat pada gambar 11.

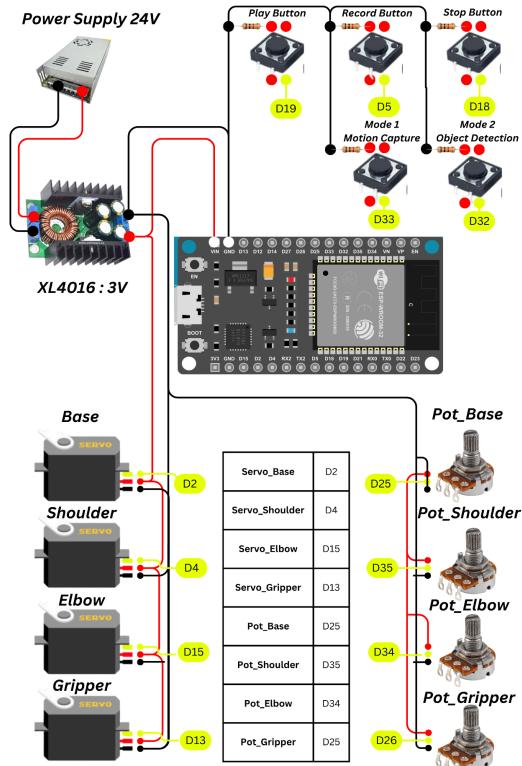


*Gambar 10. Robot arm design 4 DOF per degree*



Gambar 11. Robot arm design 4 DOF per degree

### 3.4 Perancangan Elektrikal

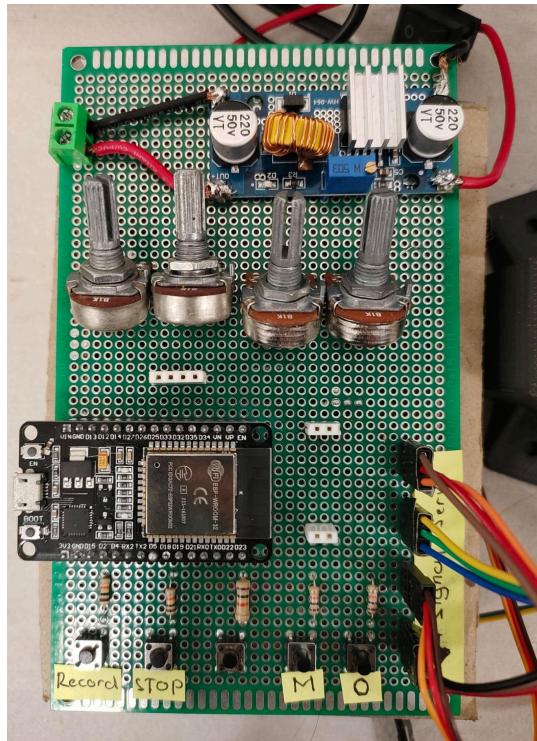


Gambar 12. Wiring Diagram

Diagram wiring dari sebuah sistem kendali berbasis ESP32 untuk menggerakkan arm robot 4-DoF (Degrees of Freedom). Sistem ini menggunakan sumber daya utama berupa Power Supply 24V yang diturunkan menjadi 3V menggunakan modul regulator XL4016 untuk memberikan daya yang stabil kepada komponen. Mikrokontroler ESP32 berfungsi sebagai pengendali utama, dengan berbagai pin digital yang digunakan untuk mengontrol servo motor dan membaca input dari potensiometer serta tombol.

Empat servo motor terhubung ke pin ESP32, masing-masing untuk menggerakkan pangkalan (Servo\_Base di D2), bahu (Servo\_Shoulder di D4), siku (Servo\_Elbow di D15), dan gripper (Servo\_Gripper di D13). Posisi masing-masing bagian arm robot dipantau menggunakan empat potensiometer yang terhubung ke pin D25, D35, D34, dan D26. Untuk input pengguna, terdapat lima tombol yang memiliki fungsi berbeda: Play Button (D19) untuk menjalankan gerakan yang telah direkam, Record Button (D5) untuk merekam gerakan manual, Stop Button (D18) untuk menghentikan gerakan, serta tombol Mode 1 (D33) dan Mode 2 (D32) untuk mengaktifkan mode penangkapan gerak (motion capture) dan pendekripsi objek (object detection).

Setiap tombol dilengkapi resistor pull-down untuk mencegah noise pada sinyal input, dan seluruh komponen berbagi ground yang sama dengan ESP32 untuk memastikan arus listrik berjalan dengan baik. Sistem ini dirancang untuk mengoperasikan arm robot secara manual melalui potensiometer maupun secara otomatis menggunakan mode playback dan deteksi objek. Desain wiring ini memastikan semua komponen terintegrasi dengan baik, mendukung fleksibilitas dan akurasi dalam pengoperasian robot.



Gambar 13. PCB Sistem Electrical

Sistem kendali arm robot 4-DoF diimplementasikan pada papan prototipe PCB berlubang, juga dikenal sebagai perfboard, seperti yang ditunjukkan pada gambar di atas. Mikrokontroler ESP32 berfungsi sebagai pusat kendali untuk sistem ini. Mikrokontroler ini terhubung ke empat potensiometer untuk mengetahui posisi sudut arm robot dan memiliki lima tombol input yang memungkinkan fungsi seperti merekam, menghentikan operasi, dan memilih mode pengambilan gerakan dan deteksi objek. Di bagian atas papan, modul regulator step-down menurunkan tegangan sumber daya eksternal menjadi tegangan operasi yang stabil untuk sistem. Semua koneksi dirancang dengan cermat menggunakan pin header dan kawat jumper untuk menghubungkan servo motor. Label yang ditempelkan secara manual pada setiap komponen memungkinkan fungsi untuk diidentifikasi selama pengujian.

### 3.5 Algoritma

Pada pengembangan sistem robotik, algoritma pengendalian perangkat keras seperti servo motor, sensor, dan komunikasi antar perangkat memegang peranan penting dalam memastikan kinerja sistem yang efektif dan akurat. Dalam proyek ini, algoritma yang diterapkan pada pengendalian robot berbasis mikrokontroler ESP32 melibatkan berbagai teknik matematis dan komputasional yang dirancang untuk mengelola gerakan robot secara presisi, pengambilan dan pemutaran data posisi, serta interaksi dengan pengguna melalui antarmuka tombol dan komunikasi nirkabel. Terdapat 2 metode yang digunakan sebagai fitur yaitu motion capture dan object detection.

#### 3.5.1 Motion Capture.

Beberapa teknik utama yang digunakan pada metode *motion capture* mencakup interpolasi linier untuk kontrol servo dan teknik sampling.

#### Pengendalian Servo Motor (Interpolasi Gerakan)

Salah satu aspek utama dari pengendalian robot ini adalah gerakan servo. Gerakan servo dilakukan dengan menggunakan interpolasi linier, yang memastikan bahwa servo bergerak dengan mulus dari satu posisi ke posisi lainnya. Persamaan untuk interpolasi linear dimana posisi servo pada waktu dinyatakan dengan rumus :

$$\theta(t) = \theta_0 + (\theta_f - \theta_0) \times \frac{t}{T}$$

$\theta(t)$  adalah posisi servo pada waktu  $t$

$\theta_0$  adalah posisi awal servo

$\theta_f$  adalah posisi akhir servo

$T$  adalah waktu total untuk mencapai posisi akhir

$t$  adalah waktu saat ini dalam interval  $[0, T]$

Persamaan ini digunakan pada program untuk membuat gerak dari servo lebih *smooth* atau lebih halus secara bertahap dari posisi saat ini menuju posisi target dengan interval waktu tertentu[11]

#### Perekaman dan Pemutaran posisi (Sampling dan Playback)

Dalam fungsi rekaman posisi pada fitur motion capture, data posisi servo setiap interval waktu dicatat, kemudian diputar kembali sesuai dengan urutan yang telah direkam. Ini adalah bentuk dari *sampling*, di mana posisi servo dicatat pada interval waktu tertentu. sampling posisi servo dilakukan dengan interval waktu tetap, yaitu setiap  $\Delta t$  dan posisi servo pada waktu  $t_k$  dicatat sebagai  $\theta_k$  :

$$\theta(t_k) = \theta_k$$

$\theta_k$  adalah posisi servo yang dicatat pada waktu  $t_k$

$$t_k = k \cdot \Delta t$$

$k$  adalah indeks langkah sempel,

$\Delta t$  adalah interval waktu tetap antar sampel

Saat memutar kembali posisi yang sudah direkam, posisi servo akan diputar berdasarkan indeks sampel yang tersimpan, algoritma ini hampir sama dengan *discrete-time systems*, di mana gerakan servo dicatat dan diputar dalam bentuk sampel diskrit[12].

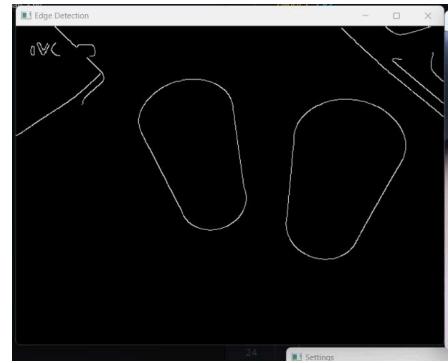
#### 3.5.2 Object Detection

Dalam implementasinya proses ini akan menggunakan kamera. Dimana kamera akan menjadi sensor utama dalam proses deteksi objek. Algoritma yang digunakan pada proses ini melibatkan berbagai teknik matematis dan komputasional yang dirancang untuk mendeteksi objek.

#### Deteksi Citra dan Pengolahan Canny Edge Detection

Deteksi tepi (edge detection) adalah teknik dasar dalam pengolahan citra yang digunakan untuk mengidentifikasi batas objek dalam gambar. Dalam proyek ini, digunakan algoritma Canny Edge Detection, yang terkenal karena kemampuannya mendeteksi tepi dengan akurat meskipun terdapat gangguan pada gambar. Canny edge detection bekerja melalui beberapa tahapan, termasuk perataan citra, deteksi gradien, pengurangan non-maximum, dan penerapan threshold ganda untuk menemukan tepi yang lebih kuat (Canny, 1986)[13].

Pada projek ini edge detection menjadi dasar utama untuk mendeteksi objek.



Gambar 14. Edge Detection

Untuk meningkatkan akurasi, ditambahkan sebuah algoritma menggunakan masker selektif yang hanya mempertahankan tepi objek yang valid sesuai dengan warna dan posisi tertentu. Tepi-tepi yang terdeteksi ini menjadi dasar untuk proses deteksi objek, pelacakan posisi, dan penghitungan koordinat 3D dalam ruang. Hasil akhir dari proses edge detection adalah citra dengan tepi-tepi objek yang jelas dan siap digunakan untuk pelacakan serta pengendalian perangkat keras yang terhubung dengan sistem.

#### Pendeteksian Kontur dan Morfologi Operasi

Setelah deteksi tepi dilakukan, kontur objek digunakan untuk mengidentifikasi bentuk dan ukuran objek dalam citra. Operasi morfologi seperti dilasi dan penutupan (closing) digunakan untuk memperbaiki kualitas deteksi tepi dengan menghilangkan noise dan memperbesar objek yang terlalu kecil. Teknik ini penting untuk meningkatkan presisi dalam mendeteksi objek dan memungkinkan deteksi objek yang lebih stabil meskipun kondisi pencahayaan dan sudut kamera berubah.

#### Pelacakan Objek (Object Tracking)

Dalam program ini, digunakan teknik pelacakan berbasis Tracking ID, di mana setiap objek diberi ID unik yang memungkinkan pemantauan secara real-time. Algoritma ini menggunakan jarak Euclidean untuk mencocokkan posisi objek yang terdeteksi dengan objek yang sudah dilacak sebelumnya. Jarak Euclidean adalah ukuran antara dua titik dalam ruang dua dimensi atau tiga dimensi[14].

Persamaan jarak Euclidean dalam ruang dua dimensi adalah sebagai berikut :

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$d$  adalah jarak antara dua titik

$x_1, y_1$  adalah koordinat pertama

$x_2, y_2$  adalah koordinat kedua

Dalam konteks pelacakan objek, jarak ini digunakan untuk menentukan kesamaan posisi antara objek yang terdeteksi pada frame saat ini dan objek yang dilacak pada frame sebelumnya. Dalam sistem ini, jika suatu objek tidak terdeteksi pada frame berikutnya, maka diasumsikan objek tersebut menghilang sementara. Jika jumlah frame berturut-turut yang tidak mendeteksi objek melebihi ambang batas tertentu ( $T_{hilang}$ ), objek akan dianggap keluar dari frame. Strategi ini dapat diformulasikan sebagai berikut :

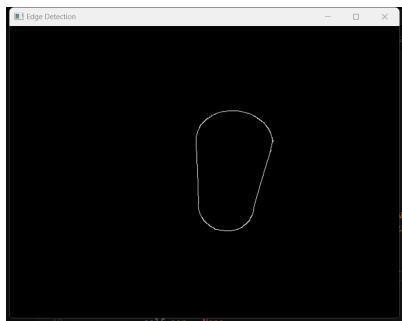
$$disappearedcount_i \geq T_{hilang} \Rightarrow \text{hapus\_Objek}$$

Pendekatan ini memastikan bahwa sistem pelacakan tetap robust meskipun ada frame dengan objek yang sulit dideteksi, seperti karena obstruksi atau pencahayaan yang buruk. Kombinasi algoritma pelacakan berbasis jarak Euclidean dan mekanisme toleransi kehilangan ini menjadikan sistem pelacakan lebih efisien untuk aplikasi real-time.



Gambar 15. Deteksi garis tepi pada objek

Pada gambar asli terdapat beberapa benda pada *background* objek, algoritma ini memungkinkan untuk menghilangkan benda benda yang terdeteksi di belakang objek dan menyisakan objek yang akan dideteksi. Hal ini memungkinkan komputasi jauh lebih ringan.



Gambar 16. Menghilangkan garis tepi selain objek

#### Perhitungan Posisi 3D dan Pengenalan Kedalaman

Untuk mendeteksi posisi objek dalam ruang tiga dimensi, algoritma ini menggunakan pengukuran jarak berdasarkan informasi lebar objek yang diketahui dan panjang fokus kamera. Posisi objek dihitung menggunakan prinsip geometri dasar yang menghubungkan ukuran objek di citra dengan jarak relatif objek terhadap kamera.

#### Komunikasi JSON dan Deserialisasi

Untuk komunikasi antar perangkat, program ini menggunakan format JSON yang dikirim melalui Wi-Fi dan diproses oleh mikrokontroler. Proses ini melibatkan deserialisasi data untuk mengekstrak informasi seperti warna objek yang akan diproses oleh robot. Deserialisasi JSON pada sisi penerima dilakukan dengan menyusun objek JSON menjadi struktur data yang dapat diproses oleh sistem[4]. Secara matematis, deserialisasi dapat dianggap sebagai transformasi fungsi dari data terstruktur (JSON) menjadi data yang dapat dipahami oleh program :

$$\text{JSON} \xrightarrow{\text{deserialize}} \text{Object Data Structure}$$

Setelah data dikirimkan dari perangkat menuju esp selanjutnya data akan diproses untuk menggerakkan servo sesuai dengan data yang didapatkan.

#### Algoritma Pick and Place dengan Posisi Terpredefini

Pada fitur Pick and Place, posisi objek yang akan diambil dan diletakkan bergantung pada warna objek. Setiap posisi tersebut didefinisikan dengan variabel  $\theta_{base}$ ,  $\theta_{shoulder}$ ,  $\theta_{elbow}$ ,  $\theta_{gripper}$ , yang masing-masing menunjukkan sudut servo untuk komponen tertentu.

Posisi Servo dapat diformulasikan sebagai berikut :

$$\begin{aligned}\theta_{base} &= F_{base}(\text{color}) \\ \theta_{shoulder} &= F_{shoulder}(\text{color}) \\ \theta_{elbow} &= F_{elbow}(\text{color})\end{aligned}$$

$$\theta_{gripper} = F_{gripper}(\text{color})$$

Dimana  $F_{component}(\text{color})$  adalah fungsi posisi servo untuk komponen tertentu berdasarkan warna objek yang terdeteksi yang dapat berupa nilai tetap atau variabel berdasarkan kondisi.



Gambar 17. Deteksi Warna

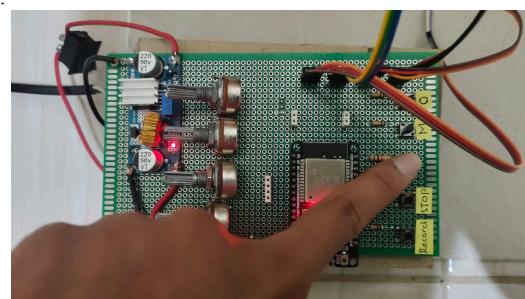
Hasil deteksi color pada objek akan dikirimkan menggunakan komunikasi wifi berupa data JSON agar mudah diproses oleh mikrokontroler.

## 4 Pembahasan

Hasil pengujian dilakukan terhadap setiap fitur yang dihadirkan pada Arm robot 4 DOF. Setiap fitur yang dihadirkan yaitu Motion Capture dan Objek Detection akan dilihat dari seberapa baik fitur tersebut dapat bekerja (Terutama pada Motion Capture), selain itu juga dilihat dari waktu yang dibutuhkan, tingkat keberhasilan, dan berbagai posisi yang dapat dijangkau oleh robot.

### 4.1 Motion Capture

Pengujian fitur ini dilakukan dengan merekam suatu rangkaian gerakan dari robot lalu mengulangi rangkaian gerakan tersebut apakah serangkaian gerakan hasil dari rekaman akan sama dengan gerakan yang sebenarnya. Untuk dapat membuktikannya, pengujian dilakukan dengan Robot Arm akan mengambil objek berupa tabung dari sebelah kiri robot lalu meletakkannya pada sisi lain robot (bagian kanan). Total pergerakan pada pengujian ini adalah 6 langkah, yaitu diawali dengan menekan tombol start play, lalu robot akan bergerak kekiri, setelahnya gripper akan mengambil benda, selanjutnya robot bergerak kearah kanan, dan melepaskan benda dari gripper, terakhir robot akan kembali ke posisi awal (tengah). Hasil pengujian dapat dilihat pada Gambar XX hingga Gambar XX.



Gambar 18. Tombol Start pada PCB ditekan



Gambar 19. gripper membuka pada posisi awal arm robot yaitu ditengah



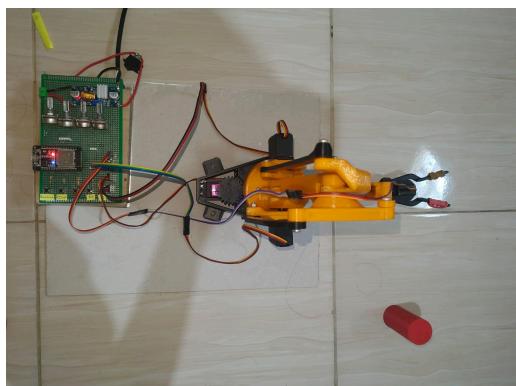
Gambar 20. Robot arm mengambil benda pada bagian kiri robot arm



Gambar 21. Robot memindahkan benda menjadi ke bagian kanan robot arm



Gambar 22. Gripper melepaskan benda

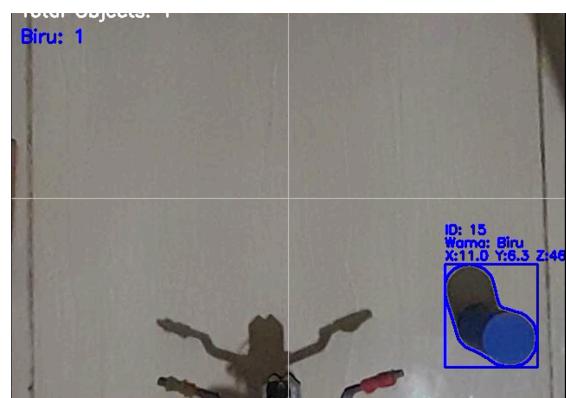


Gambar 23. Robot arm kembali pada posisi awal

Pengujian dilakukan sebanyak 3 kali untuk fitur Motion capture ini, dan ketiganya menunjukkan hasil yang sama. Ketepatan robot arm dalam mengambil benda sangat ditentukan dari letak dari benda tersebut. Setiap percobaan yang dilakukan, benda harus pada lokasi yang sama pada saat perekaman,

#### 4.2 Objek Detection

Pada pengujian tahap ini, memerlukan beberapa tahap. Tahap pertama ada membaca objek menggunakan kamera, kemudian setelah objek terdeteksi adalah mengirimkan informasi objek menuju mikrokontroler. Setelah data diterima mikrokontroler selanjutnya adalah mikrokontroler menggerakkan servo. Pada tahap ini robot akan melakukan gerak *pick and place*. Dimana pada saat melakukan *pick* robot akan menuju ke titik dimana objek terdeteksi, kemudian akan membawa objek kemudian melakukan *place* atau menaruh barang pada titik yang sudah ditentukan berdasarkan warna. Pada tahap ini dilakukan pengujian sebanyak 6 kali yaitu 3 kali objek berwarna biru dan 3 kali objek berwarna merah. Tujuan pengujian ini adalah untuk melihat fungsi kontrol invers kinematic yang ada pada robot apakah sudah berjalan dengan baik atau belum.



Gambar 24. Posisi Objek pada pengujian pertama

```
Sending data and capturing image...
Image saved: captured_20241214_195404.png
Data sent: {"15": {"x": 11.0, "y": 6.4, "z": 46.7, "color": "Biru"}}
```

Gambar 25. pengiriman data

Robot akan bergerak sesuai dengan titik koordinat x,y, dan z yang kemudian akan diolah menggunakan *invers kinematic*. Dimana nilai koordinat x yang dibaca akan di mapping menggunakan fungsi *map* yang ada pada arduino. Ketika nilai x bernilai positif maka akan servo pada bagian base akan bergerak dengan besaran sudut 90-180 derajat. Sedang kan ketika koordinat x bernilai negatif maka servo akan bergerak dengan besaran sudut 90-0 derajat. Agar dapat mementukan gerak dari elbow dan juga shoulders diperlukan pengolahan menggunakan *inverse kinematic* dengan persamaan sebagai berikut :

$$I = \sqrt{x^2 + y^2}$$

$$h = \sqrt{I^2 + z^2}$$

$$\text{Sudut } \phi = \arctan\left(\frac{z}{I}\right) \cdot \frac{60}{\pi}$$

$$\text{Sudut } \theta = \arccos\left(\frac{h}{2 \times 40}\right) \cdot \frac{60}{\pi}$$

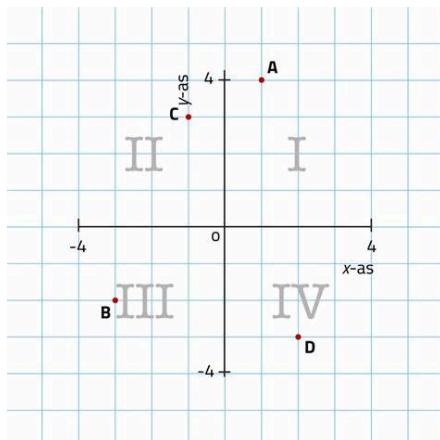
$$\text{ShoulderAngle} = \theta + \phi + 10$$

$$\text{EllbowAngle} = \phi - \theta - 5$$

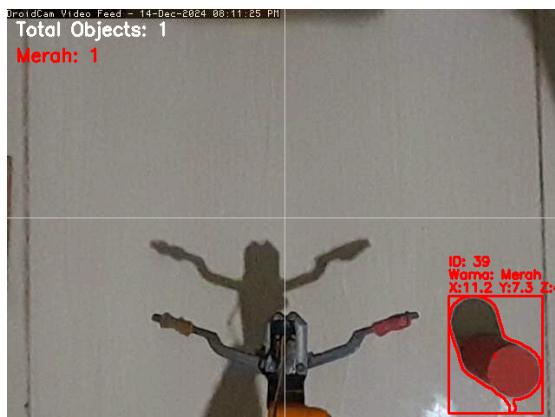
I menggabungkan x dan y untuk menentukan sejauh mana target berada dari asal pada bidang horizontal. Kemudian h adalah panjang total yang menghubungkan asal dengan titik target dalam ruang 3D. Selanjutnya Sudut  $\phi$  adalah sudut

elevasi antara bidang horizontal (XY) dan vektor yang mengarah ke titik target (x,y,z). Fungsinya adalah untuk mengontrol sudut awal lengan untuk menjangkau target di ketinggian tertentu. Sudut  $\theta$  mewakili rotasi yang diperlukan untuk mengatur jangkauan lengan dengan mempertimbangkan panjang total  $h$ . Dalam hal ini  $2 \cdot 40$  adalah batas panjang maksimum lengan yang diasumsikan dari dua segmen masing-masing panjang shoulder dan elbow adalah 40 cm. Gerak pada bagian shoulder dihasilkan dari kombinasi  $\theta$  dan  $\phi$ , ditambah offset tetap 10 derajat. Offset ini bertujuan untuk mengimbangi perbedaan mekanik atau fisik. Selanjutnya adalah gerak elbow dihasilkan dari pengurangan  $\theta$  dari  $\phi$ , dikurangi dengan offset tetap 5 derajat, tujuannya untuk menyesuaikan gerak alami siku.

Terdapat batasan dimana robot arm tidak mampu untuk mencapai objek. Jika dalam koordinat robot dapat mengambil objek yang berada pada kuadran 3 dan 4.



Gambar 26. Ilustrasi Fungsi kuadran



Gambar 27. Posisi objek pada kuadran 4



Gambar 28. Posisi objek pada kuadran 3

Dalam pengujian ini nilai koordinat objek yang terbaca adalah  $x = 11$ ,  $y = 6.4$ , dan  $z = 46.7$  maka gerak servo yang dihasilkan jika dikalkulasikan menggunakan persamaan diatas didapatkan gerak servo sebesar :

$$\begin{aligned} \text{ShoulderAngle} &= 52.51 \text{ derajat} \\ \text{ElbowAngle} &= 2.33 \text{ derajat} \\ \text{BaseAngle} &= 110 \text{ derajat} \end{aligned}$$

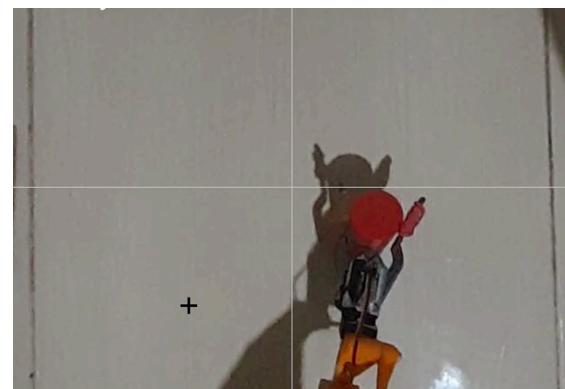


Gambar 29. Mengambil objek sesuai dengan hasil invers kinematic

Setelah melakukan gerak *pick* selanjutnya pada gambar 24 robot akan bergerak menuju titik koordinat sesuai dengan warna yang dideteksi oleh kamera. Ketika objek berwarna biru maka putaran servo base akan sebesar 60 derajat. Sedangkan pada warna merah servo akan bergerak sebesar 120 derajat.



Gambar 30. Robot bergerak menuju titik menaruh objek berwarna biru



Gambar 31. Robot bergerak menuju titik menaruh objek berwarna merah

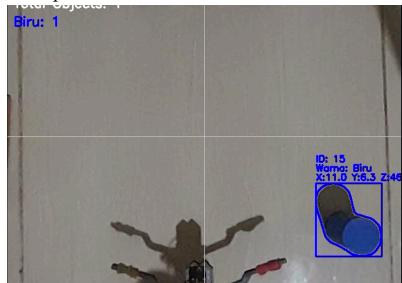
Dalam tahap ini pengujian robot dilakukan sebanyak 6 kali agar dapat melihat kinerja dari *invers kinematic control*.

Tabel

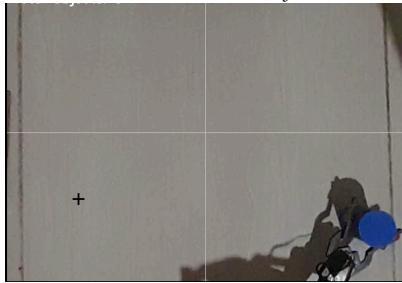
Uji coba	Warna	Keberhasilan	Waktu
Pertama	Biru	Berhasil	11s
Kedua	Biru	Gagal	11s
Ketiga	Biru	Berhasil	10s
Keempat	Merah	Berhasil	13s
Kelima	Merah	Berhasil	11s
Keenam	Merah	Berhasil	12s

#### 4.2.1 Draf Pengujian

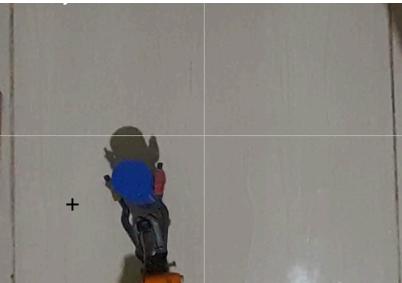
##### 1. Uji coba pertama



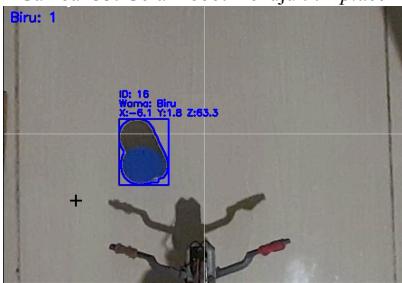
Gambar 31. Posisi Objek



Gambar 32. Gerak Pick

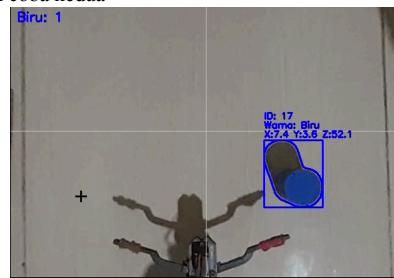


Gambar 33. Gerak robot menuju titik place

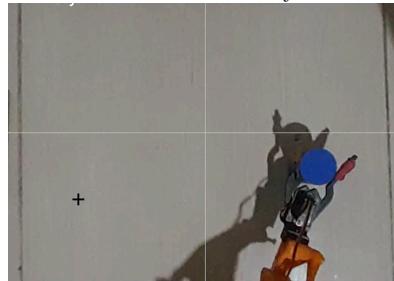


Gambar 34. Gerak place

##### 2. Uji coba kedua



Gambar 35. Posisi Objek



Gambar 36. Gerak Pick



Gambar 37. Gerak robot menuju titik place



Gambar 38. Gerak place

##### 3. Uji coba ketiga



Gambar 39. Posisi Objek



Gambar 40. Gerak Pick



Gambar 37. Gerak robot menuju titik *place*

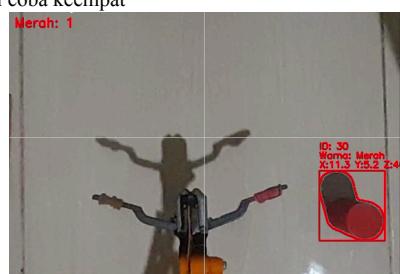


Gambar 41. Gerak robot menuju titik *place*



Gambar 38. Gerak *place*

4. Uji coba keempat

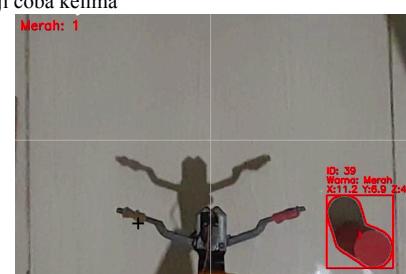


Gambar 35. Posisi Objek



Gambar 36. Gerak Pick

5. Uji coba kelima



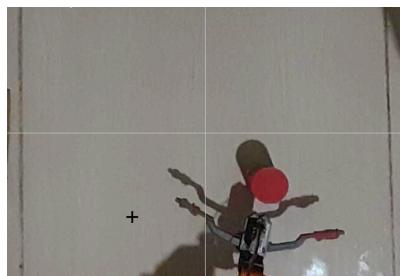
Gambar 35. Posisi Objek



Gambar 36. Gerak Pick



Gambar 37. Gerak robot menuju titik *place*



Gambar 38. Gerak place

## 6. Uji coba keenam



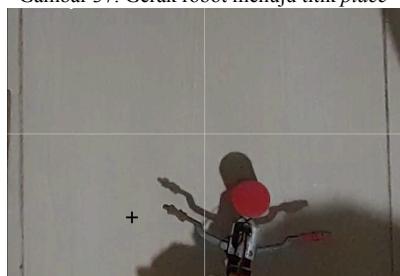
Gambar 35. Posisi Objek



Gambar 36. Gerak Pick



Gambar 37. Gerak robot menuju titik place



Gambar 38. Gerak place

Dari 6 kali percobaan terdapat 1 kali gagal pada saat uji coba, kegagalan tersebut dapat diakibatkan oleh hasil perhitungan yang meleset dari titik koordinat atau terdapat noise pada sinyal servo yang terhubung dengan sinyal esp32. Waktu yang dibutuhkan untuk melakukan gerak *pick and place* relatif konsisten.

## 5 Kesimpulan

Tujuan dari proyek pengembangan sistem kendali arm robot 4-DoF berbasis ESP32 dengan mode perekaman gerakan (*motion capture*) dan deteksi objek (*object detection*) telah tercapai. dengan menggabungkan berbagai bagian, telah dihasilkan kendali gerakan yang responsif dan tepat Dengan menggunakan algoritma sampling dan interpolasi linier untuk menghasilkan pergerakan servo yang halus, fitur pengambilan gerakan memungkinkan robot untuk merekam dan memutar ulang gerakan dengan tingkat akurasi yang tinggi. Untuk fitur deteksi objek, algoritma Canny Edge Detection yang menggunakan teknik operasi morfologi dan kamera berhasil meningkatkan akurasi deteksi objek, meskipun masalah seperti pencahayaan dan sudut kamera masih mempengaruhi kinerjanya. Dengan menggunakan inverse kinematics untuk mengontrol gerakan robot, objek dapat dipindahkan sesuai dengan warna dengan tingkat keberhasilan yang tinggi dalam pengujian pick-and-place.

Kemampuan robot untuk mengangkat beban dan fleksibilitas telah ditingkatkan melalui desain mekanik arm robot yang menggunakan "EAZY botRAM MK2" dari Thingiverse, bersama dengan modifikasi pada gripper. Baik pengoperasian manual maupun otomatis didukung oleh sistem elektrikal dan wiring yang baik. Secara keseluruhan, sistem ini berhasil mencapai tujuan utama proyek dengan membuat robot arm yang dapat merekam gerakan dan menangani objek secara mandiri. Tetapi masih ada ruang untuk peningkatan pada deteksi objek dan akurasi koordinat, yang dapat meningkatkan kinerja sistem secara keseluruhan.

## Kontribusi Penulis

Penulis 1, Penulis 2 Penulis 3 bertanggung jawab atas pengembangan konsep utama dari proyek ini. Penulis 1, Penulis 2, dan Penulis 3 berkontribusi dalam merancang metodologi penelitian. Penulis 2 bertanggung jawab atas pengembangan perangkat lunak, sementara Penulis 1 dan Penulis 3 bertanggung jawab atas pengembangan perangkat keras yang digunakan dalam proyek ini. Validasi sistem dilakukan oleh Penulis 1, Penulis 2, dan Penulis 3, sedangkan Penulis 1 melakukan analisis formal terhadap data yang diperoleh. Penulis 1, Penulis 2, Penulis 3 , dan Penulis 4 menyediakan sumber daya dan melakukan kurasi data untuk memastikan kualitas dan integritas data. Penulisan draf awal dilakukan oleh Penulis 1, Penulis 2, Penulis 3 , dan Penulis 4. Peninjauan dan pengeditan draf dilakukan oleh Penulis 1 dan Penulis 3. Penulis 5 memberikan pengawasan umum terhadap proyek dan mendukung administrasi proyek yang dikelola oleh Penulis 1, Penulis 2, dan Penulis 3. Pendanaan untuk proyek ini disediakan oleh Penulis 5.

## Ucapan Terima Kasih

Kami mengucapkan terima kasih kepada Universitas Negeri Yogyakarta, yang telah memberikan bimbingan dan dukungan selama dilaksanakannya penelitian ini. Kami juga ingin mengucapkan terimakasih terhadap tim dari Jurnal *El sains* yang telah memberikan kami kesempatan untuk melakukan *submit* pada jurnal kami. Ucapan terima kasih juga kami tujuhan kepada kreator Thingiverse, khususnya pada pengguna daGHIIZmo, atas desain mekanik robot kami "EAZY botRAM MK2", yang kami gunakan sebagai referensi utama pada robot arm 4 DOF yang kami gunakan.

## Referensi

- [1] G. Sharma, A. Gupta, dan R. Malik, "Pengenalan Objek Berbasis Bentuk dalam Gambar: Tinjauan," *Jurnal Internasional Aplikasi Komputer*, vol. 58, no. 21, hlm. 8–11, November 2012, doi: <https://doi.org/10.5120/9405-3684>.
- [2] Zhang, L., & Li, H.. Geometric Transformations for Depth Estimation in Computer Vision. *IEEE Transactions on Robotics*, 35(2), 145–153. doi:10.1109/TRO.2018.2876105.
- [3] Muslimin, S. Analisis Pulse Motor Servo Sebagai Penggerak Utama Lengan Robot Berjari Berbasis

- Mikrokontroler. *Jurnal PROTON*, 2018, pp. 1–4. doi: <https://doi.org/10.31328/jp.v10i1.800>.
- [4] Prototipe Modul Praktik untuk pengembangan aplikasi internet of things (IOT), [https://www.researchgate.net/publication/350068280\\_Protopipe\\_Modul\\_Praktik\\_untuk\\_Pengembangan\\_Aplikasi\\_Internet\\_of\\_Things\\_IoT](https://www.researchgate.net/publication/350068280_Protopipe_Modul_Praktik_untuk_Pengembangan_Aplikasi_Internet_of_Things_IoT).
- [5] S. Muslimin, ‘ANALISIS PULSE MOTOR SERVO SEBAGAI PENGGERAK UTAMA LENGAN ROBOT BERJARI BERBASIS MIKROKONTROLER’, *Jurnal PROTON*, 2018, pp. 1–4, doi: <https://doi.org/10.31328/jp.v10i1.800>.
- [6] A. Hilal, and S. Manan, “PEMANFAATAN MOTOR SERVO SEBAGAI PENGGERAK CCTV UNTUK MELIHAT ALAT-ALAT MONITOR DAN KONDISI PASIEN DI RUANG ICU,” *Gema Teknologi*, vol. 17, no. 2, Aug. 2015. <https://doi.org/10.14710/gt.v17i2.8924>.
- [7] Rusdiyanto, Ibrahim, I. A. Bangsa, ‘Implementasi Motor Servo MG996r Sebagai Robot Pemegang Batang Nosel Pada Sprayer Elektrik Berbasis Arduino Mega2560’, *JURNAL ILMIAH ELEKTRONIKA DAN KOMPUTER*, 2021, Vol. 14, No. 1, pp. 162–170, doi: <https://doi.org/10.51903/elkom.v14i1.443>.
- [8] L. S. Utami, \*Muhammad Nasir, and A. Anwar, “Teknik motion capture Dalam Proses pergerakan Karakter Manusia menggunakan kinect,” *Jurnal Teknologi Rekayasa Informasi dan Komputer*, <https://e-jurnal.pnl.ac.id/TRIK/article/view/2568/2141>.
- [9] Miller, A., & Johnson, T. (2020). *Motion Capture Techniques for 3D Object Tracking*. *Journal of Robotics and Automation*, 12(4), 56–63.
- [10] Gupta, R., & Sharma, P. (2021). *Edge-based Object Detection in Computer Vision*. *International Journal of Image Processing*, 14(3), 25–30.
- [11] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Neural Networks*, vol. 14, no. 4, pp. 727–736, 2001.
- [12] O. Khatib, M. Kaneko, and S. Miyamoto, "Control of robot motion using sampled trajectories," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 1, pp. 67–74, 1999.
- [13] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.
- [14] M. Yassir, O. Cherkaoui, and S. Boudjadar, "JSON-based communication protocol for IoT," *International Journal of Computer Applications*, vol. 94, no. 12, pp. 11–15, 2014.