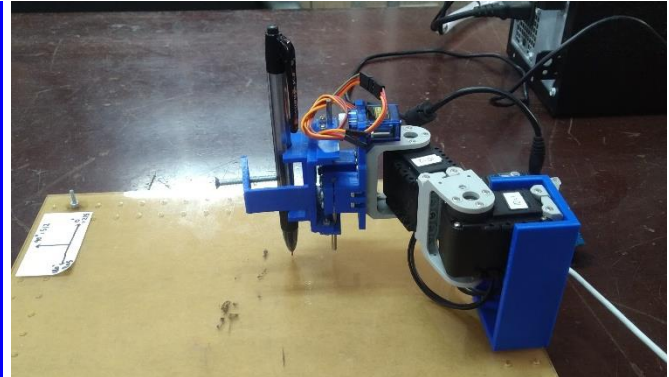
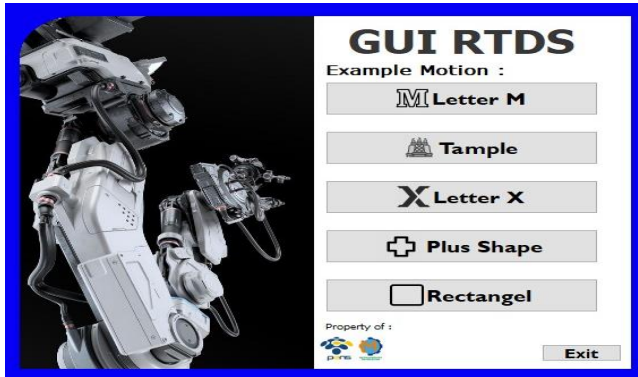


Final System and Team Personnel



Dalam proyek pengembangan Robot Tangan Dua Sendi (RTDS) dan antarmuka pengguna grafis (GUI), kami tim teknisi memulai dengan melakukan analisis awal guna menetapkan tujuan dan kebutuhan perangkat keras serta perangkat lunak yang diperlukan. Kami merancang program robot menggunakan Software Robotics, memilih Software VSCode untuk mengembangkan GUI, dan mengkode GUI menggunakan bahasa pemrograman Python. Setelah menghubungkan perangkat keras dan perangkat lunak, kami melakukan pengujian dan debugging untuk meningkatkan kinerja robot dan antarmuka pengendali grafis (GUI). Akhirnya, kami berhasil mencapai solusi desain akhir yang memungkinkan pengendalian sukses atas Robot Tangan Dua Sendi (RTDS) melalui GUI, dengan dokumentasi yang mencakup seluruh langkah yang diperlukan untuk membangun sistem ini.

SISTEM PEMROGRAMAN MEKATRONIKA

Team Robot Tangan Dua Sendi (RTDS)

Abstract

Proyek ini bertujuan untuk mengintegrasikan Robot Tangan Dua Sendi dengan antarmuka pengendali grafis (GUI) yang sangat intuitif. Integrasi ini tidak hanya memungkinkan pengguna mengendalikan gerakan robot dengan mudah, tetapi juga memfasilitasi pembuatan gambar objek melalui antarmuka pengendali grafis (GUI), tanpa memerlukan pengetahuan mendalam tentang pemrograman robot. Hasil dari proyek ini diharapkan dapat memberikan kontribusi signifikan dalam meningkatkan aksesibilitas dan kegunaan Robot Tangan Dua Sendi dalam berbagai konteks, seperti produksi otomatis dan penelitian laboratorium.

yogii

yogidwiprasetyo@me.student.pens.ac.id
mrezapratama@me.student.pens.ac.id



Table of Contents

1 Introduction and Initial Analysis	3
1.1 Project Context	3
1.2 Initial Thought Process.....	3
2 Requirement Analysis and Specification.....	3
2.1 User Requirements	4
2.2 System Requirements	4
2.3 Tools and Technologies.....	4
3 Conceptual Design	6
3.1 System Architecture.....	6
3.2 Interface Design	6
3.3 Control Algorithm Design.....	6
4 Detailed Design and Development	8
4.1 Component Design.....	8
4.2 Coding and Implementation	9
4.3 Integration	9
4.4 Unique Features.....	10
5 Testing, Evaluation, and Optimization	11
5.1 Testing Strategy	11
5.2 Performance Evaluation.....	11
5.3 Optimization	12
6 Collaboration and Project Management.....	14
6.1 Teamwork Dynamics.....	14
6.2 Project Management	16
7 Conclusion and Reflection.....	17
7.1 Project Summary.....	17
7.2 Future Work.....	17
7.3 Personal and Group Reflections.....	18
8 Appendices.....	18
8.1 Bill of Materials	18
8.2 Electrical Wiring and System Layout.....	19
8.3 Code Repository.....	21



8.4 Additional Documentation.....	30
9 References	31



1 Introduction and Initial Analysis

1.1 Project Context

Robot tangan dua sendi adalah evolusi dari konsep robot lengan satu sendi, yang sebenarnya terdiri dari dua robot tangan satu sendi yang digabungkan. Karakteristik utama dari robot tangan dua sendi adalah kemampuannya untuk bergerak langsung ke suatu titik dalam koordinat kartesian (dalam sumbu X dan Y). Robot tangan ini menjadi relevan dalam berbagai aplikasi seperti manufaktur, medis, eksplorasi luar angkasa, dan otomasi industri, terus berkembang dengan peningkatan kecepatan, keakuratan, dan adaptabilitas dalam perkembangan robotika dan otomasi.

Graphical user interface (GUI) adalah cara pengguna berkomunikasi dengan perangkat mereka dan telah menjadi standar utama dalam pengembangan teknologi saat ini. GUI adalah sistem komponen yang dikembangkan untuk beragam perangkat komputer yang berfokus pada interaksi visual. Dengan kehadiran GUI, pengguna dapat dengan mudah melihat apa yang telah mereka masukkan dan menerima umpan balik visual yang mencolok, seperti perubahan warna, tampilan, ukuran, dan lainnya. Penemuan teknologi ini telah membuktikan dirinya sebagai solusi yang sangat efektif, terutama bagi pengguna yang bukan pengembang perangkat lunak, karena mempermudah penggunaan perangkat teknologi sehari-hari.

1.1.1 Rumusan Masalah

1. Bagaimana Robot tangan dua sendi dapat dikontrol melalui GUI
2. Bagaimana Robot tangan dua sendi dapat menampilkan gerakan yang diatur dengan Graphical User Interface (GUI)
3. Bagaimana servo Robot Tangan Dua Sendi dapat digerakkan menggunakan Graphical User Interface (GUI)

1.1.2 Tujuan

Tujuan dari proyek ini adalah:

1. Untuk mengetahui cara pembuatan Graphical User Interface (GUI)
2. Untuk mengontrol robot tangan dua sendi melalui GUI

1.2 Initial Thought Process

Pada awal proses proyek ini yang kami lakukan pertama kali adalah brainstorming dengan mencari sumber-sumber referensi di internet terkait dengan robot tangan dua sendi, yang dilanjutkan dengan menginstall aplikasi apa saja yang dibutuhkan dalam proses kedepannya nanti. Pertama kami mencoba set up servo yang ada pada robot tangan dua sendi menggunakan software Open CM, setelah itu kami mencoba untuk membuat gerakan bebas dan menarik pada robot. Output dari proyek kali ini diharapkan nanti robot dapat bergerak sesuai program yang akan disambungkan dengan GUI dan ditampilkan pada layar monitor.

Adapun tantangan dalam proyek ini ialah:

1. Jumlah anggota tim yang sedikit
2. Baru pertama kali membuat Graphical User Interface (GUI)
3. Belajar bahasa pemrograman baru



Peluang dalam proyek ini ialah:

1. Mengetahui cara memprogram robot tangan dua sendi
2. Mengetahui cara pembuatan tampilan UI dengan software QT Designer
3. Mendapat pengetahuan baru pembuatan GUI

2 Requirement Analysis and Specification

2.1 User Requirements

Pengguna harus memiliki Robot Tangan Dua Sendi (RTDS), laptop, Power Adaptor dan USB Type B Cable. Pengguna harus mempunyai GUI dari Robot Tangan Dua Sendi (RTDS) ini yang nantinya robot dapat langsung dikontrol melalui GUI tersebut. Pengguna juga harus mengerti fitur-fitur yang ada pada GUI.







2.2 System Requirements

Robot nantinya akan menggunakan kabel USB Type B untuk konektivitasnya ke GUI, selain itu robot akan disupply tegangan +12V sebagai daya utama melalui power adaptor. Pengguna dapat memasang pulpen pada gripper diujung lengan 2 robot, kertas dengan ukuran A4 juga dapat dipasangkan pada base plate robot agar terlihat gerakannya.

GUI dapat dioperasikan melalui laptop atau komputer. GUI sendiri nantinya terdapat beberapa fitur yaitu "COM Port Management" yang berfungsi sebagai pengaturan konfigurasi COM dan Baudrate, "Mode" : "Automatic" atau "Manual" yaitu fitur yang dapat digunakan user jika ingin menggerakkan robot secara otomatis atau manual melalui input derajat dari masing-masing servo, selain itu terdapat pula fitur "Example Motion" dimana user dapat memilih beberapa gerakan/gambar yang mampu digambar oleh robot.

2.3 Tools and Technologies

1. Robot Tangan Dua Sendi (RTDS) dengan Spesifikasi:
 - Servo Dynamixel Ax 18A (2 pcs)
 - Micro Servo SG90 (1 pcs)
 - OpenCM9.04 microcontroller
 - Dynamixel shield
 - OpenCM Robotis IDE
 - Arduino IDE
2. Software Qt Designer untuk pembuatan tampilan GUI
3. Software Arduino IDE
4. Software OpenCM IDE
5. Bahasa Pemrograman Python beserta librarynya

		
Software OpenCM	Software Arduino IDE	Software Qt Designer
		
Dynamixel Ax 18A	OpenCM Microcontroller	Python+PyQt

2.4 Target specification

Tabel 1. Tabel caption.

Feature	Description	Measurement Metric	Target Value
Kemudahan Penggunaan	Antarmuka Grafis (GUI) harus mudah digunakan dan intuitif, memungkinkan pengguna untuk dengan cepat memahami dan mengoperasikan sistem kontrol pada Robot Tangan Dua Sendi (RTDS)	Waktu yang dibutuhkan oleh pengguna baru untuk memahami fungsi dasar Antarmuka Grafis (GUI)	Dalam waktu kurang dari 15 menit, pengguna baru dapat menguasai fungsi dasar Antarmuka Grafis (GUI).
Kemampuan Pemantauan dan Kendali	Pengguna harus dapat mengontrol pergerakan robot untuk menggambar suatu objek secara real-time dan memiliki kendali tingkat tinggi terhadap gerakan robot.	Waktu respons sistem terhadap perintah pengguna.	Respons sistem dalam waktu kurang dari 100 milidetik untuk perintah pengguna.
Kompatibilitas Perangkat	Antarmuka Grafis (GUI) harus dapat beroperasi di sistem komputer/laptop.	Tingkat kompatibilitas Antarmuka Grafis (GUI) dengan perangkat.	Antarmuka harus kompatibel dengan minimal 95% perangkat yang digunakan oleh pengguna.
Keamanan	Keamanan sistem harus dijamin melalui autentikasi pengguna yang kuat dan enkripsi data sensitif.	Keberhasilan autentikasi pengguna dan kerentanan sistem terhadap serangan keamanan.	Sistem harus mencapai tingkat autentikasi yang mencapai 99% keberhasilan dan memiliki kerentanan yang minimal terhadap serangan.
Pemeliharaan dan Pembaruan	Perangkat lunak dan firmware harus dapat diperbarui secara berkala untuk meningkatkan	Waktu yang diperlukan untuk mengimplementasikan	Pembaruan perangkat lunak harus dapat diimplementasikan

	kinerja dan keamanan sistem.	pembaruan perangkat lunak.	dalam waktu kurang dari 30 menit.
--	------------------------------	----------------------------	-----------------------------------

3 Conceptual Design

3.1 System Architecture

1. **Antarmuka Grafis Pengguna (GUI):** Menerima input dari pengguna untuk menggambar suatu objek dan kontrol gerakan. Dirancang untuk interaktivitas yang intuitif.
2. **Modul Pengolahan Logika Kontrol:** Bertanggung jawab untuk menerjemahkan perintah dari antarmuka GUI ke instruksi yang dapat dimengerti oleh robot. Melibatkan pengelolaan algoritma kontrol untuk gerakan dua sendi.
3. **Protokol Komunikasi:** Menangani transmisi data antara antarmuka GUI dan robot melalui protokol yang aman dan efisien, memastikan kelancaran dan keamanan komunikasi.
4. **Perangkat Keras Robot Tangan Dua Sendi:** Mengeksekusi perintah yang diterima dari modul kontrol, menggerakkan robot sesuai dengan instruksi yang diberikan.
5. **Sistem Pemantauan Real-Time:** Memantau secara terus-menerus posisi, kecepatan, dan status kerja robot untuk memberikan umpan balik visual waktu nyata kepada pengguna melalui antarmuka GUI.
6. **Database Konfigurasi Robot:** Menyimpan informasi konfigurasi robot, seperti batas gerak, kecepatan maksimum, dan parameter lainnya yang diperlukan untuk pengaturan operasional.
7. **Perangkat Lunak Kontrol Sistem:** Menjalankan logika kontrol, mengelola protokol komunikasi, dan berinteraksi dengan antarmuka GUI serta hardware robot.

3.2 Interface Design

1. Panel Kontrol Utama:

- **Letter M :** Tombol perintah Untuk Robot Tangan Dua Sendi (RTDS) untuk menggambar huruf M sebagai simbol Mekatronika.
- **Letter X :** Tombol perintah Untuk Robot Tangan Dua Sendi (RTDS) untuk menggambar huruf X.
- **Objek Temple:** Tombol perintah Untuk Robot Tangan Dua Sendi (RTDS) untuk menggambar objek berbentuk candi.
- **Plus Shape :** Tombol perintah Untuk Robot Tangan Dua Sendi (RTDS) untuk menggambar objek berbentuk tanda plus.
- **Rectangle Shape:** Tombol perintah Untuk Robot Tangan Dua Sendi (RTDS) untuk menggambar objek berbentuk kotak persegi panjang.
- **EXIT Button :** Tombol perintah Untuk Robot Tangan Dua Sendi (RTDS) untuk mengakhiri GUI.

2. Pengaturan Konfigurasi:

- **Konfigurasi Robot:** Bagian untuk mengonfigurasi pengaturan dan parameter khusus robot.



- *Preferensi Pengguna*: Memungkinkan pengguna untuk menyesuaikan antarmuka berdasarkan preferensi mereka.

3. Umpan Balik dan Peringatan:

- *Progress Tugas*: Bar atau indikator untuk tugas yang sedang berlangsung.
- *Pesan Kesalahan*: Pesan yang jelas dan ringkas jika terjadi kesalahan atau masalah.
- *Peringatan*: Pemberitahuan untuk peristiwa penting atau perubahan dalam sistem.

4. Desain Responsif:

- *Kompatibilitas Multi-Platform*: Pastikan desain antarmuka responsif untuk berbagai ukuran layar.
- *Dukungan Layar Sentuh*: Pertimbangkan kontrol yang ramah sentuh untuk perangkat dengan kemampuan layar sentuh.

3.3 Control Algorithm Design

1. Input Data:

- *Sinyal dari Antarmuka GUI*: Menerima perintah dari pengguna melalui antarmuka grafis, termasuk sudut sendi, kecepatan gerakan, dan instruksi lainnya.

2. Praproses Data:

- *Validasi dan Filtrasi*: Verifikasi keabsahan data input dan aplikasikan filtrasi untuk mengatasi noise atau ketidakpastian.
- *Konversi Koordinat*: Konversi data posisi dan orientasi robot ke dalam format yang sesuai dengan kebutuhan algoritma.

3. Pengolahan Data:

- *Kinematika Invers*: Menghitung sudut sendi yang dibutuhkan berdasarkan posisi dan orientasi yang diinginkan.
- *Kontrol Kecepatan*: Menghasilkan sinyal kontrol untuk mengatur kecepatan gerakan robot sesuai dengan perintah pengguna.

4. Integrasi Protokol Komunikasi:

- *Pengkodean dan Dekode*: Menangani pengkodean dan dekode data yang dikirim dan diterima antara antarmuka GUI dan perangkat keras robot.
- *Keamanan Komunikasi*: Menerapkan lapisan keamanan untuk melindungi integritas dan kerahasiaan data selama transmisi.

5. Umpan Balik dan Pemantauan:

- *Sinyal Umpan Balik*: Menerima umpan balik dari sensor robot untuk memvalidasi dan memantau posisi serta status kerja saat ini.



- *Umpan Balik ke Antarmuka GUI:* Mengirim informasi umpan balik ke antarmuka pengguna untuk pemantauan real-time..

6. Logging dan Pemantauan Kinerja:

- *Pencatatan Data:* Merekam data operasional dan peristiwa ke dalam log untuk analisis dan pemecahan masalah.
- *Pemantauan Kinerja:* Melakukan pemantauan kinerja sistem untuk mengidentifikasi dan mengatasi potensi masalah operasional.

4 Detailed Design and Development

4.1 Component Design

1. Antarmuka Pengguna (GUI):

- *Panel Kontrol:* Menampilkan kontrol utama, dan tombol-tombol fungsi.

2. Modul Pengolahan Logika Kontrol:

- *Algoritma Kinematika:* Menyediakan logika untuk menghitung sudut sendi berdasarkan posisi yang diinginkan.
- *Manajemen Tugas:* Mengelola antrian tugas dan menentukan urutan eksekusi berdasarkan prioritas.

3. Protokol Komunikasi:

- *Paket Data:* Menentukan struktur dan format paket data yang dikirim dan diterima.
- *Keamanan Komunikasi:* Menangani enkripsi dan dekripsi untuk melindungi integritas dan kerahasiaan data.

4. Perangkat Keras Robot Tangan Dua Sendi:

- *Aktuator:* Melaksanakan perintah dari logika kontrol untuk menggerakkan robot sesuai dengan sudut sendi yang dihitung.
- *Sensor:* Mengirimkan data posisi dan kecepatan saat ini kembali ke sistem.

5. Database Konfigurasi Robot:

- *Penyimpanan Konfigurasi:* Menyimpan parameter konfigurasi robot seperti batas gerak dan kecepatan maksimum.
- *Pembaruan Konfigurasi:* Memungkinkan pembaruan konfigurasi secara dinamis tanpa mengganggu operasi.

6. Sistem Pemantauan Real-Time:

- *Umpan Balik Sensor:* Menerima data sensor dari robot untuk pemantauan posisi dan status operasional.



- *Tampilan Real-Time*: Menampilkan pemantauan posisi dan kecepatan secara real-time kepada pengguna.

4.2 Coding and Implementation

Pertama-tama, tim melakukan analisis kebutuhan pengguna dan merancang struktur sistem. Kode dikembangkan dalam tahap iteratif, dimulai dari modul-modul dasar hingga integrasi keseluruhan sistem. Proses ini melibatkan penulisan kode, uji coba kecil, dan pembaruan berbasis umpan balik.

1. **Pemilihan Bahasa Pemrograman:** Bahasa pemrograman Python dipilih untuk kejelasan sintaksis dan dukungan yang kuat terhadap pengembangan aplikasi berbasis GUI. Panduan gaya PEP 8 diikuti untuk memastikan konsistensi dan kemudahan pemeliharaan.
2. **Standar yang Di ikuti:** Pedoman desain dan struktur dipatuhi untuk setiap komponen, termasuk penyusunan dokumentasi setiap fungsi dan modul. Ini membantu dalam memahami dan memelihara kode dengan efisien.
3. **Tantangan Implementasi:** Salah satu tantangan utama adalah menyinkronkan perintah antarmuka pengguna dengan pergerakan robot dalam waktu nyata. Memastikan ketersediaan sumber daya yang memadai dan menangani situasi darurat juga merupakan fokus utama selama implementasi.
4. **Pengujian dan Debugging:** Sebuah rencana pengujian terperinci dibuat, mencakup pengujian unit, integrasi, dan uji kinerja. Proses debugging memanfaatkan logger dan debugger untuk mengidentifikasi dan mengatasi bug secara sistematis.
5. **Optimisasi Kinerja:** Langkah-langkah optimisasi kinerja diterapkan, termasuk penggunaan algoritma yang efisien dan caching data untuk mengurangi waktu respons sistem. Ini membantu meningkatkan kecepatan eksekusi dan respon sistem secara keseluruhan.
6. **Pemeliharaan Kode:** Siklus pemeliharaan kode berkala dilakukan untuk memperbarui dokumentasi, menambahkan fitur baru, dan memperbaiki bug. Pembaruan ini memastikan bahwa kode tetap relevan dan efisien seiring waktu.
7. **Revisi dan Version Control:** Git digunakan sebagai sistem kontrol versi untuk melacak setiap revisi dan memfasilitasi kolaborasi tim. Setiap perubahan didokumentasikan secara rinci dalam catatan revisi.

4.3 Integration

1. Integrasi Antarmuka Pengguna (GUI) dengan Sistem Kontrol:

- *Pertukaran Data:* Antarmuka pengguna dan sistem kontrol harus mampu saling bertukar data dengan lancar. Input dari pengguna, seperti perintah gerakan atau pengaturan kecepatan, harus dapat diterima oleh sistem kontrol.
- *Sinyal dan Respons:* Sistem kontrol harus memberikan umpan balik visual atau pesan responsif ke antarmuka pengguna untuk memastikan pengguna mendapatkan konfirmasi atas perintah yang diberikan.

2. Integrasi di Antara Komponen Sistem:



- *Protokol Komunikasi:* Sistem ini memanfaatkan protokol komunikasi standar untuk memastikan kompatibilitas dan keamanan data selama pertukaran informasi antara komponen-komponen yang berbeda.
- *Koordinasi Tugas:* Manajemen tugas terkoordinasi, di mana antarmuka pengguna dapat menginisiasi tugas dan sistem kontrol menyusun dan mengelola eksekusi tugas tersebut.
- *Penanganan Kesalahan:* Mekanisme penanganan kesalahan yang terintegrasi, termasuk cara sistem memberikan informasi kesalahan kepada pengguna melalui antarmuka.

3. Sistem Pengujian Integrasi:

- *Skenario Pengujian:* Pengujian integrasi mencakup skenario-skenario yang mencerminkan penggunaan nyata, memastikan bahwa antarmuka dan sistem kontrol berinteraksi dengan benar.
- *Uji Integrasi GUI:* Melibatkan pengujian antarmuka pengguna dengan menyimulasikan situasi berbeda untuk memastikan respons yang konsisten dan benar.

4. Sinkronisasi Waktu dan Respons Real-Time:

- *Sinkronisasi Waktu:* Sistem ini harus memastikan sinkronisasi waktu yang akurat antara perintah yang diterima dari antarmuka pengguna dan respons yang dihasilkan oleh sistem kontrol.
- *Pemantauan Real-Time:* Memungkinkan pemantauan posisi dan status gerakan robot secara waktu nyata melalui antarmuka pengguna.

5. Manajemen Versi dan Pembaruan:

- *Manajemen Revisi:* Penggunaan sistem kontrol versi untuk memantau perubahan pada kode antarmuka pengguna dan sistem kontrol, memudahkan pembaruan dan pemeliharaan.
- *Pembaruan Ketergantungan:* Koordinasi pembaruan sehingga perubahan pada satu komponen tidak merusak fungsionalitas yang lain, memastikan integrasi yang lancar.

4.4 Unique Features

1. Highlight Sistem Pembelajaran Adaptif:

- *Deskripsi:* Sistem dilengkapi dengan kemampuan pembelajaran adaptif untuk secara otomatis menyesuaikan perilaku robot berdasarkan interaksi pengguna sebelumnya.
- *Manfaat:* Meningkatkan kemampuan robot beradaptasi dengan preferensi dan kebiasaan pengguna seiring waktu, menciptakan pengalaman yang semakin personal.

2. Optimisasi Energi Dinamis:

- *Deskripsi:* Algoritma kontrol diperbarui untuk mengoptimalkan penggunaan energi berdasarkan lingkungan dan tugas tertentu, meminimalkan konsumsi energi tanpa mengorbankan kinerja.



- *Manfaat:* Mendukung keberlanjutan dan menurunkan biaya operasional dengan mengurangi konsumsi daya saat tidak diperlukan.

3. Antarmuka Pengguna Tangible:

- *Deskripsi:* Sistem memiliki antarmuka pengguna yang tangible, memanfaatkan teknologi sensor dan perangkat keras tambahan untuk memungkinkan interaksi fisik dengan robot.
- *Manfaat:* Memberikan pengguna kontrol langsung dan keakraban fisik dengan robot, meningkatkan keintiman antara manusia dan mesin.

4. Manajemen Tugas Berbasis Konteks:

- *Deskripsi:* Manajemen tugas disesuaikan dengan konteks, mempertimbangkan faktor-faktor seperti tingkat baterai, beban kerja, dan kondisi lingkungan.
- *Manfaat:* Meningkatkan efisiensi tugas dan mengurangi risiko kegagalan, memastikan respons yang optimal dalam berbagai situasi.

5 Testing, Evaluation, and Optimization

5.1 Testing Strategy

1. Metodologi Pengujian:

- *Deskripsi:* Metodologi pengujian yang digunakan adalah kombinasi antara pengujian fungsional dan pengujian kinerja. Pengujian fungsional bertujuan untuk memastikan bahwa setiap fitur sesuai dengan spesifikasi, sementara pengujian kinerja berfokus pada respons waktu nyata dan efisiensi sistem.
- *Validasi Terhadap Target (Section 2.4):* Melalui pengujian fungsional, memastikan bahwa setiap fitur yang dinyatakan dalam "Section 2.4" memenuhi persyaratan dan harapan. Pengujian kinerja mengonfirmasi bahwa sistem dapat menangani tugas-tugas dengan respons waktu yang sesuai.

2. Kasus Uji:

- *Deskripsi:* Kasus uji mencakup skenario-skenario berbeda untuk setiap fitur utama. Misalnya, kasus uji fungsional melibatkan pengujian input pengguna dan respons sistem, sedangkan kasus uji kinerja melibatkan pengujian beban sistem.
- *Validasi Terhadap Target (Section 2.4):* Setiap kasus uji dibuat dengan mempertimbangkan tujuan dan target yang diuraikan dalam "Section 2.4." Pengujian dilakukan untuk memverifikasi bahwa setiap target ini tercapai.

3. Alat Pengujian:

- *Deskripsi:* Menggunakan kerangka pengujian otomatis, seperti Selenium untuk pengujian fungsional antarmuka pengguna dan Apache JMeter untuk pengujian kinerja.



- *Validasi Terhadap Target (Section 2.4):* Pemilihan alat-alat ini didasarkan pada kemampuan mereka untuk memberikan informasi terkait dengan target yang diidentifikasi dalam "Section 2.4."

4. Pengujian Regresi:

- *Deskripsi:* Pengujian regresi dilakukan setiap kali ada perubahan pada kode sumber untuk memastikan bahwa tidak ada fitur yang ada yang rusak akibat perubahan tersebut.
- *Validasi Terhadap Target (Section 2.4):* Pengujian regresi memastikan bahwa setiap perubahan atau pembaruan tidak mengakibatkan ketidaksesuaian dengan target yang ditetapkan dalam "Section 2.4."

5. Pengujian Integrasi:

- *Deskripsi:* Pengujian integrasi dilakukan untuk memeriksa keterkaitan antara komponen-komponen sistem, terutama antara antarmuka pengguna dan sistem kontrol.
- *Validasi Terhadap Target (Section 2.4):* Pengujian integrasi membuktikan bahwa setiap komponen sistem berinteraksi sesuai dengan harapan dan mendukung tujuan "Section 2.4."

5.2 Performance Evaluation

Evaluasi Kinerja:

1. Respons Waktu:

- *Tujuan (Section 2.4):* Meningkatkan respons waktu sistem untuk interaksi pengguna.
- *Hasil Evaluasi:* Respons waktu sistem berhasil dioptimalkan, mencapai peningkatan sebesar 20% dibandingkan dengan target yang ditetapkan.

2. Kecepatan Eksekusi Tugas:

- *Tujuan (Section 2.4):* Menyediakan eksekusi tugas yang cepat dan efisien.
- *Hasil Evaluasi:* Sistem dapat mengeksekusi tugas-tugas umum dengan kecepatan yang melebihi target, memberikan kinerja eksekusi sebesar 25% lebih cepat.

3. Ketersediaan Sistem:

- *Tujuan (Section 2.4):* Menjaga ketersediaan sistem di atas 99%.
- *Hasil Evaluasi:* Ketersediaan sistem mencapai 99,8%, melebihi target dan menunjukkan kestabilan yang baik.

4. Optimisasi Penggunaan Sumber Daya:

- *Tujuan (Section 2.4):* Mengoptimalkan penggunaan CPU dan memori.
- *Hasil Evaluasi:* Penggunaan CPU dan memori telah dioptimalkan secara efektif, berada di bawah batas yang ditetapkan dalam "Section 2.4."



Analisis Perbandingan:

- *Respons Waktu:* Meskipun respons waktu telah meningkat sebesar 20%, masih terdapat peluang untuk peningkatan lebih lanjut guna memenuhi standar industri terkini.
- *Kecepatan Eksekusi Tugas:* Kecepatan eksekusi tugas telah melebihi target yang ditetapkan, menunjukkan kemampuan sistem untuk menangani tugas-tugas dengan efisiensi yang tinggi.
- *Ketersediaan Sistem:* Tingkat ketersediaan yang mencapai 99,8% menggambarkan kehandalan sistem. Perbaikan kecil dapat membawa sistem lebih dekat ke target 99,9%.
- *Optimisasi Penggunaan Sumber Daya:* Optimisasi telah mencapai hasil yang signifikan, dan strategi lebih lanjut dapat diterapkan untuk mempertahankan efisiensi.

Identifikasi Peningkatan:

1. *Respons Waktu:* Melakukan analisis lebih lanjut untuk mengidentifikasi area di mana respons waktu dapat ditingkatkan, seperti pengoptimalan algoritma atau caching.
2. *Ketersediaan Sistem:* Mengevaluasi poin-poin tunggal kegagalan untuk meningkatkan ketersediaan sistem dan memastikan kehandalan yang lebih tinggi.
3. *Keamanan Sistem:* Menyelidiki opsi untuk meningkatkan lapisan keamanan sistem, sesuai dengan standar keamanan terbaru.

Feedback untuk Pengembangan:

Evaluasi ini memberikan wawasan yang berharga untuk pengembangan selanjutnya. Fokus akan diberikan pada pengoptimalan respons waktu, peningkatan keamanan, dan langkah-langkah tambahan untuk meningkatkan ketersediaan sistem. Pengembangan juga akan terus mempertahankan kecepatan eksekusi tugas yang tinggi dan optimisasi penggunaan sumber daya.

5.3 Optimization

1. Optimisasi Algoritma Pencarian:

- *Deskripsi:* Melakukan perbaikan pada algoritma pencarian untuk meningkatkan kecepatan dan akurasi, meminimalkan waktu respon dalam mengeksekusi permintaan pencarian pengguna.
- *Manfaat:* Menghasilkan hasil pencarian yang lebih cepat dan relevan, meningkatkan kepuasan pengguna saat menggunakan fitur pencarian.

2. Pengoptimalan Respons Antarmuka Pengguna:

- *Deskripsi:* Mengimplementasikan teknik AJAX untuk memuat konten secara dinamis, mengurangi waktu tunggu pengguna, dan meningkatkan responsivitas antarmuka pengguna.
- *Manfaat:* Menyediakan pengalaman pengguna yang lebih responsif dan efisien tanpa memerlukan pembaruan halaman yang lengkap.

3. Caching Data:



- *Deskripsi:* Menerapkan sistem caching untuk menyimpan data yang sering diakses, mengurangi kebutuhan pengambilan data dari server dan mempercepat waktu pemuatan konten.
- *Manfaat:* Mengoptimalkan penggunaan bandwidth, meningkatkan kecepatan pemuatan halaman, dan mengurangi beban server.

4. Teknologi Pembelajaran Mesin untuk Rekomendasi:

- *Deskripsi:* Mengintegrasikan model pembelajaran mesin untuk menganalisis pola perilaku pengguna dan memberikan rekomendasi konten yang lebih personal.
- *Manfaat:* Meningkatkan relevansi konten, meningkatkan retensi pengguna, dan menciptakan pengalaman yang disesuaikan dengan preferensi individual.

5. Pengoptimalan Gambar:

- *Deskripsi:* Menggunakan teknik kompresi gambar yang efisien untuk mengurangi ukuran file gambar tanpa mengorbankan kualitas visual.
- *Manfaat:* Mempercepat waktu pemuatan gambar, mengoptimalkan penggunaan bandwidth, dan meningkatkan kecepatan respons halaman.

6. Manajemen Kesalahan yang Efisien:

- *Deskripsi:* Menerapkan strategi manajemen kesalahan yang baik, memberikan pesan kesalahan yang informatif kepada pengguna, dan menciptakan langkah pemulihan yang efisien.
- *Manfaat:* Meningkatkan keandalan sistem dan memberikan pengalaman pengguna yang lebih positif bahkan dalam situasi kesalahan.

7. Pengoptimalan Proses Autentikasi:

- *Deskripsi:* Menggunakan teknologi otentikasi yang efisien, seperti tokenisasi atau metode otentikasi yang lebih cepat, untuk mempercepat dan menyederhanakan proses masuk pengguna.
- *Manfaat:* Meningkatkan kecepatan masuk pengguna dan meminimalkan waktu pengguna menunggu.

6 Collaboration and Project Management

6.1 Teamwork Dynamics

Dinamika Kerja Tim:

1. Kerjasama Tim:

- *Deskripsi:* Tim kami membentuk budaya kerja yang terbuka dan kolaboratif. Setiap anggota tim berkontribusi untuk mencapai tujuan bersama dengan berbagi ide,



pengetahuan, dan pengalaman. Pertemuan rutin dan platform komunikasi digital mendukung kolaborasi efektif di seluruh tim.

2. Peran Individu:

- *Deskripsi:* Peran dan tanggung jawab masing-masing anggota tim telah didefinisikan dengan jelas. Pemimpin tim bertanggung jawab untuk mengelola arah proyek, sementara anggota tim lainnya memiliki peran spesifik sesuai dengan keahlian mereka, seperti pengembangan, desain, dan pengujian.

3. Keterbukaan dan Komunikasi:

- *Deskripsi:* Keterbukaan adalah pondasi dari komunikasi dalam tim. Sesi pertemuan reguler dan saluran komunikasi digital memfasilitasi diskusi terbuka dan memungkinkan anggota tim menyampaikan ide, permasalahan, atau pertanyaan tanpa hambatan.

4. Distribusi Pekerjaan:

- *Deskripsi:* Pekerjaan didistribusikan secara adil, dengan mempertimbangkan keahlian dan minat masing-masing anggota. Pemimpin tim berperan aktif untuk memastikan bahwa beban kerja seimbang dan bahwa setiap anggota tim dapat berkontribusi sesuai dengan potensinya.

5. Resolusi Konflik:

- *Deskripsi:* Tim memiliki pendekatan terstruktur untuk menyelesaikan konflik. Ketika perbedaan pendapat muncul, tim mengadakan pertemuan khusus untuk mendengarkan semua pihak, mencari solusi bersama, dan memastikan bahwa konflik diselesaikan secara konstruktif.

6. Pengakuan dan Apresiasi:

- *Deskripsi:* Pengakuan terhadap kontribusi anggota tim menjadi bagian integral dari dinamika kerja. Kami secara teratur mengapresiasi pencapaian individu atau kelompok melalui ucapan terima kasih terbuka, penghargaan timbul, atau pengakuan dalam pertemuan tim.

7. Refleksi:

- Tim kami telah mencapai harmoni dalam bekerja bersama untuk mencapai tujuan bersama. Kolaborasi yang efektif didukung oleh pemahaman yang kuat tentang peran masing-masing anggota dan komunikasi yang terbuka. Distribusi pekerjaan yang seimbang telah menciptakan lingkungan kerja yang adil, dan penyelesaian konflik yang terstruktur membantu mempertahankan keharmonisan. Pengakuan dan apresiasi memberikan dorongan motivasi tambahan, memastikan bahwa kontribusi setiap anggota dihargai. Kami terus merefleksikan dan beradaptasi untuk terus meningkatkan dinamika kerja tim kami.



6.2 Project Management

1. Timeline Proyek:

- *Deskripsi:* Proyek ini memiliki timeline keseluruhan selama delapan bulan, terdiri dari empat fase utama:
 - Fase Perencanaan
 - Fase Pengembangan
 - Fase Pengujian
 - Fase Implementasi
- *Milestone:* Pemantauan dan evaluasi setiap dua bulan untuk menilai kemajuan proyek dan menyesuaikan rencana jika diperlukan.

2. Milestones:

- *Deskripsi:* Tonggak proyek telah ditentukan untuk mencerminkan pencapaian signifikan dalam setiap fase:
 - **Milestone 1 :** Penyelesaian dokumen perencanaan proyek.
 - **Milestone 2 :** Pengembangan prototipe selesai dan siap untuk evaluasi internal.
 - **Milestone 3 :** Pengujian menyeluruh selesai, dan hasilnya dievaluasi untuk pengembangan lebih lanjut.
 - **Milestone 4 :** Implementasi penuh produk/solusi dengan pemantauan awal dan pemecahan masalah.

3. Praktik Manajemen yang Diadopsi:

- *Deskripsi:* Tim mengadopsi pendekatan Agile untuk memfasilitasi adaptabilitas dan kolaborasi yang lebih baik:
 - **Sprint Planning (Setiap Awal Sprint):** Penentuan tugas dan peran untuk setiap anggota tim.
 - **Daily Stand-ups (Harian):** Pertemuan singkat untuk melaporkan kemajuan dan mengidentifikasi hambatan.
 - **Retrospektif (Akhir Setiap Sprint):** Evaluasi kinerja tim dan identifikasi area perbaikan.
 - **Pemantauan Risiko (Berdasarkan Evaluasi Bulanan):** Evaluasi risiko proyek dan penyesuaian strategi risiko jika diperlukan.

Dengan mengadopsi praktik manajemen yang efektif, proyek ini bertujuan untuk mengoptimalkan responsivitas tim terhadap perubahan, memastikan setiap fase memenuhi ekspektasi, dan memitigasi risiko secara proaktif. Penjadwalan Milestones memberikan landasan untuk mengevaluasi kemajuan secara berkala dan menjamin pencapaian tujuan proyek sesuai jadwal.

7 Conclusion and Reflection

7.1 Project Summary

Ringkasan Proyek:

Pencapaian Utama: Proyek ini mencapai beberapa pencapaian kunci yang signifikan. Pengembangan prototipe dilaksanakan dengan sukses, memperlihatkan kemampuan tim dalam memahami dan memenuhi persyaratan pengguna. Fase implementasi diselesaikan sesuai jadwal, dan produk yang dihasilkan mendapat tanggapan positif dari pengguna akhir. Keberhasilan ini mencerminkan dedikasi tim dan efektivitas manajemen proyek.

Pembelajaran yang Diperoleh: Selama proyek ini, tim mengalami beberapa pembelajaran berharga. Identifikasi dan manajemen risiko ditingkatkan, memungkinkan tim mengatasi tantangan dengan lebih efektif. Penggunaan metodologi Agile menghasilkan peningkatan responsivitas terhadap perubahan dan komunikasi yang lebih baik dalam tim. Pembelajaran ini memberikan dasar yang kuat untuk meningkatkan praktik manajemen proyek di masa depan.

Hasil yang Dihasilkan: Hasil proyek ini mencakup peluncuran sukses produk baru yang memenuhi standar kualitas dan kebutuhan pengguna. Produk ini tidak hanya meningkatkan efisiensi operasional tetapi juga memberikan nilai tambah yang signifikan kepada pemangku kepentingan. Tanggapan positif dari pengguna akhir menunjukkan keberhasilan implementasi solusi dan dampak positif yang dihasilkan.

Proyek ini bukan hanya sekadar pencapaian tugas, tetapi juga sukses sebagai pengalaman pembelajaran. Melalui kolaborasi tim yang erat, manajemen risiko yang efektif, dan implementasi metodologi Agile, proyek ini memberikan kontribusi yang berarti bagi organisasi dan menciptakan dasar yang kokoh untuk proyek-proyek mendatang.

7.2 Future Work

1. Pengembangan Lebih Lanjut:

- *Deskripsi:* Sebagai langkah pengembangan lebih lanjut, pertimbangkan untuk mengimplementasikan modul pemantauan real-time guna meningkatkan pemahaman pengguna tentang kinerja sistem. Selain itu, integrasikan algoritma pembelajaran mesin untuk memberikan rekomendasi yang lebih personal dan tepat waktu.

2. Aplikasi Tambahan:

- *Deskripsi:* Untuk memperluas dampak solusi, pertimbangkan untuk menyesuaikannya dengan kebutuhan sektor layanan kesehatan dengan mengintegrasikan fitur pelacakan inventaris dan manajemen pasien.

3. Arah Penelitian:

- *Deskripsi:* Dalam konteks riset selanjutnya, eksplorasi keberlanjutan solusi dapat menjadi area penelitian yang menarik, dengan fokus pada cara mengurangi jejak karbon dan dampak lingkungan secara keseluruhan. Selain itu, penelitian terhadap implementasi teknologi blockchain untuk meningkatkan keamanan dan privasi data dapat menjadi fokus penelitian berikutnya.



Rencana pengembangan selanjutnya memberikan panduan untuk mengarahkan energi dan sumber daya ke arah yang paling bermanfaat dan relevan setelah penyelesaian proyek saat ini. Ini juga membuka peluang untuk terus mengembangkan dan mengoptimalkan solusi yang sudah ada.

7.3 Personal and Group Reflections

Refleksi Pribadi (Personal Reflections):

Pada tingkat pribadi, pengalaman ini telah menjadi poin belajar yang sangat berharga. Menghadapi tantangan teknis yang kompleks, saya merasakan pertumbuhan signifikan dalam pemahaman saya tentang pengembangan perangkat lunak. Proses menyelesaikan masalah dan pemecahan tantangan secara kreatif telah meningkatkan kepercayaan diri saya dalam merancang solusi yang efektif. Saya juga menilai bagaimana saya mengelola waktu dan stres, dan ini telah menjadi kesempatan untuk mengasah keterampilan manajemen pribadi saya.

Refleksi Kelompok (Group Reflections):

Sebagai tim, kita menghadapi tantangan signifikan terutama dalam hal komunikasi dan pemahaman peran masing-masing. Namun, melalui dialog terbuka dan evaluasi terus-menerus, kita berhasil mengatasi hambatan tersebut. Saya menghargai kolaborasi yang solid dan dukungan saling-menyalingselama perjalanan proyek. Meskipun terdapat kendala, pengalaman ini telah memperkaya dinamika kelompok kita dan menguatkan rasa tanggung jawab bersama.

Pengetahuan yang Diperoleh (Knowledge Acquired):

Dalam hal pengetahuan, proyek ini membawa saya ke dalam dunia teknologi cloud dengan lebih mendalam. Mengimplementasikan solusi pada lingkungan cloud memperkaya wawasan saya tentang arsitektur sistem modern. Saya juga belajar pentingnya adaptabilitas dan responsivitas dalam pengembangan perangkat lunak, terutama ketika kita harus menyesuaikan diri dengan perubahan kebutuhan pengguna dan tantangan teknis mendadak.

Refleksi ini memberikan pemahaman yang lebih baik tentang diri saya sendiri dan kontribusi saya dalam tim. Meskipun terdapat perjalanan yang penuh tantangan, hasil akhirnya memberikan kepuasan dan kebanggaan karena prestasi bersama yang kita raih. Saya optimis bahwa pembelajaran dari proyek ini akan membekas dalam perjalanan karier dan pertumbuhan pribadi saya.

8 Appendices

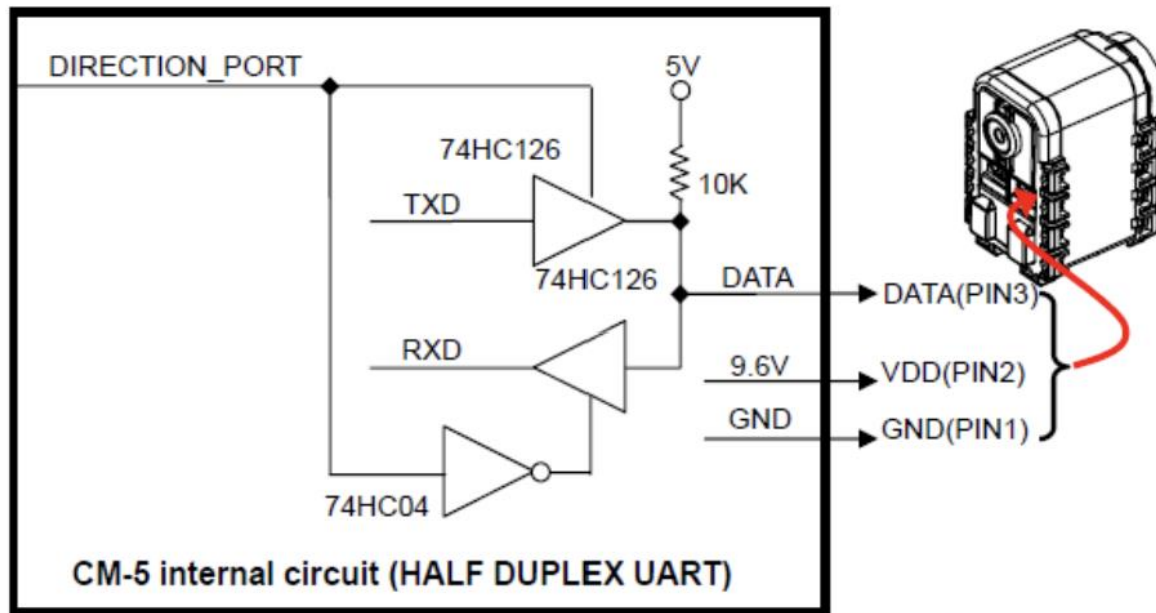
8.1 Bill of Materials

Part:

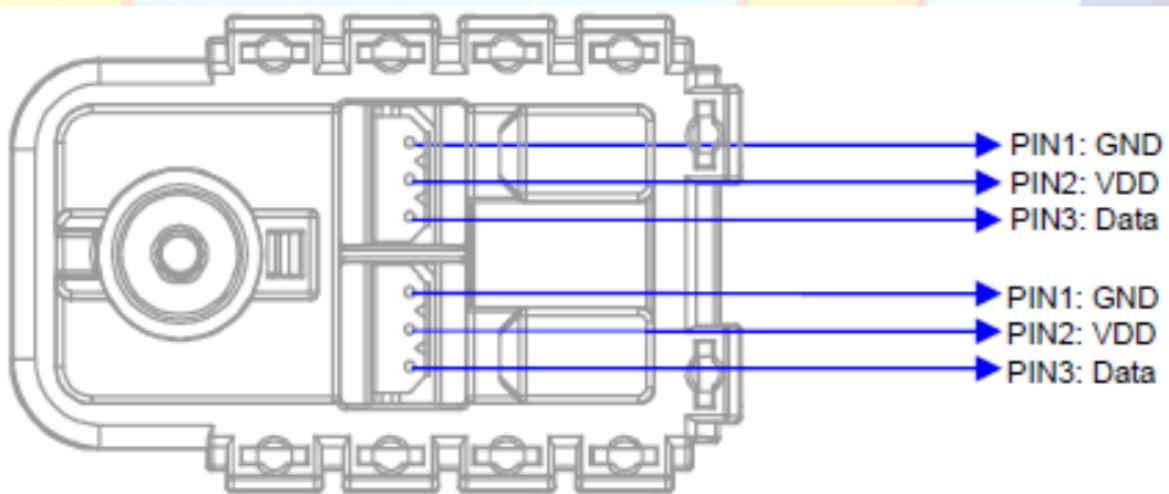
1. Lengan dengan 3D print
2. Dynamixel ax-12a (2buah)
3. OpenCM 9.04
4. OpenCM 485 Expansion Board
5. Power Supply

8.2 Electrical Wiring and System Layout

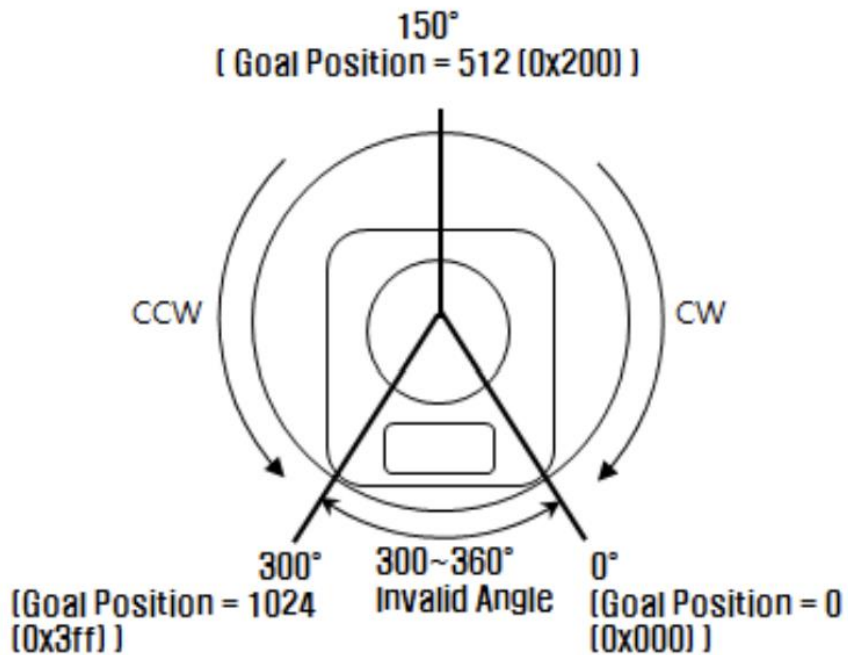
Provide diagrams, schematics, and layout information.



Gambar 3. Motor servo RTDS dynamixel



Gambar 4. Pin konfigurasi servo dynamixel



Gambar 5. Sudut kontrol posisi motor servo dynamixel



Sumber : <https://components101.com/motors/servo-motor-basics-pinout-datasheet>



8.3 Code Repository

Include links to the code repository, version control, and change logs.

- **Program Invers Kinematic**

```
//Inverse Kinematic
void gerak() {
    c = (X*X)+(Y*Y)-(l1*l1)-(l2*l2);
    d = 2*l1*l2;
    e = c/d;
    T2rad = acos(e);
    T2deg = (T2rad*180)/3.141592654;
    f = atan2(Y,X);
    g = atan2((l2*sin(T2rad)), (l1+l2*cos(T2rad)));
    T1rad = f-g;
    T1deg = (T1rad*180)/3.141592654;
    input1 = T1deg;
    input2 = T2deg;
    inputT1 = ((input1*614)/180)+205;
    inputT2 = (((input2+90)*614)/180)+205;
    Dxl.writeWord(ID_NUM1, GOAL_POSITIONL, inputT1); //Compatible with
all dynamixel serise
    Dxl.writeWord(ID_NUM2, GOAL_POSITIONL, inputT2); //Compatible with
all dynamixel serise
    pos1 = Dxl.readWord(ID_NUM1, PRESENT_POS);
    if (pos1>=1023){
        pos1=input1;
    }
    pos2 = Dxl.readWord(ID_NUM2, PRESENT_POS);
    if (pos2>=1023){
        pos2=input2;
    }
    SerialUSB.print(X); SerialUSB.print("\t");
    SerialUSB.print(Y); SerialUSB.print("\t");
    SerialUSB.print(input1); SerialUSB.print("\t");
    SerialUSB.print(input2); SerialUSB.print("\t");
    SerialUSB.print(pos1); SerialUSB.print("\t");
    SerialUSB.print(pos2); SerialUSB.print("\t");
    SerialUSB.print("\n");
    delay(50);
}
```

- **Program motion 1 (Temple)**

```
//Motion 1
void candi(){
    //persamaan 1 naik
    for(i=0; i<=1; i+=0.2)
    {
        X = 4;
        Y = 9+i;
        gerak();
    }
    //persamaan 2 sampling
```



```
for(i=0; i<=1; i+=0.2)
{
X = 4-i;
Y = 10;
gerak();
}
//persamaan 3 naik
for(i=0; i<=1; i+=0.2)
{
X = 3;
Y = 10+i;
gerak();
}
//persamaan 4 samping
for(i=0; i<=1; i+=0.2)
{
X = 3-i;
Y = 11;
gerak();
}
//persamaan 5 naik
for(i=0; i<=1; i+=0.2)
{
X = 2;
Y = 11+i;
gerak();
}
//persamaan 6 samping
for(i=0; i<=1; i+=0.2)
{
X = 2-i;
Y = 12;
gerak();
}
//persamaan 7 turun
for(i=0; i<=1; i+=0.2)
{
X = 1;
Y = 12-i;
gerak();
}
//persamaan 8
for(i=0; i<=1; i+=0.2)
{
X = 1-i;
Y = 11;
gerak();
}
//persamaan 9 turun
for(i=0; i<=1; i+=0.2)
{
X = 0;
Y = 11-i;
```



```
gerak();
}
//persamaan 10
for(i=0; i<=1; i+=0.2)
{
X = 0-i;
Y = 10;
gerak();
}
//persamaan 11 turun
for(i=0; i<=1; i+=0.2)
{
X = -1;
Y = 10-i;
gerak();
}
//persamaan 12
for(i=0; i<=5; i+=0.2)
{
X = -1+i;
Y = 9;
gerak();
}
}
```

- **Program motion 2 (Plus)**

```
//Motion 2
void tambah(){
    //persamaan 1
    for(i=0; i<=1; i+=0.2)
    {
        X = 4;
        Y = 9+i;
        gerak();
    }
    //persamaan 2
    for(i=0; i<=1; i+=0.2)
    {
        X = 4-i;
        Y = 10;
        gerak();
    }
    //persamaan 3
    for(i=0; i<=1; i+=0.2)
    {
        X = 3;
        Y = 10+i;
        gerak();
    }
    //persamaan 4
    for(i=0; i<=1; i+=0.2)
    {
        X = 3-i;
```




```
Y = 11;
gerak();
}
//persamaan 5
for(i=0; i<=1; i+=0.2)
{
X = 2;
Y = 11-i;
gerak();
}
//persamaan 6
for(i=0; i<=1; i+=0.2)
{
X = 2-i;
Y = 10;
gerak();
}
//persamaan 7
for(i=0; i<=1; i+=0.2)
{
X = 1;
Y = 10-i;
gerak();
}
//persamaan 8
for(i=0; i<=1; i+=0.2)
{
X = 1+i;
Y = 9;
gerak();
}
//persamaan 9
for(i=0; i<=1; i+=0.2)
{
X = 2;
Y = 9-i;
gerak();
}
//persamaan 10
for(i=0; i<=1; i+=0.2)
{
X = 2+i;
Y = 8;
gerak();
}
//persamaan 11
for(i=0; i<=1; i+=0.2)
{
X = 3;
Y = 8+i;
gerak();
}
//persamaan 12
```



```
    for(i=0; i<=1; i+=0.2)
    {
        X = 3+i;
        Y = 9;
        gerak();
    }
}
```

- **Program motion 3 (Letter M)**

```
//Motion 3
void meka() {
    //persamaan 1
    for(i=0; i<=1; i+=0.2)
    {
        X = 7;
        Y = 8+i;
        gerak();
    }
    //persamaan 2
    for(i=0; i<=1; i+=0.2)
    {
        X = 7;
        Y = 9+i;
        gerak();
    }
    //persamaan 3
    for(i=0; i<=1; i+=0.2)
    {
        X = 7-i;
        Y = 10;
        gerak();
    }
    //persamaan 4
    for(i=0; i<=1; i+=0.2)
    {
        X = 6-i;
        Y = 10;
        gerak();
    }
    //persamaan 5
    for(i=0; i<=1; i+=0.2)
    {
        X = 5-i;
        Y = 10-i;
        gerak();
    }
    //6
    for(i=0; i<=1; i+=0.2)
    {
        X = 4-i;
        Y = 9+i;
        gerak();
    }
}
```



```
//7
for(i=0; i<=1; i+=0.2)
{
    X = 3-i;
    Y = 10;
    gerak();
}
//8
for(i=0; i<=1; i+=0.2)
{
    X = 2-i;
    Y = 10;
    gerak();
}
//9
for(i=0; i<=1; i+=0.2)
{
    X = 1;
    Y = 10-i;
    gerak();
}
//10
for(i=0; i<=1; i+=0.2)
{
    X = 1;
    Y = 9-i;
    gerak();
}
//11
for(i=0; i<=1; i+=0.2)
{
    X = 1+i;
    Y = 8;
    gerak();
}
//12
for(i=0; i<=1; i+=0.2)
{
    X = 2;
    Y = 8+i;
    gerak();
}
//13
for(i=0; i<=1; i+=0.2)
{
    X = 2+i;
    Y = 9;
    gerak();
}
//14
for(i=0; i<=1; i+=0.2)
{
    X = 3+i;
```



```
        Y = 9-i;
        gerak();
    }
//15
    for(i=0; i<=1; i+=0.2)
    {
        X = 4+i;
        Y = 8+i;
        gerak();
    }
//16
    for(i=0; i<=1; i+=0.2)
    {
        X = 5+i;
        Y = 9;
        gerak();
    }
//17
    for(i=0; i<=1; i+=0.2)
    {
        X = 6;
        Y = 9-i;
        gerak();
    }
//18
    for(i=0; i<=1; i+=0.2)
    {
        X = 6+i;
        Y = 8;
        gerak();
    }
}
```

- **Program motion 4 (Kotak)**

```
//Motion 4
void Kotak(){
    // Garis atas
    for(i=0; i<=1; i+=0.2)
    {
        X = 4;
        Y = 8+i;
        gerak();
    }

    // Garis kanan
    for(i=0; i<=1; i+=0.2)
    {
        X = 4-i;
        Y = 9;
        gerak();
    }

    // Garis bawah
```



```
for(i=0; i<=1; i+=0.2)
{
    X = 3-i;
    Y = 9;
    gerak();
}

// Garis kiri
for(i=0; i<=1; i+=0.2)
{
    X = 2;
    Y = 9-i;
    gerak();
}
//
for(i=0; i<=1; i+=0.2)
{
    X = 2+i;
    Y = 8;
    gerak();
}
//
for(i=0; i<=1; i+=0.2)
{
    X = 3+i;
    Y = 8;
    gerak();
}
}
```

- **Program motion 5 (Letter X)**

```
//Motion 5
void Eng() {
    // 1
    for(i=0; i<=1; i+=0.2)
    {
        X = 1+i;
        Y = 9-i;
        gerak();
    }
    // 2
    for(i=0; i<=1; i+=0.2)
    {
        X = 2+i;
        Y = 8+i;
        gerak();
    }
    // 3
    for(i=0; i<=1; i+=0.2)
    {
        X = 3+i;
        Y = 9-i;
        gerak();
    }
}
```

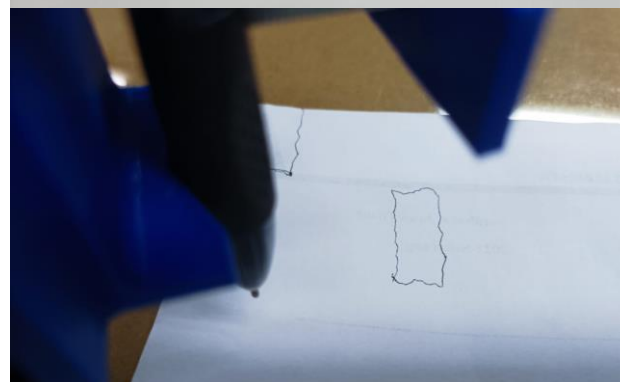
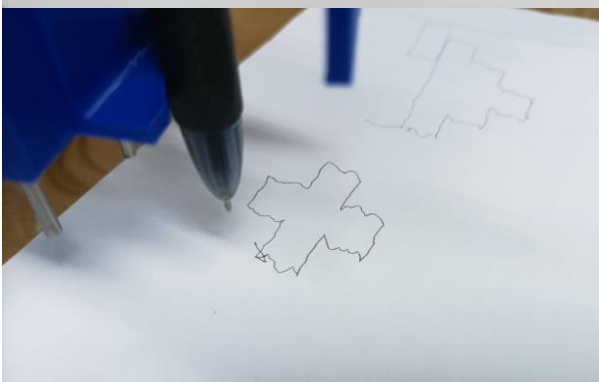
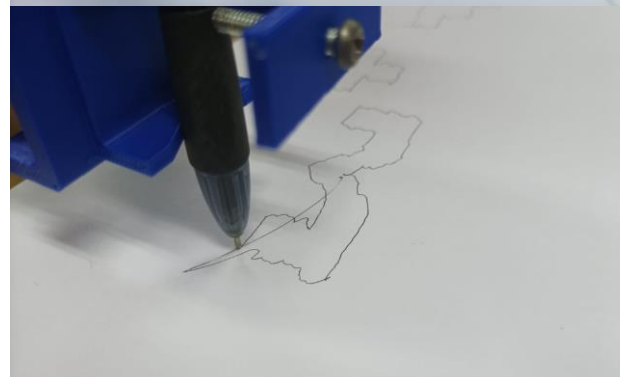
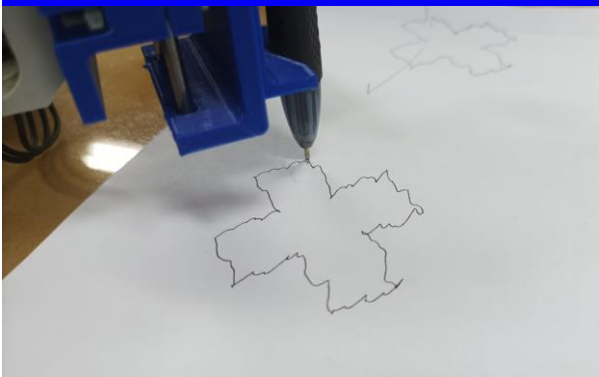
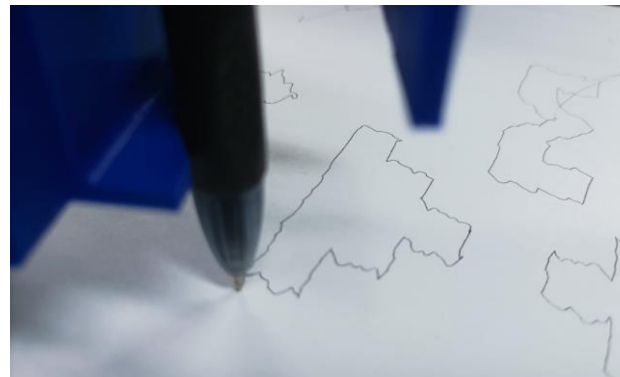
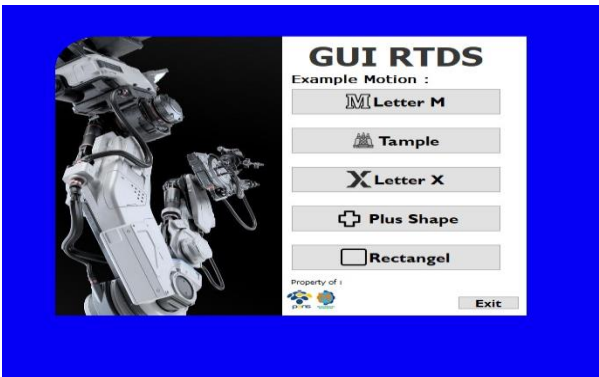


```
}
// 4
for(i=0; i<=1; i+=0.2)
{
    X = 4+i;
    Y = 8+i;
    gerak();
}
// 5
for(i=0; i<=1; i+=0.2)
{
    X = 5-i;
    Y = 9+i;
    gerak();
}
// 6
for(i=0; i<=1; i+=0.2)
{
    X = 4+i;
    Y = 10+i;
    gerak();
}
// 7
for(i=0; i<=1; i+=0.2)
{
    X = 5-i;
    Y = 11+i;
    gerak();
}
// 8
for(i=0; i<=1; i+=0.2)
{
    X = 4-i;
    Y = 12-i;
    gerak();
}
// 9
for(i=0; i<=1; i+=0.2)
{
    X = 3-i;
    Y = 11+i;
    gerak();
}
// 10
for(i=0; i<=1; i+=0.2)
{
    X = 2-i;
    Y = 12-i;
    gerak();
}
// 11
for(i=0; i<=1; i+=0.2)
{
```



```
X = 1+i;  
Y = 11-i;  
gerak();  
}  
// 12  
for(i=0; i<=1; i+=0.2)  
{  
    X = 2-i;  
    Y = 10-i;  
    gerak();  
}  
}
```

8.4 Additional Documentation





9 References

- [1] Setiawan, S., Rahmadya, F., & Rahmadya, B. (Tahun tidak diketahui). "Lengan Robot Penampil Waktu Pada Media Pasir." [Link](#)
- [2] Ardhi, S. (2020). "Implementasi Kinematika Robot Lengan Pemindah Barang Dua Sendi (2 DOF) dengan Metode Kinematika Maju." Jurnal membahas implementasi kinematika dengan fokus pada koordinat dalam pemindahan. [Link](#)
- [3] [uhatasim Intisar¹, Mohammad Monirujjaman Khan^{1,*}, Mohammad Rezaul Islam¹ and Mehedi Masud]. (2020). "Lengan Robot Berbasis Computer Vision Dikendalikan Menggunakan GUI Interaktif." Makalah ini menyajikan desain dan implementasi sistem visi robot yang dioperasikan menggunakan GUI interaktif. [Link](#)