M. Fahrizal (4121600002), M. Ishom Muwaffaq (4121600010), M. Yusuf Fitrian Nur Ihsan (4121600017)

{fahrizal, Ishommuwaffaq, 17yusuf}@me.student.pens.ac.id

Abstract

Proyek ini berfokus pada pengembangan gerakan robot humanoid, dengan desain fisik robot yang telah tersedia. Tim teknisi kami bertugas merencanakan, mengintegrasikan komponen perangkat keras, serta mengembangkan kode pengendali gerakan. Dengan fokus pada pengembangan gerakan yang beragam, kami berhasil mengimplementasikan berbagai gerakan yang telah ditentukan pada robot humanoid yang sudah ada. Dokumentasi proyek ini mencakup langkah-langkah yang kami ambil dalam mencapai solusi desain gerakan robot yang sukses. Proyek ini merupakan kontribusi yang berharga dalam pengembangan teknologi robotika.

Laporan Akhir Pemrograman Sistem Mekatronik Kelompok 2 – Robot Humanoid

Sistem pemrograman mekatronika

Team Humanoid 2

Dalam proyek pengembangan robot humanoid dan antarmuka pengendali GUI kami, tim teknisi kami mulai dengan analisis awal untuk menentukan tujuan dan kebutuhan perangkat keras serta perangkat lunak. Kami merancang struktur robot, memilih komponen yang sesuai, dan mengembangkan perangkat lunak untuk mengendalikan gerakan robot melalui GUI. Setelah integrasi perangkat keras dan lunak, kami melakukan pengujian dan debugging, mengoptimalkan kinerja robot dan antarmuka GUI. Hasilnya adalah solusi desain akhir yang berhasil menghasilkan robot humanoid yang dapat dikendalikan dengan sukses melalui GUI, dengan dokumentasi yang mencakup semua langkah yang diperlukan untuk membangun sistem ini.

Daftar Isi

[1 Pendahuluan dan Analisis Awal 4](#_Toc152761615)

[1.1 Konteks Proyek 4](#_Toc152761616)

[1.1.1 Rumusan Masalah 4](#_Toc152761617)

[1.1.2 Tujuan 5](#_Toc152761618)

[1.2 Proses Pemikiran Awal 5](#_Toc152761619)

[2 Analisis dan Spesifikasi Kebutuhan 5](#_Toc152761620)

[2.1 Persyaratan Pengguna 5](#_Toc152761621)

[2.2 Keperluan yang Dibututhkan 6](#_Toc152761622)

[2.3 Alat dan Teknologi 6](#_Toc152761628)

[2.4 Spesifikasi Target 6](#_Toc152761630)

[3 Konsep Desain 7](#_Toc152761631)

[3.1 Arsitektur Sistem 7](#_Toc152761632)

[3.2 Desain Antar Muka 8](#_Toc152761633)

[3.3 Desain Kontrol Algoritma 8](#_Toc152761637)

[4 Develop Detail Desain 9](#_Toc152761646)

[4.1 Desain Komponen 9](#_Toc152761647)

[4.2 Implementasi dan Koding 10](#_Toc152761648)

[4.3 Integrasi 12](#_Toc152761649)

[4.4 Fitur Unik 15](#_Toc152761650)

[5 Pengujian, Evaluasi, dan Optimasi 16](#_Toc152761651)

[5.1 Strategi Test 16](#_Toc152761652)

[5.2 Evaluasi Performa 16](#_Toc152761653)

[5.3 Optimalisasi 16](#_Toc152761654)

[6 Kolaborasi dan Management Proyek 17](#_Toc152761655)

[6.1 Dinamika Kerja Tim 17](#_Toc152761656)

[6.2 Management Proyek 17](#_Toc152761657)

[7 Kesimpulan dan Refleksi 19](#_Toc152761658)

[7.1 Kesimpulan Proyek 19](#_Toc152761659)

[7.2 Proyek Kedepanya 19](#_Toc152761660)

[7.3 Refleksi Pribadi dan Kelompok 19](#_Toc152761661)

[8 Lampiran 19](#_Toc152761662)

[8.1 Harga dari Tiap Material 19](#_Toc152761663)

[8.2 Pengkabelan Listrik dan Tata Letak Sistem 20](#_Toc152761664)

[8.3 Repositori Kode 21](#_Toc152761665)

[8.4 Dokumentasi Tambahan 22](#_Toc152761666)

[9 Referensi 22](#_Toc152761667)

# 1 Pendahuluan dan Analisis Awal

## Konteks Proyek

Pengembangan gerakan robot humanoid adalah aspek penting dalam pengembangan robotika karena mencakup kemampuan robot untuk meniru gerakan manusia atau mengekspresikan berbagai gerakan yang serupa dengan manusia. Robot humanoid memiliki bentuk yang menyerupai manusia, dengan struktur tubuh, anggota gerak, dan kemampuan untuk melakukan tugas-tugas yang biasanya dilakukan oleh manusia.

Potensi besar robot humanoid terletak pada berbagai aplikasi yang dapat dilakukannya:

* Industri : Robot humanoid dapat digunakan dalam lingkungan industri untuk melakukan tugas-tugas seperti pemindahan barang, perakitan, dan pekerjaan yang memerlukan presisi tinggi. Kemampuan untuk menyesuaikan diri dengan lingkungan kerja manusia dapat mengurangi risiko kecelakaan dan memperbaiki efisiensi.\
* Penelitian : Pengembangan robot humanoid menjadi alat yang penting dalam riset dan pengembangan, terutama dalam studi ergonomi manusia, interaksi manusia-mesin, dan simulasi lingkungan yang kompleks.
* Hiburan: Di bidang hiburan, robot humanoid dapat digunakan sebagai karakter animasi hidup atau untuk pertunjukan yang menarik minat penonton. Mereka juga dapat berperan sebagai asisten dalam sektor layanan pelanggan.

Meskipun robot humanoid memiliki potensi besar, kemampuan geraknya masih menjadi tantangan utama. Beberapa alasan utama termasuk:

* Kompleksitas Mekanika: Merancang sistem mekanika yang dapat meniru gerakan manusia secara akurat memerlukan teknologi yang canggih dan presisi tinggi.
* Kontrol dan Perangkat Lunak: Memprogram robot untuk bergerak dengan cara yang serupa dengan manusia memerlukan pengembangan perangkat lunak yang kompleks dan algoritma kontrol yang canggih.
* Koordinasi dan Keseimbangan: Robot humanoid harus memiliki kemampuan untuk menjaga keseimbangan saat melakukan gerakan kompleks dan koordinasi yang serupa dengan gerakan manusia.=
* Sensor dan Feedback: Pentingnya sistem sensor yang tepat untuk memberikan umpan balik yang akurat kepada robot tentang lingkungan sekitarnya agar dapat menyesuaikan gerakannya dengan baik.

Proyek-proyek dalam pengembangan gerakan robot humanoid bertujuan untuk mengatasi tantangan-tantangan ini. Hal ini meliputi penggunaan teknologi sensor yang lebih baik, perangkat keras yang lebih canggih, pengembangan algoritma kontrol yang lebih pintar, dan pemahaman yang lebih mendalam tentang biomekanika manusia. Dengan meningkatnya kemampuan gerak robot humanoid, diharapkan bahwa kegunaannya dalam berbagai aplikasi akan semakin ditingkatkan dan menjadi lebih luas.

## Rumusan Masalah

Keterbatasan dalam bagaimana robot humanoid bergerak menghambat potensinya dalam situasi di mana gerakan yang diperlukan harus lebih kompleks dan menyerupai gerakan manusia secara lebih nyata.

## Tujuan

Tujuan dari proyek ini adalah untuk meningkatkan kualitas gerakan yang lebih nyata dan beragam pada robot humanoid yang sudah ada, dengan tujuan meningkatkan efisiensi penggunaannya dalam berbagai konteks aplikasi. Keberhasilan proyek ini diharapkan akan mendorong kemajuan dalam pengembangan robot humanoid dan meningkatkan kontribusinya dalam berbagai bidang.

## 1.2 Proses Pemikiran Awal

Proses pengembangan gerakan robot humanoid kami dimulai dengan menulis tujuan untuk mengidentifikasi peluang dan ide-ide kreatif. Ide-ide awal meliputi pembaruan pada algoritma gerakan, perbaikan dalam navigasi, dan peningkatan dalam interaksi manusia-robot.

Selama analisis ide-ide tersebut, kami mengidentifikasi peluang untuk meningkatkan gerakan dengan mengoptimalkan algoritma pengendali dan perangkat keras yang ada. Kami melihat peluang dalam membuat gerakan lebih alami dan efisien dengan peningkatan penyesuaian lingkungan.

Kemudian, kami memutuskan untuk mengimplementasikan perbaikan tersebut, dengan fokus pada pengembangan gerakan yang lebih alami dan adaptif tanpa mengandalkan sensor eksternal. Selama proses ini, kami terus mendokumentasikan perkembangan dan evaluasi yang berperan dalam menghadapi tantangan dan memaksimalkan potensi perbaikan gerakan pada robot humanoid.

# 2 Analisis dan Spesifikasi Kebutuhan

## 2.1 Persyaratan Pengguna

Persyaratan pengguna adalah elemen kunci dalam pengembangan antarmuka pengguna grafis (GUI) dan sistem kontrol robot humanoid. Untuk memastikan keberhasilan proyek, kami telah mengidentifikasi persyaratan yang harus dipenuhi sesuai dengan kebutuhan dan harapan pengguna. Persyaratan ini mencakup:

* **Kemudahan Penggunaan:** Antarmuka harus dirancang agar mudah digunakan oleh pengguna dengan berbagai tingkat kemampuan. Pengguna menginginkan navigasi yang intuitif, ikon yang jelas, dan tata letak yang mudah dimengerti.
* **Kemampuan Pemantauan dan Kendali:** Pengguna ingin dapat memantau status robot humanoid secara real-time, termasuk posisi, kecepatan, dan kondisi perangkat lainnya. Mereka juga mengharapkan kemampuan untuk mengontrol gerakan robot dengan tingkat detail yang tinggi, termasuk perintah gerakan yang spesifik dan perintah berhenti.
* **Kompatibilitas Perangkat:** Antarmuka harus berfungsi dengan baik pada berbagai perangkat yang mungkin digunakan pengguna, termasuk komputer, smartphone, dan tablet. Kompatibilitas sistem operasi yang umum digunakan harus dijamin.
* **Keamanan:** Keamanan adalah prioritas utama. Sistem harus memerlukan otentikasi pengguna yang kuat, mengenkripsi data penting, dan mengelola hak akses dengan cermat.
* **Fleksibilitas Konfigurasi:** Pengguna ingin dapat mengatur berbagai parameter seperti kecepatan gerakan, sensitivitas kendali, dan preferensi lainnya sesuai dengan kebutuhan mereka.

## 2.2 Keperluan yang Dibututhkan

## Selain persyaratan pengguna, ada persyaratan sistem yang harus dipertimbangkan dalam pengembangan GUI dan sistem kontrol robot humanoid ini. Persyaratan sistem meliputi:

## Kinerja Sistem: Sistem harus mampu merespons perintah dengan cepat dan memiliki waktu latensi yang minimal. Kinerja yang baik diperlukan untuk menghindari keterlambatan dalam mengendalikan robot.

## Stabilitas: Sistem kontrol harus stabil dan dapat diandalkan selama pengoperasian. Hal ini diperlukan untuk mencegah insiden yang tidak diinginkan.

## Fleksibilitas Perangkat Keras: Perangkat keras yang digunakan harus memenuhi spesifikasi yang diperlukan dan harus bersifat fleksibel untuk mengakomodasi perubahan atau peningkatan di masa depan.

## Pemeliharaan dan Pembaruan: Perangkat lunak dan firmware sistem harus dapat diperbarui secara berkala untuk meningkatkan kinerja dan keamanan. Proses pemeliharaan rutin perangkat keras juga harus dijadwalkan.

## 2.3 Alat dan Teknologi

## Pengembangan antarmuka pengguna grafis dan sistem kontrol robot humanoid memerlukan pemanfaatan berbagai alat dan teknologi. Kami akan menggunakan alat dan teknologi berikut untuk mencapai tujuan proyek:

1. VSCode
2. OpenCM
3. Python/C++
4. Library tkinter

## 2.4 Spesifikasi Target

Tabel . Tabel caption.

|  |  |  |  |
| --- | --- | --- | --- |
| Fitur | Deskripsi | Tolak Ukur | Uraian Target |
| Gerakan Berdiri | Robot humanoid melakukan Gerakan berdiri dari posisi jongkok | Perpindahan dari titik awal yaitu jongkok ke posisi berdiri dengan tegak dan kokoh. | Robot dapat melakukan gerakan berdiri dengan baik dan sempurna tidak terbalik atau jatuh |
| Gerakan jongkok | Robot humanoid melakukan Gerakan jongkok dari posisi berdiri | Perpindahan dari posisi berdiri ke posisi jongkok. | Robot dapat melakuakan Gerakan jongkok dengan baik dan sempurna tidak terbalik atau jatuh. |
| Gerakan Berjalan | Robot humanoid berjalan lurus ke depan. | Robot berpindah dari posisi awal dengan cara berjalan lurus ke depan | Robot dapat berjalan ke depan dengan seimbang dan sempurna tidak terbalik atau jatuh |
| Gerakan Menendang | Robot humanoid melakukan Gerakan menendang bola | Robot mampu menendang sebuah bola yang ada di depan nya menggunakan kaki kanan | Robot dapat menendang bola menggunakan kaki kanan dengan seimbang dan sempurna tidak terbalik atau jatuh |
| Gerakan jalan di tempat | Robot Humanoid dapat melakukan Gerakan jalan di tempat dengan cara mengangkat kedua kakinya dengan cara bergantian. | Robot mampu melakukan Gerakan jelan di tempat dan tidak berpindah tempat. | Robot dapat melakukan Gerakan jalan di tempat dengan seimbang dan sempurna tidak jatuh atau terbalik. |
| Gerakan berjalan menyamping | Robot humanoid dapat berjalan menyamping ke kanan dengan cara memindahkan kaki kanannya ke samping lalu di ikuti kaki kirinya. | Robot mampu berpindah dari posisi awal ke arah samping kanan robot, dengan cara berjalan menyamping | Robot dapat melakukan Gerakan menyamping dengan seimbang dan sampurna tidak terjatuh atau terbalik. |
| GUI Control Motion | Menampilkan 7 Button Virtual untuk mengontrol Gerakan dari robot humanoid secara manual oleh user | Button terdiri dari Forward,Backward,Left,Right,Stop,Kicking,Run sudah  berfungsi untuk mengirim command ke robot Humanoid secara Real-Time | Button Virtual dapat berfungsi |
| GUI Monitoring IMU | Menampilkan Value dari IMU pada robot Humanoid berupa data ROLL,PITCH,YAW dengan interval 100ms | GUI telah terdapat label untuk menampilkan 3 data dari IMU yaitu ROLL,PITCH,YAW dengan interval data yaitu 100ms | GUI mampu menampilkan data IMU |
| GUI Monitoring Dynamixel | Menampilkan Value dari tiap-tiap servo Dynamixel pada robot humanoid dengan value 0-4096 | GUI masih dalam proses development dengan menambahkan label untuk memonitoring value pada tiap-tiap servo dynamixel | GUI pada label Servo Dynamixel masih dalam pengembagan |
| GUI Connectivity SerialUSB | GUI mampu menggunakan mode konektivitas yaitu secara serial USB untuk control motion dan monitoring Nilai dari robot humanoid | GUI mampu berkomunikasi secara serialUSB dengan memilih COM Port dan Baudrate pada Combo Box GUI | GUI mampu berkomunasi secara SerialUSB |
| GUI Connectivity Bluetooth Dekstop | GUI mampu menggunakan mode konektivitas yaitu secara Bluetooth pada Dekstop User untuk control motion dan monitoring Nilai dari robot humanoid | GUI mampu berkomunikasi secara Nirkable dengan Bluetooth Dekstop dengan memilih COM Port dan Baudrate pada Combo Box GUI | GUI mampu berkomunasi secara Bluetooth |

# 3 Konsep Desain

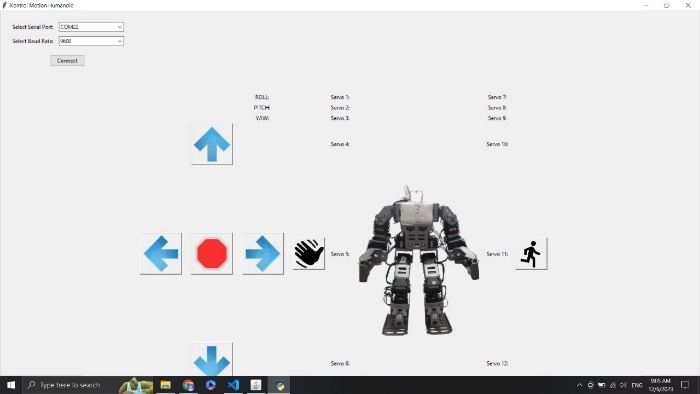
## 3.1 Arsitektur Sistem

**High-Level Architecture**

1. Antarmuka Pengguna (UI) : untuk komunikasi dengan pengguna melalui elemen UI seperti yang ditampilan GUI kami yaitu tombol berdiri , jongkok, jalan di tempat, berjalan, menendanag dan Jalan menyamping.
2. Logika : mengelola proses dan pengambilan keputusan berdasarkan input dari UI dan data dari akses data.
3. Database : untuk mendapatkan dan menyimpan informasi.

**GUI (Graphical User Interface)**

Desain sementara tampilan GUI kami :



**Control Logic**

Input : Ketika user menekan tombol yang ada pada GUI ini memberitahu control logic bahwa user ingin menjalankan robot sesuia dengan tombol yang di tekan.

Proses : Saat robot berada di posisi awal, control logic akan menginstruksikan robot untuk bergerak sesuai permintaan user saat menekan tombol.

Output : Robot akan merespon berdasarkan keputusan control logic. Jika robot bergerak, maka ini adalah output yang dihasilkan oleh control logic.

## 3.2 Desain Antar Muka

## Berikut adalah sketsa awal dari tampilan GUI :

## **Layout**

## 

## 3.3 Desain Kontrol Algoritma

# **Algoritma Kontrol**

# - Langkah 1 : Identifikasi variabel-variabel kendali utama.

* **Input:** Sensor-sensor (gyroscope, accelerometer, sensor tekanan, dll.).
* **Output:** Gerakan sendi-sendi (motor servo) pada robot.

# - Langkah 2 : Atur kontrol PID.

* Gunakan kontrol PID untuk mengatur posisi, kecepatan, dan percepatan setiap sendi.
* Pilih jenis kontrol PID (posisi, kecepatan, atau percepatan) sesuai dengan kebutuhan.

# - Langkah 3 : Lakukan tuning PID untuk optimalisasi respons system

* Sesuaikan parameter PID (Proporsional, Integral, Derivatif) untuk respons yang cepat dan stabil.
* Lakukan eksperimen dan pengukuran pada robot untuk mendapatkan parameter yang optimal.

# - Langkah 4 : Implementasikan algoritma kontrol.

# Terapkan algoritma PID pada kontrol masing-masing sendi.

# Sertakan logika keamanan untuk menghindari gerakan yang berbahaya atau konflik sendi.

# - Langkah 5 : Monitor dan sesuaikan algoritma kontrol secara berkala.

* Monitor performa robot selama operasi.
* Sesuaikan parameter PID jika diperlukan untuk mengatasi perubahan lingkungan atau kondisi kerja.

1. **Alur Pengolahan Data**

- Langkah 1 : Kumpulkan data dari sensor.

* Ambil data dari sensor-sensor yang terpasang pada robot (gyroscope, accelerometer, sensor tekanan pada kaki, dll.).

- Langkah 2 : Proses Data Sensor

* Terapkan algoritma pemrosesan data untuk mendapatkan informasi seperti orientasi tubuh, tinggi langkah, dll.

- Langkah 3 : Integrasi data ke dalam model kontrol.

* Gabungkan data hasil pemrosesan ke dalam algoritma kontrol PID untuk menentukan gerakan optimal.

- Langkah 4 : Evaluasi dan perbaiki algoritma pengolahan data secara berkala.

* Evaluasi kinerja robot humanoid dalam berbagai situasi.
* Perbaiki algoritma pengolahan data dan kontrol untuk meningkatkan kinerja dan keamanan.

# 4 Develop Detail Desain

## Desain Komponen

|  |  |
| --- | --- |
|  | OpenCm 9.04 |
|  | Dynamixel AX-12A |
| Dynamixel AX-18A Standard Smart Servo for Strength 55g | Dynamixel AX-18A |

## Implementasi dan Koding

* Program Berdiri

void berdiri(){

Dxl.setPosition(1, 512, 50); //ID 1 dynamixel moves to position 1

Dxl.setPosition(2, 512, 50); //ID 1 dynamixel moves to position 1

delay(500);

Dxl.setPosition(15, 612, 50); //tambah tinggi tambah kedepan

Dxl.setPosition(16, 412, 50); //tambah rendah tambah kedepan

Dxl.setPosition(13, 722, 50); //tambah tinggi tambah nekuk

Dxl.setPosition(14, 302, 50); //tambah rendah tambah nekuk

Dxl.setPosition(12, 352, 50); //tambah tinggi tambah mundur

Dxl.setPosition(11, 672, 50); //tambah tinggi tambah maju

delay(500);

Dxl.setPosition(10, 502, 50); //tambah rendah tambah melebar

Dxl.setPosition(9, 522, 50); //tambah tinggi tambah melebar

delay(500);

Dxl.setPosition(17, 522, 50); //ID 1 dynamixel moves to position 1

Dxl.setPosition(18, 502, 50); //ID 1 dynamixel moves to position 1

delay(500);

* **Program Duduk/jongkok**

void duduk(){

delay(3000);

Dxl.setPosition(13, 992, 50); //tambah tinggi tambah nekuk(722)

Dxl.setPosition(14, 32, 50); //tambah rendah tambah nekuk(302)

delay(500);

Dxl.setPosition(11, 872, 50); //tambah tinggi tambah maju(672)

Dxl.setPosition(12, 152, 50); //tambah tinggi tambah mundur(352)

Dxl.setPosition(15, 662, 50); ////tambah tinggi tambah kedepan(612)

Dxl.setPosition(16, 362, 50); //tambah rendah tambah kedepan(412)

}

* **Program move kanan**

void movekanan(){

//setup

delay(1000);

Dxl.setPosition(13, 622, 50); //tambah tinggi tambah nekuk(722)

Dxl.setPosition(14, 302, 50); //tambah rendah tambah nekuk(302)

Dxl.setPosition(17, 572, 50); //makin tinggi maskin ke dalam(522)

Dxl.setPosition(18, 552, 50); //makin rendah makin kedalam(502)

delay(500);

Dxl.setPosition(13, 722, 50); //tambah tinggi tambah nekuk(722)

Dxl.setPosition(14, 302, 50); //tambah rendah tambah nekuk(302)

delay(500);

//paha melebar

Dxl.setPosition(9, 542, 30); //tambah tinggi tambah melebar(522)

Dxl.setPosition(10, 482, 30); //tambah rendah tambah melebar(502)

Dxl.setPosition(17, 542, 50); //makin tinggi maskin ke dalam(522)

delay(500);

Dxl.setPosition(18, 482, 50); //makin rendah makin kedalam(502)

delay(1000);

//support kanan

Dxl.setPosition(12, 472, 175); //tambah tinggi tambah mundur(352)

Dxl.setPosition(14, 502, 175); //tambah rendah tambah nekuk(302)

Dxl.setPosition(16, 622, 175); //tambah rendah tambah kedepan(412)

Dxl.setPosition(17, 472, 120); //makin tinggi maskin ke dalam(522)

Dxl.setPosition(18, 452, 120); //makin rendah makin kedalam(502)

delay(1000);

Dxl.setPosition(12, 352, 150); //tambah tinggi tambah mundur(352)

Dxl.setPosition(14, 302, 150); //tambah rendah tambah nekuk(302)

Dxl.setPosition(16, 412, 150); //tambah rendah tambah kedepan(412)

delay(2000);

* **Program menendang**

//kick kiri

Dxl.setPosition(3, 632, 50); //semakin tinggi semakin melebar(532)

Dxl.setPosition(4, 432, 50); //semakin rendah semakin melebar(532)

Dxl.setPosition(5, 532, 50); //semakin tinggi semakin nekuk ke dalam(532)

Dxl.setPosition(6, 532, 50); //semakin rendah semakin nekuk ke dalam(532)

Dxl.setPosition(11, 672, 50); //tambah tinggi tambah maju(672)

Dxl.setPosition(12, 472, 50); //tambah tinggi tambah mundur(352)

Dxl.setPosition(13, 722, 50); //tambah tinggi tambah nekuk(722)

Dxl.setPosition(14, 102, 100); //tambah rendah tambah nekuk(302)

Dxl.setPosition(15, 612, 100); ////tambah tinggi tambah kedepan(612)

Dxl.setPosition(16, 512, 100); //tambah rendah tambah kedepan(412)

delay(2000);

Dxl.setPosition(12, 252, 250); //tambah tinggi tambah mundur(352)

Dxl.setPosition(13, 722, 50); //tambah tinggi tambah nekuk(722)

Dxl.setPosition(14, 352, 600); //tambah rendah tambah nekuk(302)

delay(1000);

* **Program jalan**

void jalan(){

//setup

delay(250);

Dxl.setPosition(13, 622, 50); //tambah tinggi tambah nekuk(722)

Dxl.setPosition(14, 302, 50); //tambah rendah tambah nekuk(302)

Dxl.setPosition(17, 572, 50); //makin tinggi maskin ke dalam(522)

Dxl.setPosition(18, 552, 50); //makin rendah makin kedalam(502)

Dxl.setPosition(15, 612, 50); ////tambah tinggi tambah kedepan(612)

Dxl.setPosition(16, 412, 50); //tambah rendah tambah kedepan(412)

delay(500);

Dxl.setPosition(13, 622, 50); //tambah tinggi tambah nekuk(722) //wasting

Dxl.setPosition(14, 302, 50); //tambah rendah tambah nekuk(302) //wasting

Dxl.setPosition(15, 412, 50); ////tambah tinggi tambah kedepan(612)

Dxl.setPosition(16, 412, 50); //tambah rendah tambah kedepan(412)

delay(500);

//paha maju

Dxl.setPosition(11, 702, 50); //tambah tinggi tambah maju(672)

Dxl.setPosition(12, 352, 50); //tambah tinggi tambah mundur(352)

Dxl.setPosition(15, 512, 100); ////tambah tinggi tambah kedepan(612)

Dxl.setPosition(16, 412, 100); //tambah rendah tambah kedepan(412)

delay(500);

//telapak normal

Dxl.setPosition(17, 522, 50); //makin tinggi makin ke dalam(522)

Dxl.setPosition(18, 502, 50); //makin rendah makin ke dalam(502)

Dxl.setPosition(15, 532, 100); ////tambah tinggi tambah kedepan(612)

Dxl.setPosition(16, 392, 100); //tambah rendah tambah kedepan(412)

delay(500);

//support

Dxl.setPosition(11, 722, 25); //tambah tinggi tambah maju(672)

Dxl.setPosition(12, 302, 25); //tambah tinggi tambah mundur(352)

delay(1000);

Dxl.setPosition(17, 472, 50); //makin tinggi makin ke dalam(522)

Dxl.setPosition(18, 452, 50); //makin rendah makin ke dalam(502)

Dxl.setPosition(15, 532, 150); ////tambah tinggi tambah kedepan(612)

Dxl.setPosition(16, 552, 150); //tambah rendah tambah kedepan(412)

delay(500);

Dxl.setPosition(15, 532, 150); ////tambah tinggi tambah kedepan(612)

Dxl.setPosition(16, 472, 150); //tambah rendah tambah kedepan(412)

delay(500);

Dxl.setPosition(13, 622, 50); //tambah tinggi tambah nekuk(722)

Dxl.setPosition(14, 402, 100); //tambah rendah tambah nekuk(302)

delay(500);

//menapak

Dxl.setPosition(17, 522, 50); //makin tinggi makin kedalam(522)

Dxl.setPosition(18, 502, 50); //makin rendah makin kedalam(502)

delay(250);

Dxl.setPosition(11, 672, 50); //tambah tinggi tambah maju(672)

Dxl.setPosition(12, 352, 50); //tambah tinggi tambah mundur(352)

Dxl.setPosition(13, 622, 50); //tambah tinggi tambah nekuk(722)

Dxl.setPosition(14, 402, 50); //tambah rendah tambah nekuk(302)

Dxl.setPosition(15, 532, 100); ////tambah tinggi tambah kedepan(612)

Dxl.setPosition(16, 502, 100); //tambah rendah tambah kedepan(412)

}

## Integrasi

* Program GUI yang kami gunakan

import tkinter as tk

from tkinter import PhotoImage

import serial

from tkinter import ttk

import serial.tools.list\_ports

ser = None

def button\_kanan\_click():

label.config(text="Tombol Kanan")

command = 'R'

ser.write(command.encode())

def button\_stop\_click():

label.config(text="Tombol STOP")

command = 'S'

ser.write(command.encode())

def button\_kiri\_click():

label.config(text="Tombol Kiri")

command = 'L'

ser.write(command.encode())

def button\_maju\_click():

label.config(text="Tombol Maju")

command = 'F'

ser.write(command.encode())

def button\_mundur\_click():

label.config(text="Tombol Mundur")

command = 'B'

ser.write(command.encode())

def button\_run\_click():

label.config(text="Tombol Run")

command = 'M'

ser.write(command.encode())

def button\_hand\_click():

label.config(text="Tombol Hand")

command = 'H'

ser.write(command.encode())

def read\_serial():

global ser

if ser and ser.in\_waiting:

data = ser.readline().decode().strip() # Membaca data serial dari Arduino

# Proses data yang diterima

if "ROLL" in data and "PITCH" in data and "YAW" in data:

values = data.split(',')

roll = float(values[0].split(':')[-1])

pitch = float(values[1].split(':')[-1])

yaw = float(values[2].split(':')[-1])

# Memperbarui label GUI dengan nilai ROLL, PITCH, dan YAW

label\_roll.config(text="ROLL : " + str(roll))

label\_pitch.config(text="PITCH : " + str(pitch))

label\_yaw.config(text="YAW : " + str(yaw))

# Memperbarui label GUI dengan nilai servo

label\_servo1.config(text=f"Servo 1: {int(values[3])}")

label\_servo2.config(text=f"Servo 2: {int(values[4])}")

label\_servo3.config(text=f"Servo 3: {int(values[5])}")

label\_servo4.config(text=f"Servo 4: {int(values[6])}")

label\_servo5.config(text=f"Servo 5: {int(values[7])}")

label\_servo6.config(text=f"Servo 6: {int(values[8])}")

label\_servo7.config(text=f"Servo 7: {int(values[9])}")

label\_servo8.config(text=f"Servo 8: {int(values[10])}")

label\_servo9.config(text=f"Servo 9: {int(values[11])}")

label\_servo10.config(text=f"Servo 10: {int(values[12])}")

label\_servo11.config(text=f"Servo 11: {int(values[13])}")

label\_servo12.config(text=f"Servo 12: {int(values[14])}")

root.after(100, read\_serial)

def connect\_serial():

global ser

selected\_port = port\_combo.get()

selected\_baudrate = baud\_combo.get()

try:

ser = serial.Serial(selected\_port, int(selected\_baudrate))

print(f"Connected to {selected\_port} with baudrate {selected\_baudrate}")

read\_serial()

# Do something with the serial connection, e.g., start reading data

except serial.SerialException as e:

print("Failed to connect:", e)

root = tk.Tk()

root.title("Kontrol Motion Humanoid")

# Get available serial ports

ports = [port.device for port in serial.tools.list\_ports.comports()]

# Baud rates list

baud\_rates = [300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400]

#ser = serial.Serial('COM23', 115200) # Sesuaikan dengan port serial Arduino Anda

# Create a frame

frame = ttk.Frame(root, padding="20")

frame.grid(row=0, column=0)

# Serial Port selection

port\_label = ttk.Label(frame, text="Select Serial Port:")

port\_label.grid(row=0, column=0, padx=5, pady=5)

port\_combo = ttk.Combobox(frame, values=ports)

port\_combo.grid(row=0, column=1, padx=5, pady=5)

port\_combo.current(0) # Set the default selection

# Baud Rate selection

baud\_label = ttk.Label(frame, text="Select Baud Rate:")

baud\_label.grid(row=1, column=0, padx=5, pady=5)

baud\_combo = ttk.Combobox(frame, values=baud\_rates)

baud\_combo.grid(row=1, column=1, padx=5, pady=5)

baud\_combo.current(4) # Set the default selection to 9600

# Connect button

connect\_button = ttk.Button(frame, text="Connect", command=connect\_serial)

connect\_button.grid(row=2, columnspan=2, padx=5, pady=15)

# Mengimpor gambar untuk tombol maju, mundur, dan pause

kanan\_image = PhotoImage(file="RIGHT.png")

stop\_image = PhotoImage(file="Stop.png")

kiri\_image = PhotoImage(file="LEFT.png")

maju\_image = PhotoImage(file="FWD.png")

mundur\_image = PhotoImage(file="BWD.png")

hand\_image = PhotoImage(file="Hand.png")

run\_image = PhotoImage(file="Run.png")

humanoid\_image = PhotoImage(file="Humanoid.png")

# Mengatur ukuran tombol dengan metode subsample

kiri\_image = kiri\_image.subsample(3) # Mengurangi ukuran gambar menjadi setengah

stop\_image = stop\_image.subsample(3) # Mengurangi ukuran gambar menjadi setengah

kanan\_image = kanan\_image.subsample(3) # Mengurangi ukuran gambar menjadi setengah

maju\_image = maju\_image.subsample(3) # Mengurangi ukuran gambar menjadi setengah

mundur\_image = mundur\_image.subsample(3) # Mengurangi ukuran gambar menjadi setengah

hand\_image = hand\_image.subsample(4) # Mengurangi ukuran gambar menjadi setengah

run\_image = run\_image.subsample(4) # Mengurangi ukuran gambar menjadi setengah

# Memperbesar gambar humanoid

humanoid\_image = humanoid\_image.zoom(2)

# Membuat tombol dengan gambar sebagai elemen tombol

button\_kiri = tk.Button(root, image=kiri\_image, command=button\_kiri\_click)

button\_stop = tk.Button(root, image=stop\_image, command=button\_stop\_click)

button\_kanan = tk.Button(root, image=kanan\_image, command=button\_kanan\_click)

button\_maju = tk.Button(root, image=maju\_image, command=button\_maju\_click)

button\_mundur = tk.Button(root, image=mundur\_image, command=button\_mundur\_click)

button\_hand = tk.Button(root, image=hand\_image, command=button\_hand\_click)

button\_run = tk.Button(root, image=run\_image, command=button\_run\_click)

# Mengatur tombol-tombol agar sejajar dan berada di tengah

button\_kiri.grid(row=7, column=2, padx=10)

button\_stop.grid(row=7, column=3, padx=10)

button\_kanan.grid(row=7, column=4, padx=10)

button\_maju.grid(row=6, column=3, padx=10)

button\_mundur.grid(row=8, column=3, padx=10)

button\_hand.grid(row=7, column=20, padx=10)

button\_run.grid(row=7, column=55, padx=10)

# Membuat label untuk menampilkan nilai ROLL, PITCH, dan YAW

label\_roll = tk.Label(root, text="ROLL: ")

label\_roll.grid(row=3, column=4, columnspan=2)

label\_pitch = tk.Label(root, text="PITCH: ")

label\_pitch.grid(row=4, column=4, columnspan=2)

label\_yaw = tk.Label(root, text="YAW: ")

label\_yaw.grid(row=5, column=4, columnspan=2)

abel\_servo1 = tk.Label(root, text="Servo 1: ")

label\_servo2 = tk.Label(root, text="Servo 2: ")

label\_servo3 = tk.Label(root, text="Servo 3: ")

label\_servo4 = tk.Label(root, text="Servo 4: ")

label\_servo5 = tk.Label(root, text="Servo 5: ")

label\_servo6 = tk.Label(root, text="Servo 6: ")

label\_servo7 = tk.Label(root, text="Servo 7: ")

label\_servo8 = tk.Label(root, text="Servo 8: ")

label\_servo9 = tk.Label(root, text="Servo 9: ")

label\_servo10 = tk.Label(root, text="Servo 10: ")

label\_servo11 = tk.Label(root, text="Servo 11: ")

label\_servo12 = tk.Label(root, text="Servo 12: ")

label\_servo1.grid(row=3, column=25, padx=0, pady=1)

label\_servo2.grid(row=4, column=25, padx=0, pady=1)

label\_servo3.grid(row=5, column=25, padx=0, pady=1)

label\_servo4.grid(row=6, column=25, padx=0, pady=1)

label\_servo5.grid(row=7, column=25, padx=0, pady=1)

label\_servo6.grid(row=8, column=25, padx=0, pady=1)

label\_servo7.grid(row=3, column=50, padx=0, pady=1)

label\_servo8.grid(row=4, column=50, padx=0, pady=1)

label\_servo9.grid(row=5, column=50, padx=0, pady=1)

label\_servo10.grid(row=6, column=50, padx=0, pady=1)

label\_servo11.grid(row=7, column=50, padx=0, pady=1)

label\_servo12.grid(row=8, column=50, padx=0, pady=1)

# Membuat gambar humanoid

humanoid\_image = humanoid\_image.subsample(3)

humanoid\_label = tk.Label(root, image=humanoid\_image)

humanoid\_label.grid(row=7, column=40)

label = tk.Label(root, text="")

label.grid(row=1, column=0, columnspan=3)

read\_serial()

root.mainloop()

## Fitur Unik

Pada beberapa gerakan robot humanoid yaitu pada gerakan menendang, berjalan, dan menyamping. Pada awal gerakan kami menbahkan gerakan menapak ke salah satu kaki robot yaitu pada kaki kanan agar robot dapat mengangkat dan memindahkan kaki kirinya sehingga robot dapat melakukan gerakan tersebut dengan sempurna, seimbang, dan tidak jatuh. Berikut adalah programnya :

//setup

delay(1000);

Dxl.setPosition(13, 622, 50); //tambah tinggi tambah nekuk(722)

Dxl.setPosition(14, 302, 50); //tambah rendah tambah nekuk(302)

Dxl.setPosition(17, 572, 50); //makin tinggi maskin ke dalam(522)

Dxl.setPosition(18, 552, 50); //makin rendah makin kedalam(502)

delay(500);

Dxl.setPosition(13, 722, 50); //tambah tinggi tambah nekuk(722)

Dxl.setPosition(14, 302, 50); //tambah rendah tambah nekuk(302)

delay(500);

//paha melebar

Dxl.setPosition(9, 542, 30); //tambah tinggi tambah melebar(522)

Dxl.setPosition(10, 482, 30); //tambah rendah tambah melebar(502)

Dxl.setPosition(17, 542, 50); //makin tinggi maskin ke dalam(522)

delay(500);

Dxl.setPosition(18, 482, 50); //makin rendah makin kedalam(502)

delay(1000);

//support kanan

Dxl.setPosition(12, 472, 175); //tambah tinggi tambah mundur(352)

Dxl.setPosition(14, 502, 175); //tambah rendah tambah nekuk(302)

Dxl.setPosition(16, 622, 175); //tambah rendah tambah kedepan(412)

Dxl.setPosition(17, 472, 120); //makin tinggi maskin ke dalam(522)

Dxl.setPosition(18, 452, 120); //makin rendah makin kedalam(502)

delay(1000);

Dxl.setPosition(12, 352, 150); //tambah tinggi tambah mundur(352)

Dxl.setPosition(14, 302, 150); //tambah rendah tambah nekuk(302)

Dxl.setPosition(16, 412, 150); //tambah rendah tambah kedepan(412)

delay(2000);

# 5 Pengujian, Evaluasi, dan Optimasi

## 5.1 Strategi Test

Langkah awal dalam melakukan pengujian program kami adalah melalui serangkaian uji coba dan koreksi, dimulai dengan perencanaan yang terperinci untuk setiap gerakan yang ingin dicapai. Kami membagi setiap gerakan menjadi beberapa bagian terpisah untuk memudahkan pengaturan, seperti saat kami mengembangkan program untuk gerakan berjalan, kami mengutamakan pengaturan untuk menapak dengan kaki kanan terlebih dahulu. Setelah mencapai tingkat keoptimalan pada program tersebut, kami melanjutkan dengan menyusun program yang mengangkat dan memindahkan kaki kiri robot ke depan, yang diikuti oleh gerakan kaki kanan. Kami melakukan penyesuaian pada posisi dan kecepatan dynamixel ketika terjadi hambatan pada kemampuan robot dalam melakukan gerakan secara maksimal.

Ketika kami menemui kendala atau keterbatasan dalam gerakan robot, kami melakukan modifikasi terhadap posisi dan kecepatan dynamixel untuk memperbaiki kinerja robot. Selanjutnya, kami memasukkan program-program tersebut ke dalam loop void agar robot dapat melakukan gerakan secara berulang dan terus menerus tanpa hambatan. Dengan pendekatan ini, kami secara sistematis mengembangkan dan menguji program-program untuk memastikan robot mampu melakukan gerakan dengan kemampuan maksimal secara berkelanjutan.

## 5.2 Evaluasi Performa

Setelah menyelesaikan pemrograman gerakan pada robot humanoid kami, kami tidak puas begitu saja. Kami secara berkesinambungan melakukan pemeriksaan dan uji coba berulang untuk memastikan apakah program yang kami buat berjalan secara optimal dan konsisten, bukan hanya memberikan hasil satu kali saja. Sebagai contoh, saat kami mengembangkan program untuk gerakan berdiri pertama, kami menciptakan program yang memungkinkan robot untuk berdiri tegak. Namun, kami menemukan kelemahan di mana robot tersebut tidak stabil dan rentan roboh saat menerima sentuhan dari luar.

Oleh karena itu, kami melakukan modifikasi pada program gerakan berdiri dengan memperhitungkan kondisi di mana robot berada dalam posisi yang sedikit membungkuk atau mirip dengan gerakan kuda-kuda. Tujuan kami adalah untuk meningkatkan kestabilan robot agar tidak mudah roboh ketika terdapat gangguan eksternal. Proses modifikasi ini merupakan bagian dari upaya kami untuk terus meningkatkan kualitas gerakan robot, memastikan bahwa robot mampu bertahan dan beroperasi secara baik dan kokoh dalam berbagai situasi.

## 5.3 Optimalisasi

* **Kinerja Sistem:** Memastikan bahwa seluruh sistem robot humanoid berjalan dengan lancar, meminimalkan bug atau kesalahan, dan memperbaiki respons sistem terhadap perintah-perintah yang diberikan.
* **Kualitas Gerakan:** Memperbaiki kualitas gerakan robot, termasuk kecepatan, ketepatan, dan keselarasan dalam melakukan gerakan yang diinginkan. Pengoptimalan dilakukan dengan melakukan modifikasi pada program, perangkat keras, atau algoritma kontrol untuk mencapai gerakan yang lebih realistis dan terkoordinasi.
* **Stabilitas dan Keandalan:** Mengatasi masalah kestabilan, khususnya dalam situasi di mana robot rentan roboh atau tidak dapat menjaga keseimbangan dengan baik. Modifikasi dilakukan untuk meningkatkan kemampuan robot dalam beradaptasi dengan perubahan lingkungan atau interaksi dengan luar.
* **Pengalaman Pengguna:** Pengoptimalan tidak hanya terfokus pada kinerja teknis, tetapi juga pada pengalaman pengguna. Ini meliputi penyesuaian pada antarmuka pengguna, pemahaman perilaku robot, serta upaya untuk memastikan interaksi antara manusia dan robot berjalan dengan mulus dan intuitif.

Dengan melakukan pengoptimalan secara terus-menerus, pengembang robot humanoid bertujuan untuk meningkatkan kemampuan robot dalam menjalankan tugasnya dengan lebih baik, lebih aman, dan lebih responsif terhadap lingkungan sekitarnya. Tujuan akhirnya adalah memberikan pengalaman pengguna yang lebih baik dan mengoptimalkan manfaat dari penggunaan robot humanoid di berbagai bidang aplikasi.

# 6 Kolaborasi dan Management Proyek

## 6.1 Dinamika Kerja Tim

Dinamika kerja tim mencakup kolaborasi, peran, dan kontribusi anggota tim dalam mencapai tujuan bersama. Dalam pengembangan robot humanoid, dinamika kerja tim yang baik sangat penting. Ini mencakup:

* **Kolaborasi**: Anggota tim bekerja bersama, berbagi ide, dan keterampilan mereka untuk mencapai tujuan yang telah ditetapkan. Dalam hal pengembangan robot humanoid, mungkin ada beragam spesialisasi, seperti perangkat lunak, perangkat keras, mekanika, dan pemrograman, yang harus bekerja secara terkoordinasi.
* **Peran dan Tanggung Jawab**: Setiap anggota tim memiliki peran dan tanggung jawab yang ditetapkan. Mereka bertanggung jawab atas bagian tertentu dari pengembangan robot, yang memungkinkan pencapaian tujuan secara efisien. Contohnya, ada yang fokus pada pengembangan perangkat lunak, sementara yang lain bekerja pada perancangan mekanika atau pengujian.
* **Kontribusi**: Setiap anggota tim memberikan kontribusi uniknya dalam memecahkan masalah, berkolaborasi, dan memberikan ide atau solusi kreatif yang dapat meningkatkan kualitas dan kinerja robot humanoid. Kontribusi ini diperlukan untuk memastikan bahwa setiap aspek pengembangan robot humanoid dikembangkan secara holistik dan efektif.

## 6.2 Management Proyek

Manajemen proyek melibatkan dokumentasi jadwal proyek, pencapaian, dan praktik manajemen yang diterapkan untuk memastikan proyek berjalan dengan lancar dan sesuai dengan tujuan yang ditetapkan. Dalam konteks pengembangan robot humanoid, manajemen proyek mencakup:

* **Dokumentasi**: Menyusun jadwal proyek yang jelas, termasuk tenggat waktu, tahapan pengembangan, dan target pencapaian. Dokumentasi ini memudahkan pemantauan kemajuan proyek serta membantu identifikasi masalah atau kendala yang mungkin timbul selama proses pengembangan.
* **Pencapaian**: Memantau pencapaian proyek sesuai jadwal yang telah ditetapkan. Ini termasuk mencatat progres, mengevaluasi setiap tahap pengembangan, dan memastikan bahwa tujuan yang ditetapkan berhasil dicapai.
* **Praktik Manajemen**: Mengaplikasikan praktik manajemen yang efektif, seperti identifikasi dan mitigasi risiko, alokasi sumber daya yang tepat, komunikasi yang efisien antar tim, dan adaptasi terhadap perubahan yang mungkin terjadi selama proses pengembangan.

Melalui dinamika kerja tim yang baik dan manajemen proyek yang efektif, pengembangan robot humanoid dapat dilakukan dengan lebih terstruktur, efisien, dan dapat mencapai tujuan yang diinginkan sesuai dengan jadwal yang telah ditetapkan.



Melangkah

Program pertama yang dibuat berupa percobaan melangkah sederhana dengan menggerakan servo yang berada di paha agar dapat melebar. Program masih belum sempurna karena belum ada titik tumpu untuk melangkah sehingga robot bergerak dengan menyeret kaki.

A robot with wires and wires

Description automatically generated

Berjalan

Program ini adalah kelanjutan dari melangkah yang disempurnakan, dengan menambahkan titik tumpu agar robot berada dalam kondisi seimbang di satu kaki maka dapat dilanjutkan dengan melangkahkan kali ke depan seperti Gerakan berjalan. Selanjutnya program ini ditambah dengan bergerak ke arah kesamping kiri dan ke kanan.

A robot on a desk

Description automatically generated

Menendang

Program ini dilanjutkan dari yang sebelumnya menyempurnakan fungui humanoid soccer dengan menambahkan gerakan menendang, hal ini dilakukan dengan menumpukan titik tumpu pada satu kaki dan dilanjutkan dengan gerakan menendang.

# 7 Kesimpulan dan Refleksi

## 7.1 Kesimpulan Proyek

Proyek pengembangan gerakan robot humanoid telah berhasil mengidentifikasi dan mengatasi sejumlah tantangan kunci dalam upaya meningkatkan kemampuan gerak robot. Kami berhasil meningkatkan kualitas gerakan dengan mengoptimalkan algoritma, pengaturan perangkat keras, dan responsivitas sistem. Langkah-langkah pengujian yang sistematis membantu memastikan robot dapat melakukan gerakan secara konsisten dan maksimal dalam berbagai konteks. Hasil dari proyek ini merupakan langkah maju dalam mengembangkan robot humanoid yang lebih adaptif dan responsif.

## 7.2 Proyek Kedepanya

Ada beberapa peningkatan yang dapat dilakukan di masa mendatang. Penggunaan teknologi sensor yang lebih mutakhir, pengembangan algoritma kontrol yang lebih cerdas, dan pemahaman yang lebih dalam mengenai interaksi manusia-robot dapat menjadi fokus pengembangan berikutnya. Aplikasi robot humanoid juga dapat diperluas ke berbagai bidang yang di perlukan pada kehidupan saat ini seperti perawatan kesehatan, edukasi, dan eksplorasi lingkungan yang berbahaya.

## 7.3 Refleksi Pribadi dan Kelompok

Selama proyek ini, kami menghadapi tantangan yang signifikan namun berhasil mengatasinya melalui kolaborasi tim yang solid dan manajemen proyek yang efektif. Kami belajar banyak tentang kompleksitas dalam pengembangan robot humanoid, baik dari segi teknis maupun interaksi manusia-robot. Refleksi pribadi kami menyoroti pentingnya kerjasama, kesabaran, dan dedikasi dalam mencapai tujuan bersama. Secara kolektif, proyek ini memperluas wawasan kami dan mempersiapkan kami untuk tantangan di masa mendatang dalam pengembangan teknologi robotika

# 8 Lampiran

## 8.1 Harga dari Tiap Material

Detail tampilan GUI untuk mengontrol gerakan robot humanoid menggunakan Python mencakup beberapa elemen. Bagian-bagian utama melibatkan antarmuka grafis itu sendiri, tombol kontrol untuk berbagai gerakan dan fungsi, serta tampilan status yang memberikan informasi tentang keadaan robot. Dalam hal biaya, komponen seperti sensor gerak, webcam untuk visualisasi, dan mikrofon untuk interaksi suara dapat mempengaruhi anggaran. Sumber daya untuk belajar Python GUI development, seperti buku, kursus online, dan dokumentasi resmi, dapat menjadi investasi tambahan. Dengan menggabungkan elemen-elemen ini, tampilan GUI dapat memberikan kontrol yang intuitif dan informatif atas gerakan robot humanoid dengan efisiensi biaya yang terkontrol.

## 8.2 Pengkabelan Listrik dan Tata Letak Sistem

* 1. A diagram of a computer

     Description automatically generatedBlock Diagram
  2. A screenshot of a computer

     Description automatically generatedA close-up of a blue circuit board

     Description automatically generatedOpenCM0.04 C-Type
  3. OpenCM9.04 GPIO PinMap

A blue circuit board with many different colors

Description automatically generated

* 1. Example : Open CM9.04 dan AX12 connection

A circuit board with wires connected to it

Description automatically generated

## 8.3 Repositori Kode

Links Repositori HumanoidRobot GUI Control : https://github.com/anh0001/HumanoidRobot-GUI-Control.git

## 8.4 Dokumentasi Tambahan

A computer screen shot of a robot

Description automatically generatedA robot standing on a table

Description automatically generated

# 9 Referensi

Mavrin, Ilya, et al. "Remote control library and GUI development for Russian crawler robot Servosila Engineer." *MATEC Web of Conferences*. Vol. 161. EDP Sciences, 2018.

Zulkefli, M. H., et al. "Graphical user interface (GUI) controlled mobile robot." *Journal of Advanced Research in Computing and Applications* 1.1 (2015): 42-49.

Intisar, Muhatasim, et al. "Computer Vision Based Robotic Arm Controlled Using Interactive GUI." *Intelligent Automation & Soft Computing* 27.2 (2021).

Pollard, Nancy S., et al. "Adapting human motion for the control of a humanoid robot." *Proceedings 2002 IEEE international conference on robotics and automation (Cat. No. 02CH37292)*. Vol. 2. IEEE, 2002.

Safonova, Alla, Nancy Pollard, and Jessica K. Hodgins. "Optimizing human motion for the control of a humanoid robot." *Proc. Applied Mathematics and Applications of Mathematics* 78 (2003): 18-55.