

HUMANOID ROBOT

Abstrak

Proyek kami berfokus pada pengembangan sistem kontrol otomatisasi untuk robot industri, dengan memanfaatkan desain fisik robot yang sudah ada. Tim teknisi kami bertanggung jawab merancang, mengintegrasikan komponen perangkat keras, dan mengembangkan kode pengendali. Dengan fokus pada peningkatan otomasi gerakan robot, kami berhasil mengimplementasikan berbagai perintah dan skenario tugas pada robot industri yang telah ada. Dokumentasi proyek ini mencakup semua langkah-langkah yang kami ambil untuk mencapai solusi desain kontrol otomatisasi yang sukses. Proyek ini memberikan kontribusi berharga dalam pengembangan teknologi otomasi industri.

Imam Tantowi Yahya (4121600038)

[imamtantowiyahya@me.student.pens.ac.id]

Yaafi Ahmad A (4121600056)

[yaafiahmadalf@me.student.pens.ac.id]

RM Nararya R (4121600060)

[nararyarakendrawiratmodjo@me.student.pens.ac.id]

Dalam proyek pengembangan robot humanoid dan antarmuka pengendali GUI, tim teknisi kami melakukan analisis awal, mencari alamat hardware, memilih cara pemrogaman, dan mengembangkan perangkat lunak untuk pengendalian gerakan melalui antarmuka GUI. Integrasi perangkat keras dan lunak dilakukan dengan serangkaian pengujian dan debugging untuk mengoptimalkan kinerja. Hasilnya adalah solusi desain yang sukses, diiringi dokumentasi lengkap untuk memandu pembangunan sistem serupa.

Table of Contents

[1 Introduction and Initial Analysis 2](#_heading=h.gjdgxs)

[1.1 Project Context 2](#_heading=h.30j0zll)

[1.2 Initial Thought Process 2](#_heading=h.1fob9te)

[2 Requirement Analysis and Specification 2](#_heading=h.3znysh7)

[2.1 User Requirements 2](#_heading=h.2et92p0)

[2.2 System Requirements 2](#_heading=h.tyjcwt)

[2.3 Tools and Technologies 2](#_heading=h.3dy6vkm)

[3 Conceptual Design 2](#_heading=h.1t3h5sf)

[3.1 System Architecture 2](#_heading=h.4d34og8)

[3.2 Interface Design 2](#_heading=h.2s8eyo1)

[3.3 Control Algorithm Design 3](#_heading=h.17dp8vu)

[4 Detailed Design and Development 3](#_heading=h.3rdcrjn)

[4.1 Component Design 3](#_heading=h.26in1rg)

[4.2 Coding and Implementation 3](#_heading=h.lnxbz9)

[4.3 Integration 3](#_heading=h.35nkun2)

[4.4 Unique Features 3](#_heading=h.1ksv4uv)

[5 Testing, Evaluation, and Optimization 3](#_heading=h.44sinio)

[5.1 Testing Strategy 3](#_heading=h.2jxsxqh)

[5.2 Performance Evaluation 3](#_heading=h.z337ya)

[5.3 Optimization 3](#_heading=h.3j2qqm3)

[6 Collaboration and Project Management 3](#_heading=h.1y810tw)

[6.1 Teamwork Dynamics 3](#_heading=h.4i7ojhp)

[6.2 Project Management 3](#_heading=h.2xcytpi)

[7 Conclusion and Reflection 3](#_heading=h.1ci93xb)

[7.1 Project Summary 3](#_heading=h.3whwml4)

[7.2 Future Work 3](#_heading=h.2bn6wsx)

[7.3 Personal and Group Reflections 3](#_heading=h.qsh70q)

[8 Appendices 4](#_heading=h.3as4poj)

[8.1 Bill of Materials 4](#_heading=h.1pxezwc)

[8.2 Electrical Wiring and System Layout 4](#_heading=h.49x2ik5)

[8.3 Code Repository 4](#_heading=h.2p2csry)

[8.4 Additional Documentation 4](#_heading=h.147n2zr)

[9 References 4](#_heading=h.3o7alnk)

# 1 Introduction and Initial Analysis

## 1.1 Pendahuluan

Kemajuan teknologi semakin berkembang sangat pesat pada kehidupan manusia pada saat ini, khususnya pada elektronika. Hal ini ditandai dengan adanya berbagai peralatan yang diciptakan dan dapat dioperasikan secara manual maupun otomatis. Kemajuan teknologi inilah maka berkembanglah suatu ilmu yang merupakan suatu pecahan dari ilmu elektronika yaitu dalam bidang robotika. Robot dapat diartikan suatu peralatan yang dioperasikan dengan atau tanpa bantuan manusia atau dengan kata lain bahwa robot merupakan suatu perangkat otomatis. Saat ini robot banyak digunakan dalam dalam industri otomotip dan penyoderan alat elektronik, bahkan dalam beberapa tahun belakang ini diadakan suatu kontes atau perlombaan robot yang bertujuan untuk mengenal maupun memperluas ilmu pengetahuan khususnya dalam bidang ilmu robotika seperti kontes robot yang diadakan di Indonesia

### Rumusan Masalah

Keterbatasan dalam kemampuan gerakan robot humanoid menghambat potensi pemanfaatannya dalam aplikasi yang memerlukan gerakan yang lebih realistis dan kompleks.

### 1.1.2 Tujuan

1. Untuk mengetahui cara penggunaan *Graphical User Interface (GUI)*
2. Untuk membuat gerakan pada robot humanoid
3. untuk melakukan gerakan gerakan seperti berdiri, hormat, squat jump, maju, dan mundur.

## 1.2 Initial Thought Process

Pada proses awal mencari referensi terlebih dahulu terkait dengan gerakan-gerakan robot humanoid, lalu mengimplementasikan dengan menggunakan software Open CM untuk set up ID servo pada robot. Setelah set up ID robot dilanjut dengan melanjutkan mencoba program pada robot, lalu membuat program berupa beberapa gerakan pada robot. Akhirnya kami menggunakan beberapa program gerakan pada robot humanoid yang akan disambungkan dengan GUI.

Adapun tantangan dalam proyek ini ialah:

1. Pertama kali menggunakan robot humanoid
2. Pertama kali menggunakan Graphical User Interface (GUI)
3. membuat program untuk robot humanoid

Peluang dalam proyek ini ialah:

1. Salah satu dari anggota team sudah berpengalaman dalam pemrograman robot
2. Mempunyai device yang mumpuni dalam proyek ini

# 2 Requirement Analysis and Specification

## 2.1 User Requirements

Graphical User Interface(GUI) atau Antarmuka Pengguna Grafis merujuk pada suatu bentuk antarmuka pengguna yang memungkinkan interaksi antara pengguna dan perangkat lunak atau sistem menggunakan elemen-elemen grafis seperti ikon, tombol, jendela, dan menu. GUI dirancang untuk dapat berinteraksi dengan perangkat lunak atau sistem dengan lebih mudah dan intuitif, tanpa perlu mengingat perintah-perintah teks atau sintaks khusus. GUI menggunakan metode interaksi pada peranti elektronik secara grafis (bukan perintah teks) antara pengguna dan komputer.

Robot ini digunakan dan dioperasikan oleh mahasiswa sebagai modul pembelajaran dalam mata kuliah pemrograman sistem mekatronik. GUI akan disambungkan dengan program yang dibuat, GUI akan menampilkan data yang telah disambungkan, lalu GUI dapat menjadi kontroller dari robot humanoid.

## 2.2 System Requirements

Spesifikasi robot humanoid:

* Sensor IMU
* OpenCM robotis
* Dynamixel Ax 12 A

Koneksivitas robot dengan GUI akan menggunakan sambungan koneksi kabel usb type B dan didukung dengan power 220 V melalui power adaptor.

Untuk pergerakan robot humanoid ini nantinya dapat menampilkan berjalan, menampilkan gerakan bebas, menampilkan opsi untuk gerakan menarik yang dibuat. Yang nantinya gerakan-gerakan ini ditampilkan pada GUI untuk mendapatkan nilai servo, tombol untuk menentukan gerakan menarik, dan slider untuk menggerakan servo-servo dari robot humanoid.

## 2.3 Tools and Technologies

* Robot humanoid dengan spesifikasi:

1. Sensor IMU
2. OpenCM robotis
3. Dynamixel Ax 12A

* Aplikasi Qt Designer
* Aplikasi OpenCM
* Aplikasi Arduino IDE

|  |  |  |
| --- | --- | --- |
| A cartoon of a grey cat  Description automatically generated  Aplikasi OpenCM | Aplikasi Arduino IDE | Aplikasi Qt Designer |
| Dynamixel Ax 12A | OpenCM | Sensor IMU |

## 2.4 Target specification

|  |  |  |  |
| --- | --- | --- | --- |
| **Fitur** | **Keterangan** | **Metrik Pengukuran** | **Nilai capaian** |
| **Gerakan 1** | Robot jalan | Tidak ada feedback yang diberikan robot dikarenakan menggunakan metode open loop progam. | Robot dapat bergerak jalan sesuai perintah yang diberikan dari user. |
| **Gerakan 2** | Robot berdiri | Tidak ada feedback yang diberikan robot dikarenakan menggunakan metode open loop progam. | Robot dapat berdiri sesuai perintah yang diberikan dari user. |
| **Gerakan 3** | Robot hormat | Tidak ada feedback yang diberikan robot dikarenakan menggunakan metode open loop progam. | Robot dapat hormat sesuai perintah yang diberikan dari user. |
| **Gerakan 4** | Robot squat | Tidak ada feedback yang diberikan robot dikarenakan menggunakan metode open loop progam. | Robot dapat squat sesuai perintah yang diberikan dari user. |
| **Gerakan 5** | Robot melambai | Tidak ada feedback yang diberikan robot dikarenakan menggunakan metode open loop progam. | Robot dapat melambai sesuai perintah yang diberikan dari user. |

*Tabel 1. Tabel caption.*

# 3 Conceptual Design

## 3.1 System Architecture

## Antarmuka Pengguna (GUI)

## Pada bagian ini, merupakan bagian yang menghubungkan antara pengguna dan sistem, dimana GUI dikembangkan dengan desain dari QT Designer dan memungkinkan pengguna untuk memberikan input dan menerima data dari sistem. Dalam projek ini, GUI didesain untuk dapat memantau nilai ketiga sensor ultrasonik, data sensor garis, dan data dari IMU, serta dapat memberikan input nilai PID pada sistem dan input jarak stop dari sensor ultrasonik.

## • Logika Kontrol

## Logika kontrol dari sistem ini adalah robot bisa diimplementasikan sekitar 30% untuk sistem deteksi atau menghindar dari halangan yang ada disekitarnya.

## • Komunikasi antar komponen

## Komunikasi antara ESP 01 dan GUI dapat dilakukan melalui protokol komunikasi yang sesuai dengan kebutuhan aplikasi, seperti HTTP/HTTPS, MQTT, WebSocket, RESTful API, dan Serial Communication

## • Perangkat keras (Hardware)

## Pada bagian ini, perangkat keras yang digunakan yaitu arduino nano, ESP 01, Sensor ultrasonik HC-SR04, line sensor, IMU, serta motor dan driver.

## • Jaringan

## Koneksi jaringan yang digunakan adalah menggunakan koneksi wireless berupa wifi antara GUI dan dan ESP 01

## 3.2 Interface Design

|  |  |  |
| --- | --- | --- |
| NO | Gambar Sketsa | Keterangan |
| 1 |  | * Fungsi : Keterangan * untuk mempermudah pengguna dan juga untuk memberi laber pada GUI |
| 2 |  | * Fungsi : Tombol * Memicu Aksi (Action Trigger): Tombol digunakan untuk memicu aksi atau perintah berdiri ketika diklik. |
| 3 |  | * Fungsi : Tombol * Memicu Aksi (Action Trigger): Tombol digunakan untuk memicu aksi atau perintah hormat ketika diklik |
| 4 |  | * Fungsi : Tombol * Memicu Aksi (Action Trigger): Tombol digunakan untuk memicu aksi atau perintah jalan ketika diklik |
| 5 |  | * Fungsi : Tombol * Memicu Aksi (Action Trigger): Tombol digunakan untuk memicu aksi atau perintah lambai ketika diklik |
| 6 |  | * Fungsi : Tombol * Memicu Aksi (Action Trigger): Tombol digunakan untuk memicu aksi atau perintah squat ketika diklik |
| 7 |  | * Fungsi : Parameter * Display seberapa besar derajat servo disetiap dof robot |

## 3.3 Control Algorithm Design

# **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 Detailed Design and Development

## 4.1 Component Design

Delve into the design of individual components, modules, and functionalities.

* OpenCM9.04
* Servo Dynamixel AX-12A : 6
* Servo Dynamixel AX-18A : 10

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

## 4.2 Coding and Implementation

#define DXL\_BUS\_SERIAL1 1 //Dynamixel on Serial1(USART1) <-OpenCM9.04

#define DXL\_BUS\_SERIAL2 2 //Dynamixel on Serial2(USART2) <-LN101,BT210

#define DXL\_BUS\_SERIAL3 3 //Dynamixel on Serial3(USART3) <-OpenCM 485EXP

#define ID\_NUM 3

//ID 1 bahu kanan

//ID 2 bahu kiri

//ID 3 lengan atas kanan

//ID 4 lengan atas kiri

//ID 5 lengan bawah kanan

//ID 6 lengan bawah kiri

//ID 9 pinggul kanan

//ID 10 pinggul kiri

//ID 11 paha kanan

//ID 12 paha kiri

//ID 13 lutut kanan

//ID 14 lutut kiri

//ID 15 tumit kanan

//ID 16 tumit kiri

//ID 17 telapak kanan

//ID 18 telapak kiri

Dynamixel Dxl(DXL\_BUS\_SERIAL3);

void setup() {

// Initialize the dynamixel bus:

// Dynamixel 2.0 Baudrate -> 0: 9600, 1: 57600, 2: 115200, 3: 1Mbps

Dxl.begin(3);

for (int i = 1; i<=20; i++) {

Dxl.jointMode(i);

}

berdiri();

delay(3000);

}

void berdiri(){

Dxl.setPosition(1, 512, 50); //ID 1 bahu kanan

Dxl.setPosition(2, 512, 50); //ID 2 bahu kiri

Dxl.setPosition(3, 512, 50); //ID 3 lengan atas kanan

Dxl.setPosition(4, 512, 50); //ID 4 lengan atas kiri

Dxl.setPosition(5, 512, 50); //ID 5 lengan bawah kanan

Dxl.setPosition(6, 512, 50); //ID 6 lengan bawah kiri

delay(500);

Dxl.setPosition(9, 532, 50); //ID 9 pinggul kanan

Dxl.setPosition(10, 492, 50); //ID 10 pinggul kiri

//===================================================

Dxl.setPosition(11, 672, 50); //ID 11 paha kanan (-- mundur)

Dxl.setPosition(12, 352, 50); //ID 12 paha kiri (-- maju)

Dxl.setPosition(13, 752, 50); //ID 13 lutut kanan (++ mundur), tidak boleh <512

Dxl.setPosition(14, 272, 50); //ID 14 lutut kiri (tidak boleh >512)

Dxl.setPosition(15, 612, 50); //ID 15 tumit kanan (++ maju)

Dxl.setPosition(16, 412, 50); //ID 16 tumit kiri (-- maju)

Dxl.setPosition(17, 532, 50); //ID 17 telapak kanan (++ miring kiri)

Dxl.setPosition(18, 492, 50); //ID 18 telapak kiri (++ miring kiri)

delay(500);

}

void lambai(){

Dxl.setPosition(1, 852, 50); //ID 1 bahu kanan

Dxl.setPosition(2, 512, 50); //ID 2 bahu kiri

Dxl.setPosition(3, 822, 50); //ID 3 lengan atas kanan

Dxl.setPosition(4, 512, 50); //ID 4 lengan atas kiri

Dxl.setPosition(5, 712, 200); //ID 5 lengan bawah kanan

Dxl.setPosition(6, 512, 50); //ID 6 lengan bawah kiri

delay(500);

Dxl.setPosition(9, 532, 50); //ID 9 pinggul kanan

Dxl.setPosition(10, 492, 50); //ID 10 pinggul kiri

//===================================================

Dxl.setPosition(11, 672, 50); //ID 11 paha kanan (-- mundur)

Dxl.setPosition(12, 352, 50); //ID 12 paha kiri (-- maju)

Dxl.setPosition(13, 752, 50); //ID 13 lutut kanan (++ mundur), tidak boleh <512

Dxl.setPosition(14, 272, 50); //ID 14 lutut kiri (tidak boleh >512)

Dxl.setPosition(15, 612, 50); //ID 15 tumit kanan (++ maju)

Dxl.setPosition(16, 412, 50); //ID 16 tumit kiri (-- maju)

Dxl.setPosition(13, 752, 50); //ID 13 lutut kanan (++ mundur), tidak boleh <512

Dxl.setPosition(14, 272, 50); //ID 14 lutut kiri (tidak boleh >512)

Dxl.setPosition(15, 612, 50); //ID 15 tumit kanan (++ maju)

Dxl.setPosition(16, 412, 50); //ID 16 tumit kiri (-- maju)

Dxl.setPosition(17, 532, 50); //ID 17 telapak kanan (++ miring kiri)

Dxl.setPosition(18, 492, 50); //ID 18 telapak kiri (++ miring kiri)

delay(500);

}void lambai2(){

Dxl.setPosition(5, 412, 200); //ID 5 lengan bawah kanan

delay(800);

}

void hormat(){

Dxl.setPosition(1, 672, 60); //ID 1 bahu kanan

Dxl.setPosition(2, 512, 50); //ID 2 bahu kiri

Dxl.setPosition(3, 722, 70); //ID 3 lengan atas kanan

Dxl.setPosition(4, 512, 50); //ID 4 lengan atas kiri

Dxl.setPosition(5, 752, 50); //ID 5 lengan bawah kanan

Dxl.setPosition(6, 512, 50); //ID 6 lengan bawah kiri

delay(500);

}

void squad(){

//posisi awal

Dxl.setPosition(1, 842, 100); //ID 1 bahu kanan

Dxl.setPosition(2, 182, 100); //ID 2 bahu kiri

Dxl.setPosition(3, 512, 100); //ID 3 lengan atas kanan

Dxl.setPosition(4, 512, 100); //ID 4 lengan atas kiri

Dxl.setPosition(5, 192, 100); //ID 5 lengan bawah kanan

Dxl.setPosition(6, 832, 100); //ID 6 lengan bawah kiri

Dxl.setPosition(9, 532, 50); //ID 9 pinggul kanan

Dxl.setPosition(10, 492, 50); //ID 10 pinggul kiri

//==========================================================

Dxl.setPosition(11, 712, 50); //ID 11 paha kanan (-- mundur)

Dxl.setPosition(12, 322, 50); //ID 12 paha kiri (-- maju)

Dxl.setPosition(13, 802, 50); //ID 13 lutut kanan (++ mundur), tidak boleh <512

Dxl.setPosition(14, 222, 50); //ID 14 lutut kiri (tidak boleh >512)

Dxl.setPosition(15, 612, 50); //ID 15 tumit kanan (++ maju)

Dxl.setPosition(16, 412, 50); //ID 16 tumit kiri (-- maju)

Dxl.setPosition(17, 532, 50); //ID 15 telapak kanan (++ miring kiri)

Dxl.setPosition(18, 492, 50); //ID 16 telapak kiri (++ miring kiri)

delay(1000);

//posisi jongkok

Dxl.setPosition(1, 842, 100); //ID 1 bahu kanan

Dxl.setPosition(2, 182, 100); //ID 2 bahu kiri

Dxl.setPosition(3, 512, 100); //ID 3 lengan atas kanan

Dxl.setPosition(4, 512, 100); //ID 4 lengan atas kiri

Dxl.setPosition(5, 192, 100); //ID 5 lengan bawah kanan

Dxl.setPosition(6, 832, 100); //ID 6 lengan bawah kiri

//==========================================================

Dxl.setPosition(9, 532, 50); //ID 9 pinggul kanan

Dxl.setPosition(10,492, 50); //ID 10 pinggul kiri

//==========================================================

Dxl.setPosition(11, 752, 70); //ID 11 paha kanan (-- mundur)

Dxl.setPosition(12, 292, 70); //ID 12 paha kiri (-- maju)

Dxl.setPosition(13, 822, 70); //ID 13 lutut kanan (++ mundur), tidak boleh <512

Dxl.setPosition(14, 202, 70); //ID 14 lutut kiri (tidak boleh >512)

Dxl.setPosition(15, 612, 50); //ID 15 tumit kanan (++ maju)

Dxl.setPosition(16, 412, 50); //ID 16 tumit kiri (-- maju)

Dxl.setPosition(17, 532, 50); //ID 15 telapak kanan (++ miring kiri)

Dxl.setPosition(18, 492, 50); //ID 16 telapak kiri (++ miring kiri)

delay(1000);

//posisi berdiri

Dxl.setPosition(1, 512, 70); //ID 1 bahu kanan

Dxl.setPosition(2, 512, 70); //ID 2 bahu kiri

Dxl.setPosition(3, 512, 70); //ID 3 lengan atas kanan

Dxl.setPosition(4, 512, 70); //ID 4 lengan atas kiri

Dxl.setPosition(5, 512, 70); //ID 5 lengan bawah kanan

Dxl.setPosition(6, 512, 70); //ID 6 lengan bawah kiri

delay(500);

Dxl.setPosition(9, 532, 50); //ID 9 pinggul kanan

Dxl.setPosition(10, 492, 50); //ID 10 pinggul kiri

//===================================================

Dxl.setPosition(11, 672, 50); //ID 11 paha kanan (-- mundur)

Dxl.setPosition(12, 352, 50); //ID 12 paha kiri (-- maju)

Dxl.setPosition(13, 752, 50); //ID 13 lutut kanan (++ mundur), tidak boleh <512

Dxl.setPosition(14, 272, 50); //ID 14 lutut kiri (tidak boleh >512)

Dxl.setPosition(15, 612, 50); //ID 15 tumit kanan (++ maju)

Dxl.setPosition(16, 412, 50); //ID 16 tumit kiri (-- maju)

Dxl.setPosition(17, 532, 50); //ID 17 telapak kanan (++ miring kiri)

Dxl.setPosition(18, 492, 50); //ID 18 telapak kiri (++ miring kiri)

delay(2000);

}

void jalanA(){

//setup

delay(250);

Dxl.setPosition(13, 622, 50); //ID 13 lutut kanan (++ mundur), tidak boleh <512

Dxl.setPosition(14, 302, 50); //ID 14 lutut kiri (++ mundur), (tidak boleh >512)

Dxl.setPosition(17, 572, 50); //ID 17 telapak kanan (++ miring kiri)

Dxl.setPosition(18, 552, 50); //ID 18 telapak kiri (++ miring kiri)

Dxl.setPosition(15, 612, 50); //ID 15 tumit kanan (++ maju)

Dxl.setPosition(16, 412, 50); //ID 16 tumit kiri (-- maju)

delay(500);

Dxl.setPosition(13, 622, 50); //ID 13 lutut kanan (++ mundur), (tidak boleh >512)

Dxl.setPosition(14, 302, 50); //ID 14 lutut kiri (++ mundur), (tidak boleh >512)

Dxl.setPosition(15, 412, 50); //ID 15 tumit kanan (++ maju)

Dxl.setPosition(16, 412, 50); //ID 16 tumit kiri (-- maju)

delay(500);

//paha kanan maju

Dxl.setPosition(11, 702, 50); //ID 11 paha kanan (-- mundur)

Dxl.setPosition(12, 352, 50); //ID 12 paha kiri (-- maju)

Dxl.setPosition(15, 512, 100); //ID 15 tumit kanan (++ maju)

Dxl.setPosition(16, 412, 100); //ID 16 tumit kiri (-- maju)

delay(500);

//menapak kanan normal

Dxl.setPosition(17, 522, 50); //ID 17 telapak kanan (++ miring kiri)

Dxl.setPosition(18, 502, 50); //ID 18 telapak kiri (++ miring kiri)

Dxl.setPosition(15, 532, 100); //ID 15 tumit kanan (++ maju)

Dxl.setPosition(16, 392, 100); //ID 16 tumit kiri (-- maju)

delay(500);

//keseimbangan

Dxl.setPosition(11, 722, 25); //ID 11 paha kanan (-- mundur)

Dxl.setPosition(12, 302, 25); //ID 12 paha kiri (-- maju)

delay(800);

Dxl.setPosition(17, 472, 50); //ID 17 telapak kanan (++ miring kiri)

Dxl.setPosition(18, 432, 50); //ID 18 telapak kiri (++ miring kiri)

Dxl.setPosition(15, 532, 150); //ID 15 tumit kanan (++ maju)

Dxl.setPosition(16, 582, 150); //ID 16 tumit kiri (-- maju)

delay(500);

Dxl.setPosition(15, 532, 150); //ID 15 tumit kanan (++ maju)

Dxl.setPosition(16, 492, 150); //ID 16 tumit kiri (-- maju)

delay(500);

Dxl.setPosition(13, 622, 50); //ID 13 lutut kanan (++ mundur), (tidak boleh >512)

Dxl.setPosition(14, 402, 100); //ID 14 lutut kiri (++ mundur), (tidak boleh >512)

delay(500);

//menapak kaki kiri

Dxl.setPosition(17, 532, 50); //ID 17 telapak kanan (++ miring kiri)

Dxl.setPosition(18, 492, 50); //ID 18 telapak kiri (++ miring kiri)

delay(500);

Dxl.setPosition(11, 672, 50); //ID 11 paha kanan (-- mundur)

Dxl.setPosition(12, 352, 50); //ID 12 paha kiri (-- maju)

Dxl.setPosition(13, 622, 50); //ID 13 lutut kanan (++ mundur), (tidak boleh >512)

Dxl.setPosition(14, 402, 50); //ID 14 lutut kiri (++ mundur), (tidak boleh >512)

Dxl.setPosition(15, 532, 100); //ID 15 tumit kanan (++ maju)

Dxl.setPosition(16, 502, 100); //ID 16 tumit kiri (-- maju)

}

void loop() {

char led = 0;

squad();

delay(1000);

//berdiri();

//delay(500);

//lambai();

//delay(200);

//lambai2();

//delay(200);

//jalanA();

//delay(1000);

}

if (SerialUSB.available())

{

char Cmd = SerialUSB.read();

if(Cmd == 'B'){

berdiri();

delay(3000);

}else if(Cmd == 'H'){

hormat();

delay(1000);

berdiri();

}else if(Cmd == 'S'){

squad();

delay(2000);

berdiri();

}else if(Cmd == 'J'){

jalanA();

delay(250);

berdiri();

}else if(Cmd == 'L'){

lambai();

delay(1000);

lambai2();

delay(1000);

berdiri();

}

}

}

## 4.3 Integration

# -\*- coding: utf-8 -\*-

# Form implementation generated from reading ui file 'NEW\_HUMANOID.ui'

# Created by: PyQt5 UI code generator 5.15.10

# WARNING: Any manual changes made to this file will be lost when pyuic5 is

# run again. Do not edit this file unless you know what you are doing.

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui\_MainWindow(object):

def setupUi(self, MainWindow):

MainWindow.setObjectName("MainWindow")

MainWindow.resize(897, 680)

MainWindow.setStyleSheet("background-color: rgb(255, 255, 255);")

self.centralwidget = QtWidgets.QWidget(MainWindow)

self.centralwidget.setObjectName("centralwidget")

self.label = QtWidgets.QLabel(self.centralwidget)

self.label.setGeometry(QtCore.QRect(0, 0, 61, 71))

self.label.setText("")

self.label.setPixmap(QtGui.QPixmap("icons8-robot-64.png"))

self.label.setObjectName("label")

self.label\_2 = QtWidgets.QLabel(self.centralwidget)

self.label\_2.setGeometry(QtCore.QRect(60, 30, 351, 41))

font = QtGui.QFont()

font.setFamily("Monotxt\_IV50")

font.setPointSize(26)

self.label\_2.setFont(font)

self.label\_2.setObjectName("label\_2")

self.label\_3 = QtWidgets.QLabel(self.centralwidget)

self.label\_3 = QtWidgets.QLabel(self.centralwidget)

.

## 

self.label\_3.setGeometry(QtCore.QRect(60, 0, 311, 31))

font = QtGui.QFont()

font.setFamily("ROG Fonts")

font.setPointSize(12)

font.setUnderline(True)

self.label\_3.setFont(font)

self.label\_3.setObjectName("label\_3")

self.frame = QtWidgets.QFrame(self.centralwidget)

self.frame.setGeometry(QtCore.QRect(20, 170, 361, 111))

self.frame.setFrameShape(QtWidgets.QFrame.StyledPanel)

self.frame.setFrameShadow(QtWidgets.QFrame.Raised)

self.frame.setObjectName("frame")

self.horizontalLayout = QtWidgets.QHBoxLayout(self.frame)

self.horizontalLayout.setContentsMargins(9, 9, -1, -1)

self.horizontalLayout.setSpacing(30)

self.horizontalLayout.setObjectName("horizontalLayout")

self.pushButton = QtWidgets.QPushButton(self.frame)

self.pushButton.setStyleSheet("")

self.pushButton.setText("")

icon = QtGui.QIcon()

icon.addPixmap(QtGui.QPixmap("icons8-stand-96.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

self.pushButton.setIcon(icon)

self.pushButton.setIconSize(QtCore.QSize(100, 100))

self.pushButton.setObjectName("pushButton")

self.horizontalLayout.addWidget(self.pushButton)

self.pushButton\_3 = QtWidgets.QPushButton(self.frame)

self.pushButton\_3.setText("")

icon1 = QtGui.QIcon()

icon1.addPixmap(QtGui.QPixmap("icons8-saluting-face-94.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

icon3 = QtGui.QIcon()

icon3.addPixmap(QtGui.QPixmap("icons8-waving-hand-light-skin-tone-96.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

self.pushButton\_2.setIcon(icon3)

self.pushButton\_2.setIconSize(QtCore.QSize(80, 80))

self.pushButton\_2.setObjectName("pushButton\_2")

self.horizontalLayout\_2.addWidget(self.pushButton\_2)

self.pushButton\_4 = QtWidgets.QPushButton(self.frame\_2)

self.pushButton\_4.setText("")

icon4 = QtGui.QIcon()

icon4.addPixmap(QtGui.QPixmap("icons8-squat-96.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

self.pushButton\_4.setIcon(icon4)

self.pushButton\_4.setIconSize(QtCore.QSize(80, 80))

self.pushButton\_4.setObjectName("pushButton\_4")

self.horizontalLayout\_2.addWidget(self.pushButton\_4)

self.label\_4 = QtWidgets.QLabel(self.centralwidget)

self.label\_4.setGeometry(QtCore.QRect(40, 150, 81, 16))

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_4.setFont(font)

self.label\_4.setObjectName("label\_4")

self.label\_5 = QtWidgets.QLabel(self.centralwidget)

self.label\_5.setGeometry(QtCore.QRect(100, 320, 71, 16))

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_5.setFont(font)

self.label\_5.setObjectName("label\_5")

self.label\_6 = QtWidgets.QLabel(self.centralwidget)

self.label\_6.setGeometry(QtCore.QRect(300, 150, 61, 16))

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_6.setFont(font)

self.label\_6.setObjectName("label\_6")

self.label\_8 = QtWidgets.QLabel(self.centralwidget)

self.label\_8.setGeometry(QtCore.QRect(160, 150, 81, 16))

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_8.setFont(font)

self.label\_8.setObjectName("label\_8")

self.label\_9 = QtWidgets.QLabel(self.centralwidget)

self.label\_9.setGeometry(QtCore.QRect(230, 320, 61, 16))

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_9.setFont(font)

self.label\_9.setObjectName("label\_9")

self.label\_10 = QtWidgets.QLabel(self.centralwidget)

self.label\_10.setGeometry(QtCore.QRect(150, 100, 101, 31))

font = QtGui.QFont()

font.setPointSize(22)

font.setBold(True)

font.setWeight(75)

self.label\_10.setFont(font)

self.label\_10.setObjectName("label\_10")

self.label\_13 = QtWidgets.QLabel(self.centralwidget)

self.label\_13.setGeometry(QtCore.QRect(510, 10, 291, 551))

self.label\_13.setStyleSheet("image: url(:/newPrefix/Pictures/Screenshots/Screenshot 2023-11-16 104400.png);")

self.label\_13.setText("")

self.label\_13.setPixmap(QtGui.QPixmap("Screenshot 2023-11-16 104400.png"))

self.label\_13.setObjectName("label\_13")

self.lcdNumber = QtWidgets.QLCDNumber(self.centralwidget)

self.lcdNumber.setGeometry(QtCore.QRect(550, 20, 64, 23))

self.lcdNumber.setObjectName("lcdNumber")

self.lcdNumber\_2 = QtWidgets.QLCDNumber(self.centralwidget)

self.lcdNumber\_2.setGeometry(QtCore.QRect(700, 20, 64, 23))

self.lcdNumber\_2.setObjectName("lcdNumber\_2")

self.lcdNumber\_3 = QtWidgets.QLCDNumber(self.centralwidget)

self.lcdNumber\_3.setGeometry(QtCore.QRect(820, 130, 64, 23))

self.lcdNumber\_3.setObjectName("lcdNumber\_3")

self.lcdNumber\_4 = QtWidgets.QLCDNumber(self.centralwidget)

self.lcdNumber\_4.setGeometry(QtCore.QRect(820, 190, 64, 23))

self.lcdNumber\_4.setObjectName("lcdNumber\_4")

self.lcdNumber\_5 = QtWidgets.QLCDNumber(self.centralwidget)

self.lcdNumber\_5.setGeometry(QtCore.QRect(810, 230, 64, 23))

self.lcdNumber\_5.setObjectName("lcdNumber\_5")

self.lcdNumber\_6 = QtWidgets.QLCDNumber(self.centralwidget)

self.lcdNumber\_6.setGeometry(QtCore.QRect(760, 280, 64, 23))

self.lcdNumber\_6.setObjectName("lcdNumber\_6")

self.lcdNumber\_7 = QtWidgets.QLCDNumber(self.centralwidget)

self.lcdNumber\_7.setGeometry(QtCore.QRect(760, 370, 64, 23))

self.lcdNumber\_7.setObjectName("lcdNumber\_7")

self.lcdNumber\_8 = QtWidgets.QLCDNumber(self.centralwidget)

self.lcdNumber\_8.setGeometry(QtCore.QRect(760, 440, 64, 23))

self.lcdNumber\_8.setObjectName("lcdNumber\_8")

self.lcdNumber\_9 = QtWidgets.QLCDNumber(self.centralwidget)

self.lcdNumber\_9.setGeometry(QtCore.QRect(663, 570, 71, 23))

self.lcdNumber\_9.setObjectName("lcdNumber\_9")

self.lcdNumber\_10 = QtWidgets.QLCDNumber(self.centralwidget)

self.lcdNumber\_10.setGeometry(QtCore.QRect(580, 570, 64, 23))

self.lcdNumber\_10.setObjectName("lcdNumber\_10")

self.lcdNumber\_11 = QtWidgets.QLCDNumber(self.centralwidget)

self.lcdNumber\_11.setGeometry(QtCore.QRect(490, 440, 64, 23))

self.lcdNumber\_11.setObjectName("lcdNumber\_11")

self.lcdNumber\_12 = QtWidgets.QLCDNumber(self.centralwidget)

self.lcdNumber\_12.setGeometry(QtCore.QRect(490, 370, 64, 23))

self.lcdNumber\_12.setObjectName("lcdNumber\_12")

self.lcdNumber\_13 = QtWidgets.QLCDNumber(self.centralwidget)

self.lcdNumber\_13.setGeometry(QtCore.QRect(490, 280, 64, 23))

self.lcdNumber\_13.setObjectName("lcdNumber\_13")

self.lcdNumber\_14 = QtWidgets.QLCDNumber(self.centralwidget)

self.lcdNumber\_14.setGeometry(QtCore.QRect(430, 230, 64, 23))

self.lcdNumber\_14.setObjectName("lcdNumber\_14")

self.lcdNumber\_15 = QtWidgets.QLCDNumber(self.centralwidget)

self.lcdNumber\_15.setGeometry(QtCore.QRect(420, 190, 64, 23))

self.lcdNumber\_15.setObjectName("lcdNumber\_15")

self.lcdNumber\_16 = QtWidgets.QLCDNumber(self.centralwidget)

self.lcdNumber\_16.setGeometry(QtCore.QRect(430, 130, 64, 23))

self.lcdNumber\_16.setObjectName("lcdNumber\_16")

self.line = QtWidgets.QFrame(self.centralwidget)

self.line.setGeometry(QtCore.QRect(740, 440, 16, 16))

self.line.setFrameShape(QtWidgets.QFrame.HLine)

self.line.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line.setObjectName("line")

self.line\_2 = QtWidgets.QFrame(self.centralwidget)

self.line\_2.setGeometry(QtCore.QRect(560, 440, 16, 16))

self.line\_2.setFrameShape(QtWidgets.QFrame.HLine)

self.line\_2.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line\_2.setObjectName("line\_2")

self.line\_3 = QtWidgets.QFrame(self.centralwidget)

self.line\_3.setGeometry(QtCore.QRect(740, 370, 16, 16))

self.line\_3.setFrameShape(QtWidgets.QFrame.HLine)

self.line\_3.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line\_3.setObjectName("line\_3")

self.line\_4 = QtWidgets.QFrame(self.centralwidget)

self.line\_4.setGeometry(QtCore.QRect(560, 370, 16, 16))

self.line\_4.setFrameShape(QtWidgets.QFrame.HLine)

self.line\_4.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line\_4.setObjectName("line\_4")

self.line\_5 = QtWidgets.QFrame(self.centralwidget)

self.line\_5.setGeometry(QtCore.QRect(740, 280, 16, 16))

self.line\_5.setFrameShape(QtWidgets.QFrame.HLine)

self.line\_5.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line\_5.setObjectName("line\_5")

self.line\_6 = QtWidgets.QFrame(self.centralwidget)

self.line\_6.setGeometry(QtCore.QRect(560, 280, 16, 16))

self.line\_6.setFrameShape(QtWidgets.QFrame.HLine)

self.line\_6.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line\_6.setObjectName("line\_6")

self.line\_7 = QtWidgets.QFrame(self.centralwidget)

self.line\_7.setGeometry(QtCore.QRect(800, 190, 16, 16))

self.line\_7.setFrameShape(QtWidgets.QFrame.HLine)

self.line\_7.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line\_7.setObjectName("line\_7")

self.line\_8 = QtWidgets.QFrame(self.centralwidget)

self.line\_8.setGeometry(QtCore.QRect(490, 190, 16, 16))

self.line\_8.setFrameShape(QtWidgets.QFrame.HLine)

self.line\_8.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line\_8.setObjectName("line\_8")

self.line\_9 = QtWidgets.QFrame(self.centralwidget)

self.line\_9.setGeometry(QtCore.QRect(800, 130, 16, 16))

self.line\_9.setFrameShape(QtWidgets.QFrame.HLine)

self.line\_9.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line\_9.setObjectName("line\_9")

self.line\_10 = QtWidgets.QFrame(self.centralwidget)

self.line\_10.setGeometry(QtCore.QRect(500, 130, 16, 16))

self.line\_10.setFrameShape(QtWidgets.QFrame.HLine)

self.line\_10.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line\_10.setObjectName("line\_10")

self.line\_11 = QtWidgets.QFrame(self.centralwidget)

self.line\_11.setGeometry(QtCore.QRect(710, 50, 20, 16))

self.line\_11.setFrameShape(QtWidgets.QFrame.VLine)

self.line\_11.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line\_11.setObjectName("line\_11")

self.line\_12 = QtWidgets.QFrame(self.centralwidget)

self.line\_12.setGeometry(QtCore.QRect(570, 50, 20, 16))

self.line\_12.setFrameShape(QtWidgets.QFrame.VLine)

self.line\_12.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line\_12.setObjectName("line\_12")

self.line\_13 = QtWidgets.QFrame(self.centralwidget)

self.line\_13.setGeometry(QtCore.QRect(610, 550, 20, 16))

self.line\_13.setFrameShape(QtWidgets.QFrame.VLine)

self.line\_13.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line\_13.setObjectName("line\_13")

self.line\_14 = QtWidgets.QFrame(self.centralwidget)

self.line\_14.setGeometry(QtCore.QRect(690, 550, 20, 16))

self.line\_14.setFrameShape(QtWidgets.QFrame.VLine)

self.line\_14.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line\_14.setObjectName("line\_14")

self.label\_11 = QtWidgets.QLabel(self.centralwidget)

self.label\_11.setGeometry(QtCore.QRect(580, 600, 71, 16))

font = QtGui.QFont()

font.setPointSize(9)

font.setBold(True)

font.setWeight(75)

self.label\_11.setFont(font)

self.label\_11.setObjectName("label\_11")

self.label\_14 = QtWidgets.QLabel(self.centralwidget)

self.label\_14.setGeometry(QtCore.QRect(670, 600, 71, 16))

font = QtGui.QFont()

font.setPointSize(9)

font.setBold(True)

font.setWeight(75)

self.label\_14.setFont(font)

self.label\_14.setObjectName("label\_14")

self.label\_15 = QtWidgets.QLabel(self.centralwidget)

self.label\_15.setGeometry(QtCore.QRect(590, 200, 16, 16))

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_15.setFont(font)

self.label\_15.setObjectName("label\_15")

self.label\_16 = QtWidgets.QLabel(self.centralwidget)

self.label\_16.setGeometry(QtCore.QRect(700, 200, 16, 16))

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_16.setFont(font)

self.label\_16.setObjectName("label\_16")

self.label\_17 = QtWidgets.QLabel(self.centralwidget)

self.label\_17.setGeometry(QtCore.QRect(410, 230, 16, 16))

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_17.setFont(font)

self.label\_17.setObjectName("label\_17")

self.label\_18 = QtWidgets.QLabel(self.centralwidget)

self.label\_18.setGeometry(QtCore.QRect(880, 230, 16, 16))

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_18.setFont(font)

self.label\_18.setObjectName("label\_18")

self.line\_15 = QtWidgets.QFrame(self.centralwidget)

self.line\_15.setGeometry(QtCore.QRect(10, 130, 381, 16))

self.line\_15.setFrameShape(QtWidgets.QFrame.HLine)

self.line\_15.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line\_15.setObjectName("line\_15")

self.line\_16 = QtWidgets.QFrame(self.centralwidget)

self.line\_16.setGeometry(QtCore.QRect(10, 450, 381, 16))

self.line\_16.setFrameShape(QtWidgets.QFrame.HLine)

self.line\_16.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line\_16.setObjectName("line\_16")

self.line\_17 = QtWidgets.QFrame(self.centralwidget)

self.line\_17.setGeometry(QtCore.QRect(10, 290, 381, 16))

self.line\_17.setFrameShape(QtWidgets.QFrame.HLine)

self.line\_17.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line\_17.setObjectName("line\_17")

self.label\_7 = QtWidgets.QLabel(self.centralwidget)

self.label\_7.setGeometry(QtCore.QRect(20, 480, 251, 31))

font = QtGui.QFont()

font.setPointSize(16)

font.setBold(True)

font.setUnderline(True)

font.setWeight(75)

self.label\_7.setFont(font)

self.label\_7.setObjectName("label\_7")

self.label\_12 = QtWidgets.QLabel(self.centralwidget)

self.label\_12.setGeometry(QtCore.QRect(20, 520, 251, 21))

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_12.setFont(font)

self.label\_12.setObjectName("label\_12")

self.label\_19 = QtWidgets.QLabel(self.centralwidget)

self.label\_19.setGeometry(QtCore.QRect(20, 550, 251, 21))

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_19.setFont(font)

self.label\_19.setObjectName("label\_19")

self.label\_20 = QtWidgets.QLabel(self.centralwidget)

self.label\_20.setGeometry(QtCore.QRect(20, 580, 251, 21))

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_20.setFont(font)

self.label\_20.setObjectName("label\_20")

self.label\_21 = QtWidgets.QLabel(self.centralwidget)

self.label\_21.setGeometry(QtCore.QRect(270, 520, 121, 21))

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_21.setFont(font)

self.label\_21.setObjectName("label\_21")

self.label\_22 = QtWidgets.QLabel(self.centralwidget)

self.label\_22.setGeometry(QtCore.QRect(270, 550, 121, 21))

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_22.setFont(font)

self.label\_22.setObjectName("label\_22")

self.label\_23 = QtWidgets.QLabel(self.centralwidget)

self.label\_23.setGeometry(QtCore.QRect(270, 580, 121, 21))

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label\_23.setFont(font)

self.label\_23.setObjectName("label\_23")

MainWindow.setCentralWidget(self.centralwidget)

self.menubar = QtWidgets.QMenuBar(MainWindow)

self.menubar.setGeometry(QtCore.QRect(0, 0, 897, 21))

self.menubar.setObjectName("menubar")

MainWindow.setMenuBar(self.menubar)

self.statusbar = QtWidgets.QStatusBar(MainWindow)

self.statusbar.setObjectName("statusbar")

MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)

QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):

\_translate = QtCore.QCoreApplication.translate

MainWindow.setWindowTitle(\_translate("MainWindow", "MainWindow"))

self.label\_2.setText(\_translate("MainWindow", "HUMANOID ROBOT"))

self.label\_3.setText(\_translate("MainWindow", "graphical user interface"))

self.label\_4.setText(\_translate("MainWindow", "BERDIRI"))

self.label\_5.setText(\_translate("MainWindow", "LAMBAI"))

self.label\_6.setText(\_translate("MainWindow", "JALAN"))

self.label\_8.setText(\_translate("MainWindow", "HORMAT"))

self.label\_9.setText(\_translate("MainWindow", "SQUAT"))

self.label\_10.setText(\_translate("MainWindow", "INPUT"))

self.label\_11.setText(\_translate("MainWindow", "BELAKANG"))

self.label\_14.setText(\_translate("MainWindow", "BELAKANG"))

self.label\_15.setText(\_translate("MainWindow", "X"))

self.label\_16.setText(\_translate("MainWindow", "Y"))

self.label\_17.setText(\_translate("MainWindow", "X"))

self.label\_18.setText(\_translate("MainWindow", "Y"))

self.label\_7.setText(\_translate("MainWindow", "ANGGOTA KELOMPOK :"))

self.label\_12.setText(\_translate("MainWindow", "1. Imam Tantowi Yahya "))

self.label\_19.setText(\_translate("MainWindow", "2. Yafi Ahmad A "))

self.label\_20.setText(\_translate("MainWindow", "3. RM Nararya R "))

self.label\_21.setText(\_translate("MainWindow", "(4121600038)"))

self.label\_22.setText(\_translate("MainWindow", "(4121600056)"))

self.label\_23.setText(\_translate("MainWindow", "(4121600060)"))

if \_\_name\_\_ == "\_\_main\_\_":

import sys

app = QtWidgets.QApplication(sys.argv)

MainWindow = QtWidgets.QMainWindow()

ui = Ui\_MainWindow()

ui.setupUi(MainWindow)

MainWindow.show()

sys.exit(app.exec\_())

## 4.4 Unique Features

Adapun dalam proses pembuatan gerakan pada robot humanoid. Kami menggunakan beberapa metode seperti program untuk tiap gerakan, seperti gerakan berdiri , melambai, hormat, berjalan dan squad. Jadi untuk tiap program yang sudah kami gunakan harus sesuai dengan apa yang kami inginkan. Lalu dari setiap program yang sudah kami ujicoba itu berhasil meskipun masih ada beberapa kelemahan.

# 5 Testing, Evaluation, and Optimization

## 5.1 Testing Strategy

## Langkah pertama dalam pengujian program kami melibatkan serangkaian uji coba dan koreksi. Kami memulainya dengan perencanaan yang rinci untuk setiap gerakan yang ingin dicapai. Selama pengujian, kami menyesuaikan posisi dan kecepatan Dynamixel jika robot mengalami hambatan dalam menjalankan gerakan secara optimal.

## Saat mengalami kendala atau keterbatasan dalam gerakan robot, kami melakukan penyesuaian pada posisi dan kecepatan Dynamixel untuk meningkatkan kinerja robot. Selanjutnya, kami menyematkan program-program tersebut ke dalam void loop sehingga robot dapat melakukan gerakan secara berulang. Dengan begitu, secara sistematis mengembangkan dan menguji program-program untuk memastikan bahwa robot dapat melakukan gerakan dengan kemampuan maksimal secara berkesinambungan.

## 5.2 Performance Evaluation

## Setelah menyelesaikan pemrograman gerakan pada robot humanoid,. Kami terus-menerus melakukan pemeriksaan dan uji coba berulang untuk memastikan bahwa program yang telah kami buat berjalan secara optimal dan konsisten, bukan hanya memberikan hasil satu kali saja. Akhirnya kami mengembangkan program untuk gerakan gerakan yang lebih menarik, kami membuat program yang memungkinkan robot dapar berdiri dengan posisi kuda kuda. Lalu juga ada beberapa gerakan seperti gerakan berjalan yang disitu banyak sekali menemukan kelemahan.

Oleh karena itu, kami mencoba untuk mengupdate program agar bisa sesuai dengan gerakan yang berjalan. Sampai pada akhirnya kami menemukan program yang sesuai meskipun dalam program tersebut masih ada sedikit kelemahan akan tetepi robot dapat melakukan gerakan berjalan dan beroperasi dengan baik.

## 5.3 Optimization

# Kinerja Sistem: Memastikan agar seluruh sistem robot humanoid beroperasi secara efisien, mengurangi bug atau kesalahan, dan memperbaiki respons sistem terhadap perintah-perintah yang diberikan.

# Kualitas Gerakan: Meningkatkan kualitas gerakan robot, termasuk kecepatan, ketepatan, dan koordinasi dalam menjalankan gerakan yang diinginkan. Pengoptimalan dilakukan melalui modifikasi pada program, perangkat keras, atau algoritma kontrol untuk mencapai gerakan yang lebih realistis dan terkoordinasi.

# Stabilitas dan Keandalan: Mengatasi masalah kestabilan, terutama dalam situasi di mana robot rentan roboh atau kesulitan 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 mencakup 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 berkelanjutan, pengembang robot humanoid bertujuan untuk meningkatkan kemampuan robot dalam menjalankan tugasnya dengan lebih baik, lebih aman, dan lebih responsif terhadap lingkungan. Tujuan akhirnya adalah memberikan pengalaman kepada pengguna dan mengoptimalkan manfaat dari penggunaan robot humanoid di berbagai bidang aplikasi.

# 6 Collaboration and Project Management

## 6.1 Teamwork Dynamics.

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**: Dalam pengembangan robot humanoid, kerja tim yang dinamis mencakup kolaborasi, penentuan peran, dan kontribusi anggota tim untuk mencapai tujuan bersama. Kolaborasi melibatkan berbagi ide dan keterampilan di antara anggota tim dengan spesialisasi yang beragam, seperti perangkat lunak, perangkat keras, mekanika, dan pemrograman, yang perlu berkerja terkoordinasi.
* **Peran dan Tanggung Jawab**: Setiap anggota tim memiliki peran dan tanggung jawab tertentu, seperti fokus pada pengembangan perangkat lunak atau terlibat dalam perancangan mekanika dan pengujian.
* **Kontribusi**: Kontribusi unik dari setiap anggota tim menjadi kunci dalam memecahkan masalah, berkolaborasi, dan menyajikan ide atau solusi kreatif untuk meningkatkan kualitas serta kinerja robot humanoid secara holistik dan efektif.

## 6.2 Project Management

Manajemen proyek berfokus pada dokumentasi jadwal proyek, pencapaian, dan praktik manajemen untuk memastikan kelancaran dan pencapaian tujuan proyek. Dalam pengembangan robot humanoid, aspek manajemen proyek mencakup:

* **Dokumentasi**: Penyusunan jadwal proyek dengan rincian tenggat waktu, tahapan pengembangan, dan target pencapaian. Dokumentasi ini memudahkan pemantauan kemajuan proyek serta membantu identifikasi masalah atau kendala selama proses pengembangan.
* **Pencapaian**: Memantau pencapaian proyek sesuai dengan jadwal yang ditetapkan. Ini mencakup pencatatan progres, evaluasi setiap tahap pengembangan, dan memastikan pencapaian tujuan yang telah ditetapkan.
* **Praktik Manajemen**: Implementasi praktik manajemen yang efektif, seperti identifikasi dan mitigasi risiko, alokasi sumber daya yang tepat, komunikasi efisien antar tim, dan adaptasi terhadap perubahan yang mungkin terjadi selama proses pengembangan.

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

# 7 Conclusion and Reflection

## 7.1 Project Summary

Proyek pengembangan gerakan robot humanoid telah sukses mengatasi sejumlah tantangan utama dalam upaya meningkatkan kemampuan geraknya. Peningkatan kualitas gerakan berhasil dicapai melalui optimalisasi algoritma, penyesuaian perangkat keras, dan peningkatan responsivitas sistem. Melalui langkah-langkah pengujian sistematis, kami memastikan bahwa robot mampu konsisten dan optimal dalam melakukan gerakan di berbagai konteks. Hasil dari proyek ini menandai kemajuan penting dalam pengembangan robot humanoid yang lebih adaptif dan responsif..

## 7.2 Future Work

Perkembangan di masa depan dapat difokuskan pada integrasi teknologi sensor terkini, pengembangan algoritma kontrol yang lebih pintar, dan pemahaman yang lebih mendalam terkait interaksi antara manusia dan robot. Penerapan robot humanoid juga dapat diperluas ke berbagai sektor penting dalam kehidupan sehari-hari, seperti perawatan kesehatan, edukasi, dan eksplorasi lingkungan berbahaya

## 7.3 Personal and Group Reflections

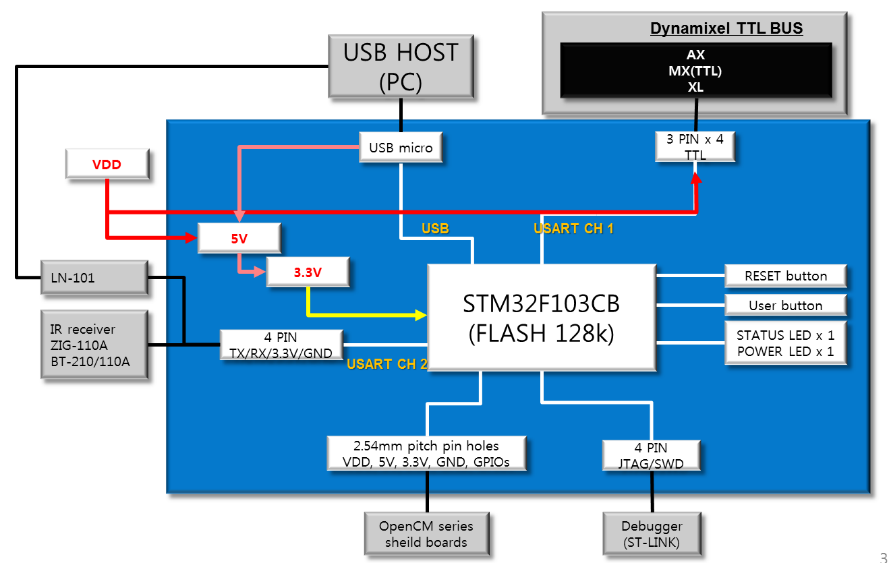
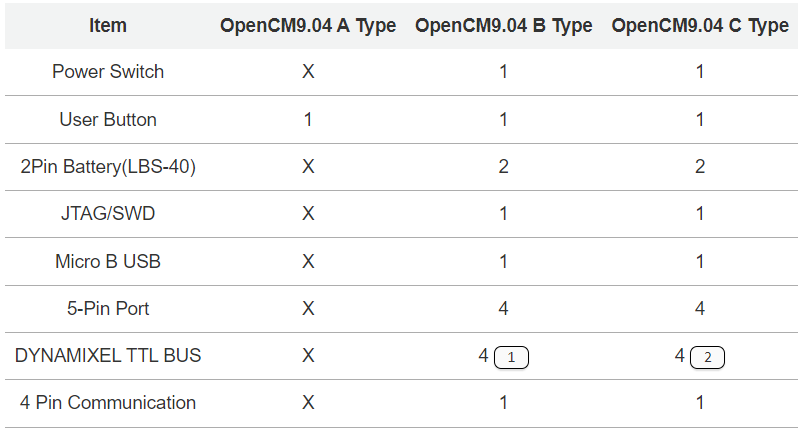
Selama perjalanan proyek ini, kami dihadapkan pada berbagai tantangan yang signifikan. Namun, melalui kerjasama tim yang solid dan manajemen proyek yang efektif, kami berhasil mengatasi setiap rintangan. Pengalaman ini mengajarkan kami banyak hal tentang kompleksitas pengembangan robot humanoid, baik dari segi teknis maupun interaksi manusia-robot. Dalam refleksi pribadi, kami menekankan betapa pentingnya kerjasama, kesabaran, dan dedikasi dalam mencapai tujuan bersama. Secara bersama-sama, proyek ini tidak hanya memperluas wawasan kami, tetapi juga mempersiapkan kami untuk menghadapi tantangan di masa depan dalam pengembangan teknologi robotika.

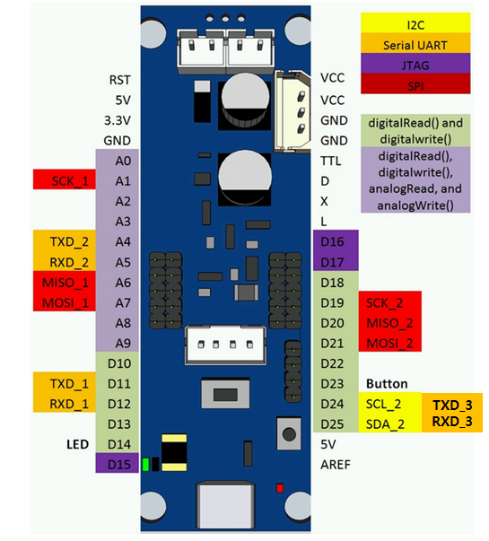
# 8 Appendices

## 8.1 Bill of Materials

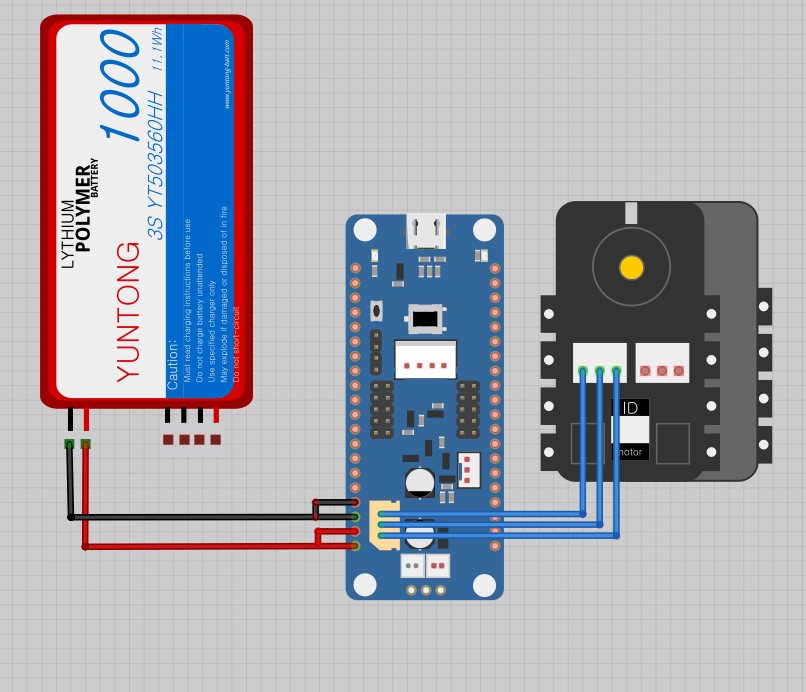
Rincian tampilan GUI untuk mengendalikan pergerakan robot humanoid menggunakan Python melibatkan beberapa elemen kunci. Bagian utama mencakup antarmuka grafis itu sendiri, tombol kontrol untuk berbagai gerakan dan fungsi, serta tampilan status yang memberikan informasi tentang kondisi robot. Dalam konteks biaya, komponen seperti sensor gerak, kamera untuk visualisasi, dan mikrofon untuk interaksi suara dapat mempengaruhi anggaran secara signifikan. Investasi tambahan mungkin diperlukan untuk sumber daya pembelajaran Python GUI development, termasuk buku, kursus online, dan dokumentasi resmi. Dengan mengintegrasikan semua elemen ini, tampilan GUI dapat memberikan pengendalian yang intuitif dan informatif atas pergerakan robot humanoid, sambil menjaga efisiensi biaya yang terkendali.

## 8.2 Electrical Wiring and System Layout

* 1. Block Diagram
  2. OpenCM0.04 C-Type
  3. OpenCM9.04 GPIO PinMap



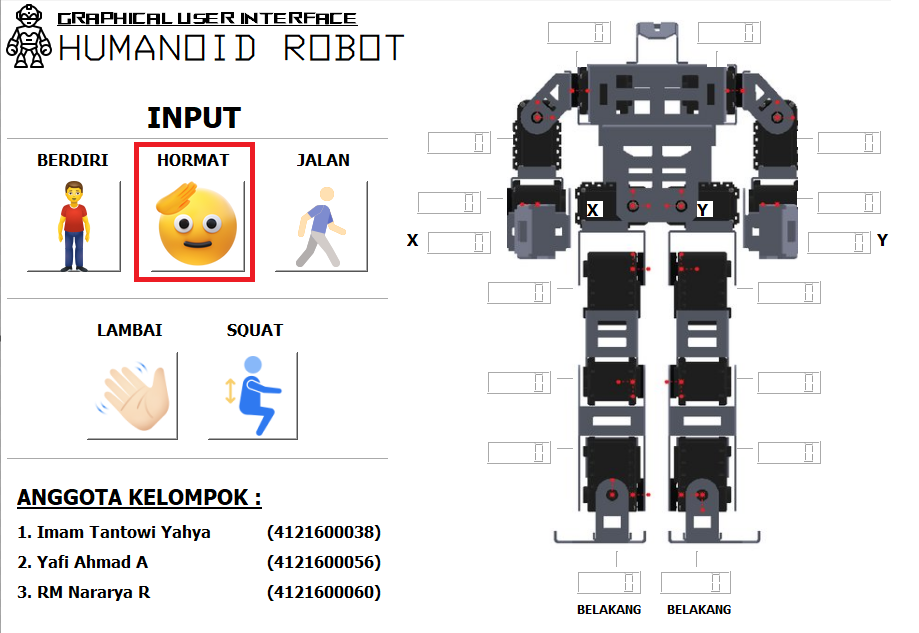
* 1. Example : Open CM9.04 dan AX12 connection



## 8.3 Code Repository

## <https://github.com/anh0001/HumanoidRobot-GUI-Control/tree/kelasB_group1>

## 8.4 Additional Documentation



# 9 References

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.