

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.

SISTEM PEMROGRAMAN MEKATRONIKA

Team Humanoid 1

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.

Zahra Athira (4121600006), Ubaidillah Ramadhan Nur Santoso (4121600012)
{zahraathira, ubaidillahramadhan850}@me.student.pens.ac.id



Table of Contents

1 Introduction and Initial Analysis	3
1.1 Project Context	3
1.2 Initial Thought Process	3
2 Requirement Analysis and Specification	3
2.1 User Requirements	3
2.2 System Requirements	4
2.3 Tools and Technologies	4
3 Conceptual Design	5
3.1 System Architecture	5
3.2 Interface Design	7
3.3 Control Algorithm Design	7
4 Detailed Design and Development	7
4.1 Component Design	8
4.2 Coding and Implementation	9
4.3 Integration	18
4.4 Unique Features	20
5 Testing, Evaluation, and Optimization	22
5.1 Testing Strategy	22
5.2 Performance Evaluation	22
5.3 Optimization	22
6 Collaboration and Project Management	23
6.1 Teamwork Dynamics	23
6.2 Project Management	23
7 Conclusion and Reflection	24
7.1 Project Summary	24
7.2 Future Work	24
7.3 Personal and Group Reflections	24
8 Appendices	24
8.1 Bill of Materials	24
8.2 Electrical Wiring and System Layout	25
8.3 Code Repository	26



8.4 Additional Documentation	27
9 References	27



1 Introduction and Initial Analysis

1.1 Project Context

Pengembangan gerakan robot humanoid merupakan komponen kunci dalam pengembangan robotika. Robot humanoid memiliki potensi besar dalam berbagai aplikasi, termasuk industri, penelitian, dan hiburan, namun kemampuan gerakannya masih menjadi tantangan. Oleh karena itu, proyek ini memiliki latar belakang dalam upaya meningkatkan kemampuan gerakan robot humanoid untuk meningkatkan kegunaannya.

1.1.1 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

Tujuan proyek ini adalah mengembangkan gerakan yang lebih realistis dan beragam pada robot humanoid yang sudah ada, untuk meningkatkan efektivitas penggunaannya dalam berbagai aplikasi. Kesuksesan proyek ini akan memajukan pengembangan robot humanoid dan meningkatkan manfaatnya dalam berbagai bidang.

1.2 Initial Thought Process

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 Requirement Analysis and Specification

2.1 User Requirements

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 System Requirements

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 Tools and Technologies

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 Target specification

Tabel 1. Tabel caption.

Feature	Description	Measurement Metric	Target Value
Gerakan Jalan Ditempat	Robot harus dapat tetap pada posisi tanpa gerakan translasional.	Perpindahan dari titik awal hingga akhir, apakah tetap diposisi awal atau berubah.	Robot dapat bergerak lebih dari 3 menit



Gerakan Senam Tangan	Robot melakukan Gerakan tangan kanan dan kekiri	Sudut rotasi tangan atau jarak linear tangan dari posisi awal.	Tangan robot harus dapat bergerak ke arah kanan dengan sudut rotasi minimal 90 derajat.
Gerakan Jalan Maju	Robot berjalan maju dengan gerakan kaki.	Perpindahan dari titik awal.	Robot dapat bergerak maju 1 meter.

3 Conceptual Design

3.1 System Architecture


High-Level Architecture

- Antarmuka Pengguna (UI) : untuk komunikasi dengan pengguna melalui elemen UI seperti yang ditampilkan GUI kami yaitu tombol Maju, Mundur, Kanan, Kiri, Stop dan Jalan ditempat.
- Logika : mengelola proses dan pengambilan keputusan berdasarkan input dari UI dan data dari akses data.
- Database : untuk mendapatkan dan menyimpan informasi.





GUI (Graphical User Interface)

Desain sementara tampilan GUI kami :



-  : Tombol Maju untuk robot bergerak maju



-  : Tombol Mundur untuk robot bergerak mundur
-  : Tombol Kiri untuk robot bergerak ke kiri
-  : Tombol Kanan untuk robot bergerak ke kanan
-  : Tombol Maju untuk robot bergerak maju

Control Logic

Input : Ketika user menekan tombol (Maju, Mundur, Kanan, dll), ini memberitahu control logic bahwa user ingin menjalankan robot.

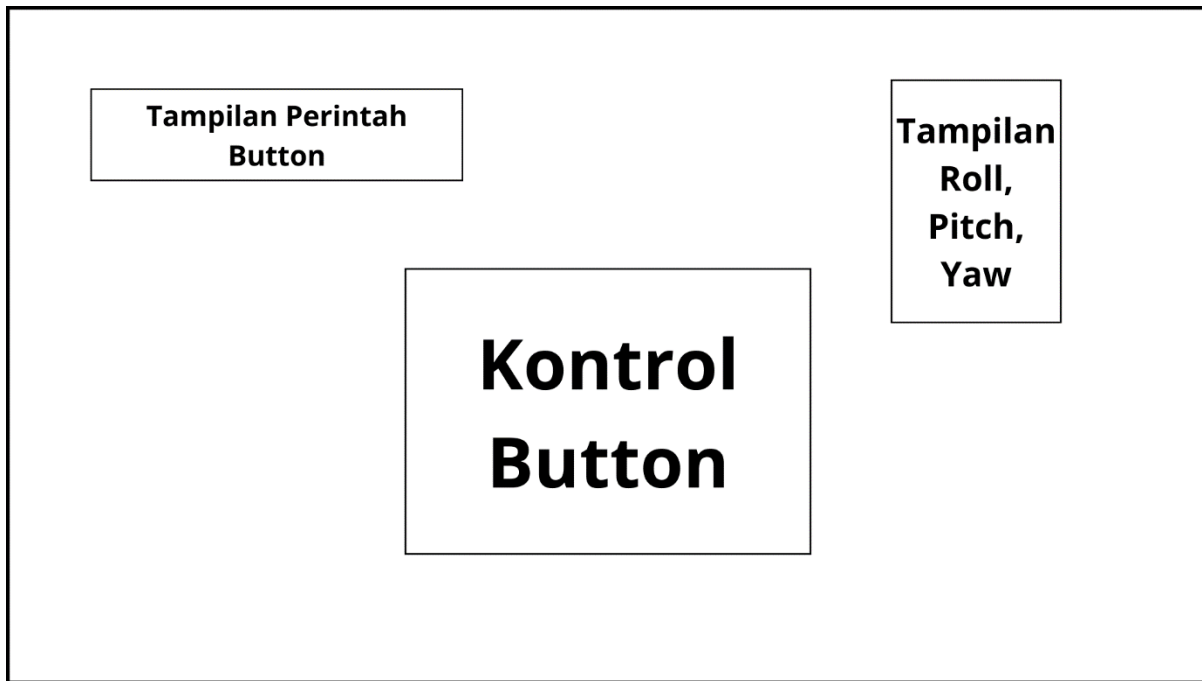
Proses : Saat robot berdiri diam atau posisi awal, control logic akan menginstruksikan robot untuk bergerak sesuai permintaan user saat menekan tombol.

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

3.2 Interface Design

Berikut adalah sketsa awal dari tampilan GUI :

Layout



3.3 Control Algorithm Design

a. 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.

b. 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

- ✓ OpenCM9.04
- ✓ OpenCM 485 Expansion Board
- ✓ Servo Dynamixel AX-12A : 6
- ✓ Servo Dynamixel AX-18A : 10

	OpenCm 9.04
---	-------------

	Dynamixel AX-12A
	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
/* mendefinisikann dynamixel dengan set ID */
#define ID_NUM1 1
#define ID_NUM2 2
#define ID_NUM3 3
#define ID_NUM4 4
#define ID_NUM5 5
#define ID_NUM6 6
#define ID_NUM7 9
#define ID_NUM8 10
#define ID_NUM9 11
#define ID_NUM10 12
#define ID_NUM11 13
#define ID_NUM12 14
#define ID_NUM13 15
#define ID_NUM14 16
#define ID_NUM15 17
#define ID_NUM16 18
```

/* library dynamixel yang akan digunakan kemudian juga nantinya akan dihubungkan serial dengan board

*/

```
Dynamixel Dxl(DXL_BUS_SERIAL3);
```

```
const long interval = 100;
```

```
unsigned long previousTime = 0;
```

```
void setup() {
```

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

```
Dxl.begin(3);
Dxl.jointMode(ID_NUM1);
Dxl.jointMode(ID_NUM2);
//Serial.begin(9600);
}

void loop()
{
  /* komunikasi menghubungkan gui */
  if (SerialUSB.available() > 0)
  {
    /* komunikasi dengan menghubungkan kabel serial*/
    char data = SerialUSB.read();

    /* mengirimkan sebuah data dalam bentuk character*/
    if (data == 'A')
    {
      /* memasukan fungsi yang lain ke dalam program yang akan di perintah */
      berdiri_tegak();
    }

    else if (data == 'B')
    {
      /* memasukan fungsi yang lain ke dalam program yang akan di perintah */
      melambatkan();
      delay(10000);
    }

    else if (data == 'C')
    {
      /* memasukan fungsi yang lain ke dalam program yang akan di perintah */
      kuda_kuda();
    }

    else if (data == 'D')
    {
      /* memasukan fungsi yang lain ke dalam program yang akan di perintah */
      pukul();
    }

    else if (data == 'E')
    {
      /* memasukan fungsi yang lain ke dalam program yang akan di perintah */
      senam_tangan();
    }
  }
}
```

```
    else if (data == 'F')
    {
        /* memasukan fungsi yang lain ke dalam program yang akan di perintah */
        berjalan();
    }

    else if (data == 'G')
    {
        /* memasukan fungsi yang lain ke dalam program yang akan di perintah */
        jalan_ditempat();
    }

}

}

void berdiri_tegak()
{
    // kaki kanan
    Dxl.setPosition(ID_NUM7,530,50);
    Dxl.setPosition(ID_NUM9,670,50);
    Dxl.setPosition(ID_NUM11,612,50);
    Dxl.setPosition(ID_NUM13,512,50);
    Dxl.setPosition(ID_NUM15,530,50);
    // kaki kiri
    Dxl.setPosition(ID_NUM8,500,50);
    Dxl.setPosition(ID_NUM10,364,50);
    Dxl.setPosition(ID_NUM12,412,50);
    Dxl.setPosition(ID_NUM14,512,50);
    Dxl.setPosition(ID_NUM16,500,50);
    //tangankanandankiri
    Dxl.setPosition(ID_NUM1,515,50);
    delay(2000);
    Dxl.setPosition(ID_NUM3,250,50);
    Dxl.setPosition(ID_NUM5,315,50);
    //----
    Dxl.setPosition(ID_NUM2,515,50);
    delay(2000);
    Dxl.setPosition(ID_NUM4,460,50);
    Dxl.setPosition(ID_NUM6,715,50);

}

void berjalan()
{
    //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)
}

void jalan_ditempat()
{
    Dxl.setPosition(ID_NUM1,815,100);
    Dxl.setPosition(ID_NUM3,215,100);
    Dxl.setPosition(ID_NUM5,215,100);
    Dxl.setPosition(ID_NUM2,215,100);
    Dxl.setPosition(ID_NUM4,515,100);
    Dxl.setPosition(ID_NUM6,515,100);
    delay(1000);
    Dxl.setPosition(ID_NUM1,815,300);
    Dxl.setPosition(ID_NUM3,115,300);
    Dxl.setPosition(ID_NUM5,215,300);
    Dxl.setPosition(ID_NUM2,215,300);
    Dxl.setPosition(ID_NUM4,615,300);
    Dxl.setPosition(ID_NUM6,515,300);
    delay(1000);
    Dxl.setPosition(ID_NUM11,612,50);
    Dxl.setPosition(ID_NUM12,412,50);
    delay(3000);
    //kanan
    Dxl.setPosition(ID_NUM11,562,50);
    Dxl.setPosition(ID_NUM12,412,50);
    delay(3000);
    //kiri
    Dxl.setPosition(ID_NUM11,612,50);
    Dxl.setPosition(ID_NUM12,362,50);
    delay(3000);
    //kanan
    Dxl.setPosition(ID_NUM11,562,50);
    Dxl.setPosition(ID_NUM12,412,50);
    delay(1000);
    //kiri
    Dxl.setPosition(ID_NUM11,612,50);
    Dxl.setPosition(ID_NUM12,362,50);
    delay(1000);
    //kanan
    Dxl.setPosition(ID_NUM11,562,50);
    Dxl.setPosition(ID_NUM12,412,50);
    delay(1000);
    //kiri
    Dxl.setPosition(ID_NUM11,612,50);
    Dxl.setPosition(ID_NUM12,362,50);
    delay(1000);
    //kanan
```

```
Dxl.setPosition(ID_NUM11,562,50);
Dxl.setPosition(ID_NUM12,412,50);
delay(1000);
//kiri
Dxl.setPosition(ID_NUM11,612,50);
Dxl.setPosition(ID_NUM12,362,50);
delay(1000);
//kanan
Dxl.setPosition(ID_NUM11,562,50);
Dxl.setPosition(ID_NUM12,412,50);
delay(1000);
//kiri
Dxl.setPosition(ID_NUM11,612,50);
Dxl.setPosition(ID_NUM12,362,50);
delay(1000);
//kanan
Dxl.setPosition(ID_NUM11,562,50);
Dxl.setPosition(ID_NUM12,412,50);
delay(1000);
//kiri
Dxl.setPosition(ID_NUM11,612,50);
Dxl.setPosition(ID_NUM12,362,50);
delay(1000);
//kanan
Dxl.setPosition(ID_NUM11,562,50);
Dxl.setPosition(ID_NUM12,412,50);
delay(1000);
//kiri
Dxl.setPosition(ID_NUM11,612,50);
Dxl.setPosition(ID_NUM12,362,50);
delay(1000);
//kanan
Dxl.setPosition(ID_NUM11,562,50);
Dxl.setPosition(ID_NUM12,412,50);
delay(1000);
//kiri
Dxl.setPosition(ID_NUM11,612,50);
Dxl.setPosition(ID_NUM12,362,50);
delay(1000);
//kanan
Dxl.setPosition(ID_NUM11,562,50);
Dxl.setPosition(ID_NUM12,412,50);
delay(1000);
//kiri
Dxl.setPosition(ID_NUM11,612,50);
Dxl.setPosition(ID_NUM12,362,50);
delay(1000);
//kanan
```

```

Dxl.setPosition(ID_NUM11,562,50);
Dxl.setPosition(ID_NUM12,412,50);
delay(1000);
//kiri
Dxl.setPosition(ID_NUM11,612,50);
Dxl.setPosition(ID_NUM12,362,50);
delay(1000);
//kanan
Dxl.setPosition(ID_NUM11,562,50);
Dxl.setPosition(ID_NUM12,412,50);
delay(1000);
//kiri
Dxl.setPosition(ID_NUM11,612,50);
Dxl.setPosition(ID_NUM12,362,50);
delay(1000);
//kanan
Dxl.setPosition(ID_NUM11,562,50);
Dxl.setPosition(ID_NUM12,412,50);
delay(1000);
//kiri
Dxl.setPosition(ID_NUM11,612,50);
Dxl.setPosition(ID_NUM12,362,50);
delay(1000);
}

void kuda_kuda()
{
    // kaki kanan
    Dxl.setPosition(ID_NUM7,530,50);
    Dxl.setPosition(ID_NUM9,670,50);
    Dxl.setPosition(ID_NUM11,612,50);
    Dxl.setPosition(ID_NUM13,450,50);
    Dxl.setPosition(ID_NUM15,530,50);
    // kaki kiri
    Dxl.setPosition(ID_NUM8,500,50);
    Dxl.setPosition(ID_NUM10,574,50);
    Dxl.setPosition(ID_NUM12,412,50);
    Dxl.setPosition(ID_NUM14,352,50);
    Dxl.setPosition(ID_NUM16,500,50);
    //tangan kanan
    Dxl.setPosition(ID_NUM1,615,50);
    delay(2000);
    Dxl.setPosition(ID_NUM3,215,50);
    Dxl.setPosition(ID_NUM5,215,50);
    //tangan kiri
    Dxl.setPosition(ID_NUM2,415,50);
    delay(2000);
    Dxl.setPosition(ID_NUM4,515,50);

```



```
Dxl.setPosition(ID_NUM6,515,50);
}

void melambakan()
{
    Dxl.setPosition(ID_NUM1,1023,300);
    delay(2000);
    Dxl.setPosition(ID_NUM3,215,100);
    Dxl.setPosition(ID_NUM5,15,100);
    delay(1000);
    Dxl.setPosition(ID_NUM3,115,300);
    delay(1000);
    Dxl.setPosition(ID_NUM3,315,300);
    delay(1000);
    Dxl.setPosition(ID_NUM3,115,300);
    delay(1000);
    Dxl.setPosition(ID_NUM3,315,300);
    delay(1000);
    Dxl.setPosition(ID_NUM3,115,300);
    delay(1000);
    Dxl.setPosition(ID_NUM3,315,300);
    delay(1000);
    Dxl.setPosition(ID_NUM1,515,100);
    Dxl.setPosition(ID_NUM3,315,100);
    Dxl.setPosition(ID_NUM5,15,100);

}

void pukul()
{
    Dxl.setPosition(ID_NUM1,615,100);
    Dxl.setPosition(ID_NUM3,215,100);
    Dxl.setPosition(ID_NUM5,215,100);
    delay(1000);

    Dxl.setPosition(ID_NUM1,515,200);
    Dxl.setPosition(ID_NUM3,215,100);
    Dxl.setPosition(ID_NUM5,215,100);
    delay(1000);
    Dxl.setPosition(ID_NUM1,815,400);
    Dxl.setPosition(ID_NUM3,215,100);
    Dxl.setPosition(ID_NUM5,215,100);
    delay(1000);
    Dxl.setPosition(ID_NUM1,615,300);
    Dxl.setPosition(ID_NUM3,215,100);
    Dxl.setPosition(ID_NUM5,215,100);
}
```

```
void senam_tangan()
{
  //tangan kiri
  Dxl.setPosition(ID_NUM1,815,100);
  Dxl.setPosition(ID_NUM3,215,100);
  Dxl.setPosition(ID_NUM5,215,100);
  //tangan kiri
  Dxl.setPosition(ID_NUM2,215,100);
  Dxl.setPosition(ID_NUM4,515,100);
  Dxl.setPosition(ID_NUM6,515,100);
  delay(1000);
  //gerakan 1
  Dxl.setPosition(ID_NUM1,815,300);
  Dxl.setPosition(ID_NUM3,315,300);
  Dxl.setPosition(ID_NUM5,215,300);

  Dxl.setPosition(ID_NUM2,215,300);
  Dxl.setPosition(ID_NUM4,415,300);
  Dxl.setPosition(ID_NUM6,515,300);
  delay(1000);
  // gerakan 2
  Dxl.setPosition(ID_NUM1,815,300);
  Dxl.setPosition(ID_NUM3,115,300);
  Dxl.setPosition(ID_NUM5,215,300);

  Dxl.setPosition(ID_NUM2,215,300);
  Dxl.setPosition(ID_NUM4,615,300);
  Dxl.setPosition(ID_NUM6,515,300);
  delay(1000);
  // gerakan 3
  Dxl.setPosition(ID_NUM1,815,300);
  Dxl.setPosition(ID_NUM3,315,300);
  Dxl.setPosition(ID_NUM5,215,300);

  Dxl.setPosition(ID_NUM2,215,300);
  Dxl.setPosition(ID_NUM4,415,300);
  Dxl.setPosition(ID_NUM6,515,300);
  delay(1000);
  // gerakan 4
  Dxl.setPosition(ID_NUM1,815,300);
  Dxl.setPosition(ID_NUM3,115,300);
  Dxl.setPosition(ID_NUM5,215,300);

  Dxl.setPosition(ID_NUM2,215,300);
  Dxl.setPosition(ID_NUM4,615,300);
  Dxl.setPosition(ID_NUM6,515,300);
  delay(1000);
  //gerakan 5
```

```
Dxl.setPosition(ID_NUM1,815,300);
Dxl.setPosition(ID_NUM3,315,300);
Dxl.setPosition(ID_NUM5,215,300);

Dxl.setPosition(ID_NUM2,215,300);
Dxl.setPosition(ID_NUM4,415,300);
Dxl.setPosition(ID_NUM6,515,300);
delay(1000);
//gerakan 6
Dxl.setPosition(ID_NUM1,815,300);
Dxl.setPosition(ID_NUM3,115,300);
Dxl.setPosition(ID_NUM5,215,300);

Dxl.setPosition(ID_NUM2,215,300);
Dxl.setPosition(ID_NUM4,615,300);
delay(1000);
//gerakan 7
Dxl.setPosition(ID_NUM1,815,300);
Dxl.setPosition(ID_NUM3,315,300);
Dxl.setPosition(ID_NUM5,215,300);

Dxl.setPosition(ID_NUM2,215,300);
Dxl.setPosition(ID_NUM4,415,300);
delay(1000);
//gerakan 8
Dxl.setPosition(ID_NUM1,815,300);
Dxl.setPosition(ID_NUM3,115,300);
Dxl.setPosition(ID_NUM5,215,300);

Dxl.setPosition(ID_NUM2,215,300);
Dxl.setPosition(ID_NUM4,615,300);
delay(1000);

// gerakan selesai
Dxl.setPosition(ID_NUM3,315,300);
Dxl.setPosition(ID_NUM5,15,300);
Dxl.setPosition(ID_NUM4,415,300);
Dxl.setPosition(ID_NUM6,715,300);
delay(500);
Dxl.setPosition(ID_NUM1,515,300);
Dxl.setPosition(ID_NUM2,515,300);

}
```

4.3 Integration

Integrasi GUI dengan sistem kontrol pada robot humanoid.



```
import tkinter as tk
from tkinter import PhotoImage
import serial

def turn_on_led():
    ser.write(b'1')

def turn_off_led():
    ser.write(b'0')

def button_kanan_click():
    label.config(text="Tombol Kanan diklik!")
    button_pause_click('H')

def button_pause_click(command):
    label.config(text="Tombol Pause diklik!")

def button_kiri_click():
    label.config(text="Tombol Kiri diklik!")
    button_pause_click('L')

def button_maju_click(command):
    label.config(text="Tombol Maju diklik!")

def button_mundur_click():
    label.config(text="Tombol Mundur diklik!")

root = tk.Tk()
root.title("Kontrol LED Arduino")

ser = serial.Serial('/dev/ttyUSB0', 9600) # Sesuaikan dengan port serial Arduino Anda

on_button = tk.Button(root, text="Hidupkan LED", command=turn_on_led)
on_button.grid()

off_button = tk.Button(root, text="Matikan LED", command=turn_off_led)
off_button.grid()

# Membuat frame untuk tombol
button_frame = tk.Frame(root)

# Mengimpor gambar latar belakang
bg_image = PhotoImage(file="bgGUI.png")

# Membuat label untuk latar belakang
bg_label = tk.Label(root, image=bg_image)
bg_label.place(relwidth=1, relheight=1)
```



```
# Mengimpor gambar untuk tombol maju, mundur, dan pause
kanan_image = PhotoImage(file="kanan.png")
pause_image = PhotoImage(file="pause.png")
kiri_image = PhotoImage(file="kiri.png")
maju_image = PhotoImage(file="maju.png")
mundur_image = PhotoImage(file="mundur.png")

# Mengatur ukuran tombol dengan metode subsample
kiri_image = kiri_image.subsample(2) # Mengurangi ukuran gambar menjadi setengah
pause_image = pause_image.subsample(2) # Mengurangi ukuran gambar menjadi setengah
kanan_image = kanan_image.subsample(2) # Mengurangi ukuran gambar menjadi setengah
maju_image = maju_image.subsample(2) # Mengurangi ukuran gambar menjadi setengah
mundur_image = mundur_image.subsample(2) # Mengurangi ukuran gambar menjadi setengah

# Membuat tombol dengan gambar sebagai elemen tombol
button_kiri = tk.Button(root, image=kiri_image, command=button_kiri_click)
button_pause = tk.Button(root, image=pause_image, command=button_pause_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)

# Mengatur tombol-tombol agar sejajar dan berada di tengah
button_kiri.grid(row=7, column=2, padx=10)
button_pause.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)

# Membuat label untuk menampilkan pesan
label = tk.Label(root, text="")
label.grid(row=1, column=0, columnspan=3)

root.mainloop()
```

4.4 Unique Features

```
void berjalan()
{
    //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)
}

```



5 Testing, Evaluation, and Optimization

5.1 Testing Strategy

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 Performance Evaluation

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 Optimization

- **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 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:** 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 Project Management

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.

7 Conclusion and Reflection

7.1 Project Summary

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 Future Work

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 Personal and Group Reflections

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 Appendices

8.1 Bill of Materials

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

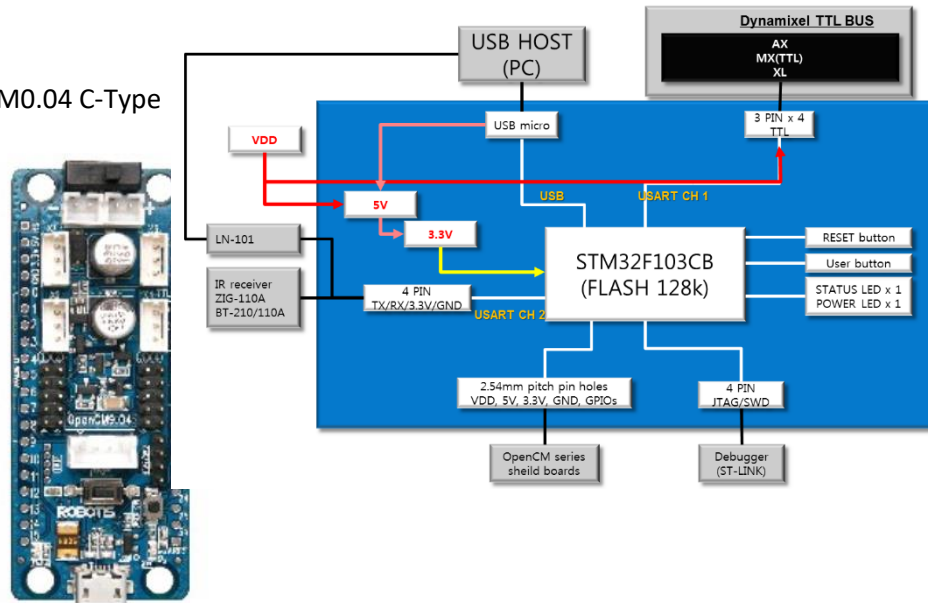
suara dapat mempengaruhi anggaran. Sumber daya untuk belajar Python GUI development, seperti buku, kursus online, dan dokumentasi resmi, dapat menjadi investasi tambahan. Dengan menggabungkan elemen-elemen ini, tampilan GUI dapat memberikan kontrol yang intuitif dan informatif atas gerakan robot humanoid dengan efisiensi biaya yang terkontrol.

8.2 Electrical Wiring and System Layout

a. Block Diagram

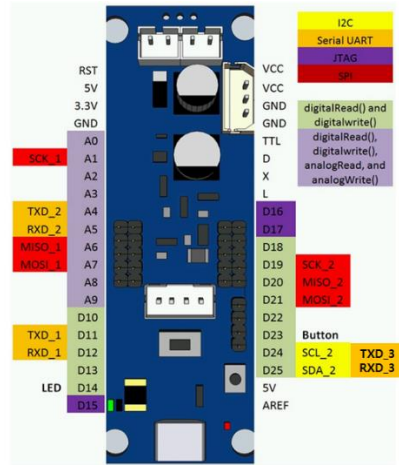
b.

c. OpenCM0.04 C-Type

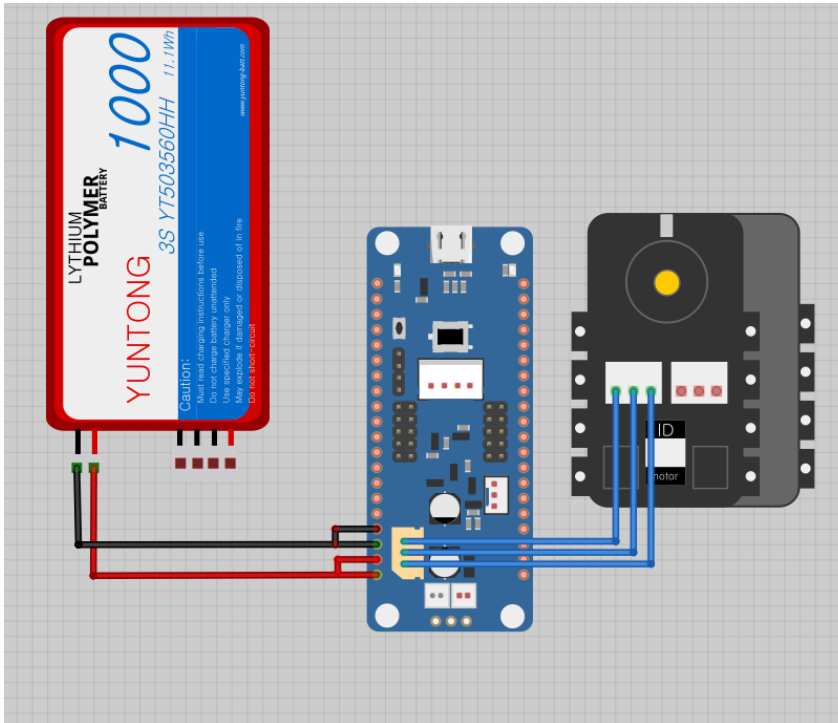


Item	OpenCM9.04 A Type	OpenCM9.04 B Type	OpenCM9.04 C Type
Power Switch	X	1	1
User Button	1	1	1
2Pin Battery(LBS-40)	X	2	2
JTAG/SWD	X	1	1
Micro B USB	X	1	1
5-Pin Port	X	4	4
DYNAMIXEL TTL BUS	X	4 (1)	4 (2)
4 Pin Communication	X	1	1

d. OpenCM9.04 GPIO PinMap



e. Example : Open CM9.04 dan AX12 connection



8.3 Code Repository

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

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.