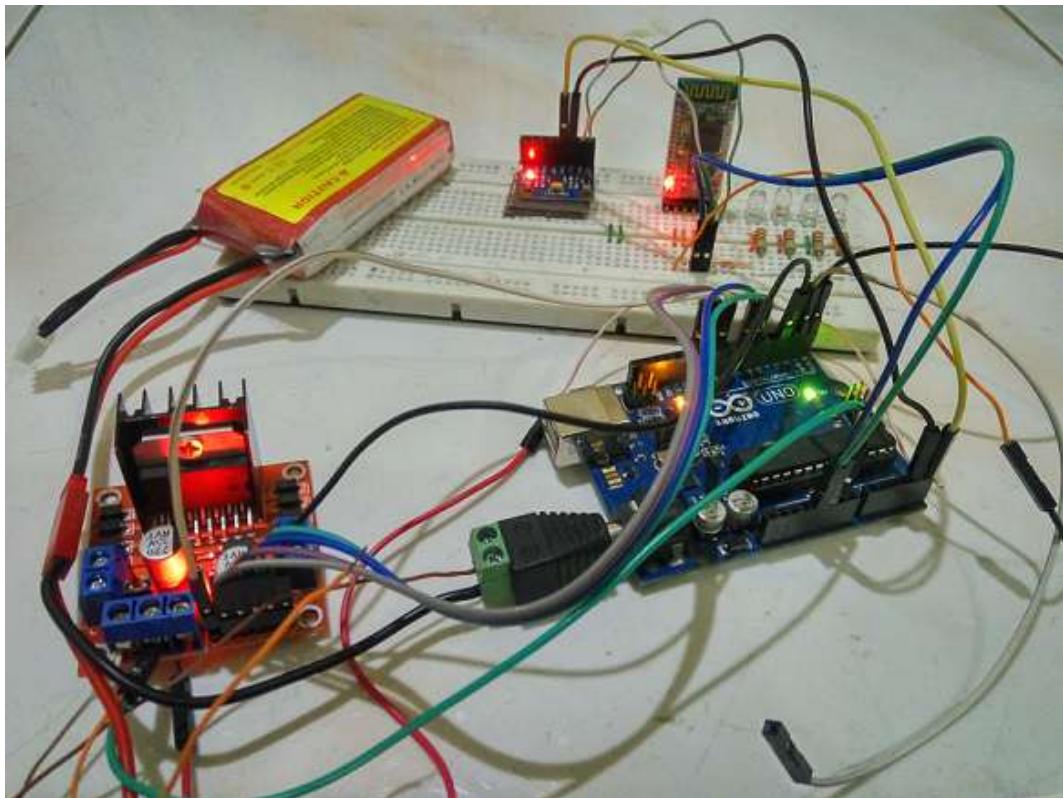


## BAB IV

### HASIL DAN UJI COBA

#### IV.1 Hasil

Pada bab ini, penulis akan menampilkan tampilan hasil perancangan yang telah dijelaskan pada bab sebelumnya dari perancangan dan implementasi robot keseimbangan menggunakan metode PID sampai perancangan *interface android* yang akan dijelaskan pada beberapa gambar IV.1 berikut :



**Gambar IV.1 Tampilan Hasil Alat Sebelum Pemasangan**

## **IV.2. Uji Coba Hasil**

Pada bab ini, akan dibahas pengujian alat mulai dari pengujian alat permodul sampai pengujian alat secara keseluruhan. Pengujian tersebut akan dilakukan secara bertahap dengan urutan sebagai berikut :

- Pengujian Rangkaian Minimum Sistem
- Pengujian *sensor GY-521*
- Pengujian *driver motor*
- Pengujian Komunikasi Serial Bluetooth
- Pengujian *Interface*
- Pengujian Sistem Secara Keseluruhan

### **IV.2.1 Pengujian Rangkaian Minimum Sistem**

Pada pengujian minimum sistem ini dilakukan percobaan yang sifatnya sederhana tapi dapat menunjukkan bekerja atau tidaknya minimum sistem tersebut. Percobaan tersebut adalah menggunakan lampu led. Untuk pengujian rangkaian minimum tersebut digunakan program sebagai berikut :

```
Void setup() {  
pinMode(6,OUTPUT);}  
  
Void loop{  
digitalWrite(6,HIGH);  
delay (10000);  
digitalWrite(6,LOW);}
```

```
delay(10000);
```

```
}
```

Pengujian ini dilakukan untuk memastikan minimum sistem bekerja dengan baik.

Pengujian dilakukan dengan menggunakan koding sederhana.

#### IV.2.2 Pengujian Rangkaian GY-521

Pada pengujian GY-521 dilakukan dengan membuat rangkaian sederhana dengan menggunakan arduino dan sensor GY-521. Program yang digunakan dapat dilihat dibawah ini :

```
MPU6050 gy_521;
```

```
int16_t ax, ay, az;
```

```
int16_t gx, gy, gz;
```

```
int ACCX,ACCY;
```

```
int GYRX,GYRY,GYRZ,sensitif;
```

```
void acc()
```

```
{
```

```
gy_521.getAcceleration(&ax, &ay, &az);
```

```
ACCX = 57.295*atan((float)ay/ sqrt(pow((float)az,2)+pow((float)ax,2)));
```

```
ACCY = 57.295*atan((float)-ax/ sqrt(pow((float)az,2)+pow((float)ay,2)));
```

```
}
```

```
void gyro()
```

```
{
```

```
gy_521.getRotation(&gx, &gy, &gz);
```

```
GYRX = gx/sensitif;
```

```

GYRY = gy/sensitif;
GYRZ = gz/sensitif;
}

void setup()
{
Serial.begin(9600);
Wire.begin();
gy_521.initialize();
sensitif=250;
}

void loop()
{
acc();
gyro();
Serial.print("ax/ay/gx/gy/gz: \t");
Serial.print(ACCX); Serial.print("\t");
Serial.print(ACCY); Serial.print("\t");
Serial.print(GYRX); Serial.print("\t");
Serial.print(GYRY); Serial.print("\t");
Serial.println(GYRZ);
}

```

Pada pengujian diatas sensor GY-521 akan menampilkan nilai dari accelero dan gyro pada sensor.

#### **IV.2.3 Pengujian Driver Motor**

Pengujian ini dilakukan dengan cara menghubungkan driver dengan motor dc, selanjutnya arduino akan mengontrol keluaran arus pada driver motor agar dapat mengendalikan percepatan dan arah perputaran motor. Berikut potongan listing program yang digunakan.

// Motor 1

```
int motor1a = 2;
```

```
int motor2a = 3;
```

// Motor 2

```
int motor1b = 4;
```

```
int motor2b = 6;
```

```
void maju()
```

```
{
```

```
  digitalWrite(motor1a, LOW);
```

```
  digitalWrite(motor2a, HIGH);
```

```
  digitalWrite(motor1b, LOW);
```

```
  digitalWrite(motor2b, HIGH);
```

```
  Serial.println("Motor Maju");
```

```
  Serial.println(" ");
```

```
}
```

```
void mundur()
{
    digitalWrite(motor1a, HIGH);
    digitalWrite(motor2a, LOW);

    digitalWrite(motor1b, HIGH);
    digitalWrite(motor2b, LOW);

    Serial.println("Motor Mundur");
    Serial.println("  ");
}
```

```
void setup() {
    pinMode(motor1a, OUTPUT);
    pinMode(motor2a, OUTPUT);

    pinMode(motor1b, OUTPUT);
    pinMode(motor2b, OUTPUT);
}
```

```
void loop()
```

```
maju();delay(5000);

mundur();delay(5000);

}
```

#### IV.2.4 Pengujian Komunikasi Serial Bluetooth

Pengujian ini dilakukan dengan menghubungkan *bluetooth* pada arduino dengan android, selanjutnya mempairingkan antara perangkat dan mengirimkan informasi baik berupa data, Berikut potongan listing program yang digunakan:

```
char val;

void setup()
{
    pinMode(6,OUTPUT);
    Serial.begin(9600);
}

void loop()
{
    if(Serial.available())
    {
        val = Serial.read();
        Serial.println(val);
    }
    if(val == 'I')
```

```
{  
analogWrite (6;HIGH);  
}  
  
else  
{  
analogWrite (6,LOW);  
}  
}
```

Pengujian ini dilakukan untuk melihat koneksi antar perangkat sudah terjadi atau belum. Apabila sudah terhubung maka data akan dapat dikirimkan ke arduino melalui *smartphone android*.

#### IV.2.5 Pengujian Interface

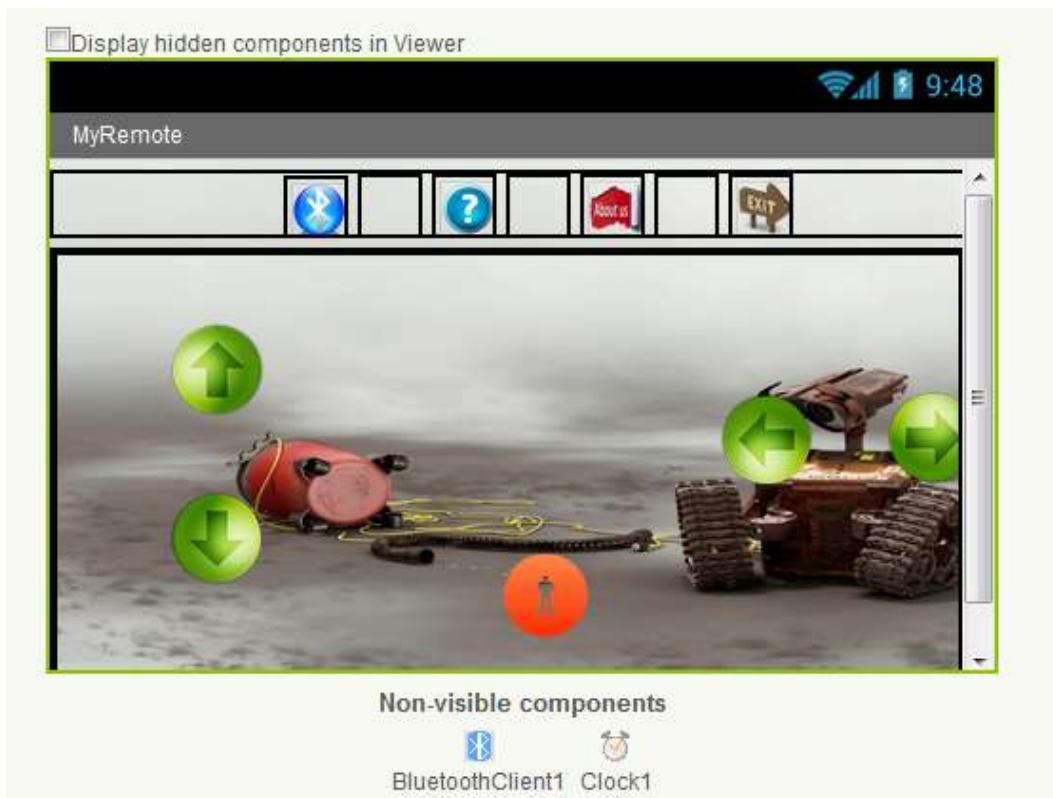
Adapun hasil pengujian yang dilakukan adalah sebuah perangkat lunak android yang dirancang dan diprogram dengan *App Inventor 2*. Tahapan pengujian interface yang dilakukan penulis adalah sebagai berikut:

- Sebelum pengujian perangkat, hidupkan kedua *bluetooth* yang berada pada *smartphone android* dan pada alat.
- Hubungkan antara *Bluetooth android* ke modul *bluetooth*.
- Apabila sudah terhubung buka aplikasi yang dirancang dan di program dengan *App Inventor 2*.
- Setelah aplikasi dijalankan pilih *bluetooth* pada aplikasi dan pilih *bluetooth* yang sudah dipairing sebelumnya.

- Apabila sudah terhubung maka *smartphone android* akan dapat mengirimkan data ke arduino.

#### **IV.2.5.1 Tampilan Menu Utama**

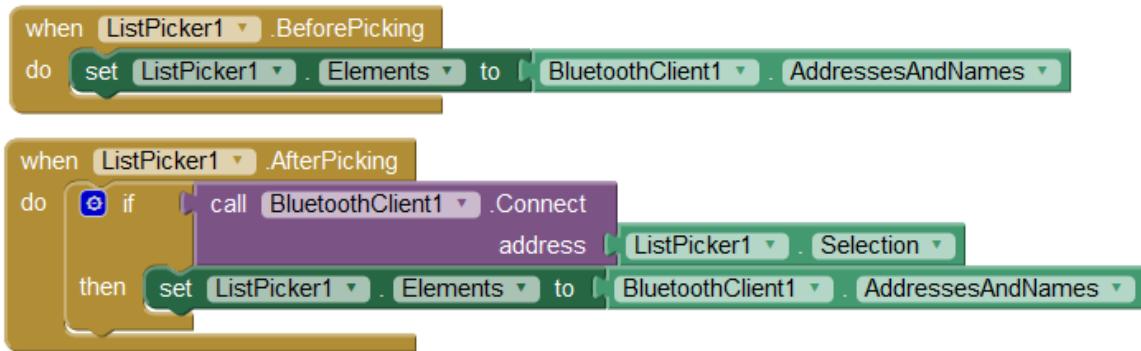
Tampilan menu utama pada aplikasi ini adalah tampilan yang pertama kali muncul ketika user membuka programnya. Pada tampilan ini terdapat beberapa menu yang dapat dipilih user. Tampilan menu utama dapat dilihat pada gambar berikut :



**Gambar IV.2 Tampilan Menu Utama**

Pada menu di atas terdapat beberapa tombol pada bagian atas aplikasi dan bagian bawah yang merupakan tombol navigasi pada perangkat robot keseimbangan menggunakan metode PID.

Menu **Bluetooth** berfungsi untuk melakukan koneksi antara smartphone ke perangkat arduino. Untuk membuat perintah pada *App Inventor 2* digunakan *block code*. *Block code* adalah serangkaian alur perintah pemrograman pada *App Inventor* yang berbentuk *block puzzle*. Adapun perintah *block code* nya adalah sebagai berikut :



**Gambar IV.3 Perintah Block Code Menu Start**

Penjelasan dari *block code* diatas adalah apabila *ListPicker1* diklik, maka akan keluar list address dan nama bluetooth.

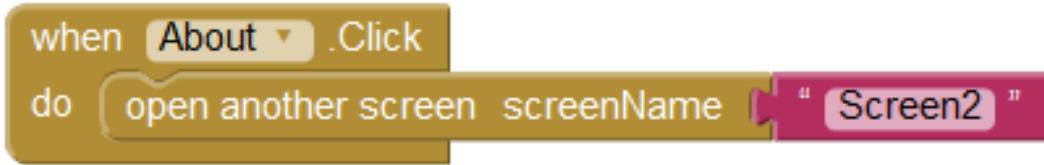
Menu **Tanda tanya (Bantuan)** ini berfungsi untuk masuk ke *screen* yang berisi tentang fungsi-fungsi tombol pada aplikasi. Adapun perintah *blockcode* nya adalah sebagai berikut :



**Gambar IV.4 Perintah Block Code Menu Bantuan**

Penjelasan dari *block code* diatas adalah apabila *button bantuan* di klik, maka akan masuk ke tampilan “Screen3” yang merupakan *screen* tampilan penjelasan tentang menu bantuan.

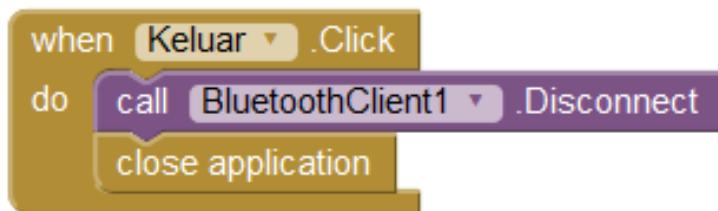
Icon **about us** berfungsi untuk masuk menuju *screen* yang berisi tentang penulis dan pembuat aplikasi perangkat robot keseimbangan dengan menggunakan metode PID ini. *Block code* nya adalah sebagai berikut :



Gambar IV.5. Perintah **Block Code** Menu *About Us*

Penjelasan dari *block code* diatas adalah apabila *button “about us”* di klik, maka akan masuk ke tampilan “Screen2” yang merupakan *screen* tampilan penjelasan tentang penulis.

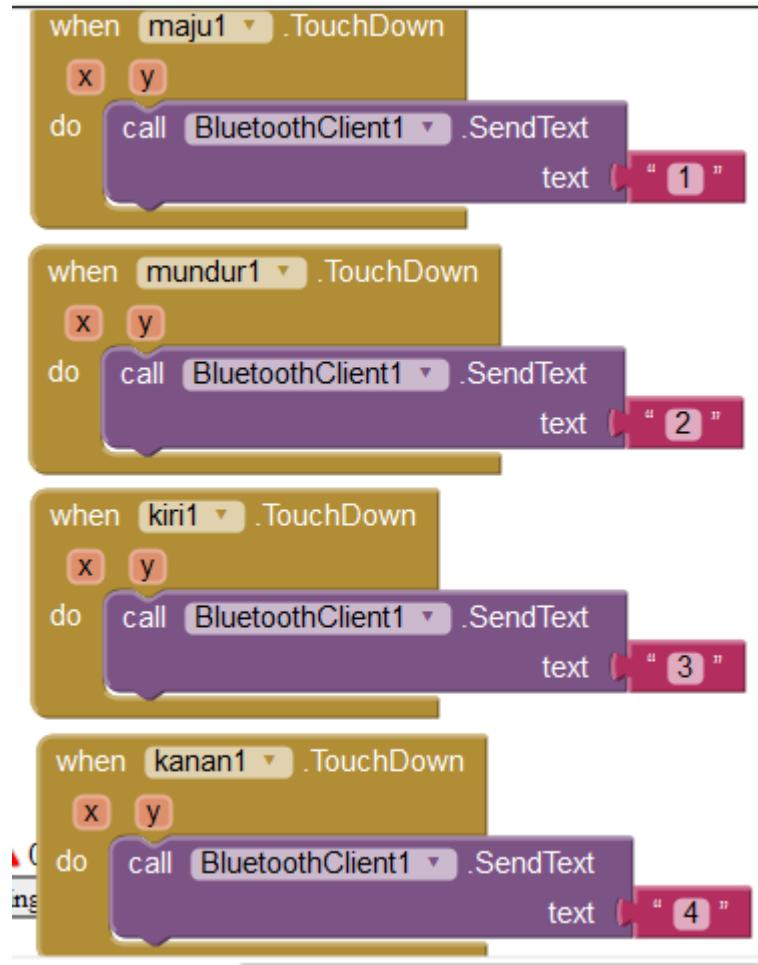
Sebuah *button “keluar”* berfungsi untuk keluar dari aplikasi. Adapun perintah *block code* nya seperti pada gambar IV.6 sebagai berikut :



Gambar IV.6. Perintah **Block Code** *Button Keluar*

Penjelasan *block code* diatas adalah ketika *button* ditekan maka aplikasi akan ditutup.

Tombol navigasi berfungsi untuk mengendalikan perangkat dengan menngunakan andriod. Adapun perintah *block code* nya seperti gambar IV.7 sebagai berikut :

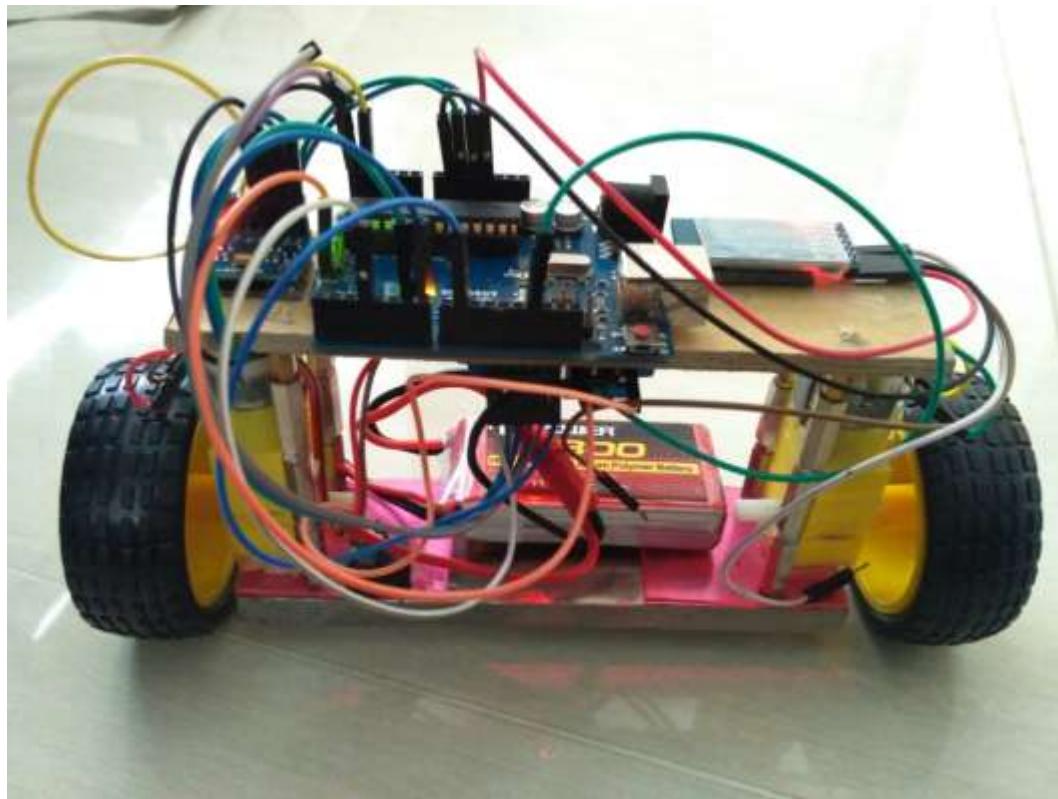


**Gambar IV.7. Perintah Block Code Tombol Navigasi**

Penjelasan *block code* di atas adalah ketika tombol atas di tekan, maka *smartphone* akan mengirimkan data “1” ke perangkat. Apabila tombol bawah di tekan, maka *smartphone* akan mengirimkan data “2” ke perangkat. Apabila tombol kiri di tekan, maka *smartphone* akan mengirimkan data “3” ke perangkat. Dan apabila tombol kanan di tekan, maka *smartphone* akan mengirimkan data “4” ke perangkat.

#### **IV.2.6 Pengujian Sistem Secara Keseluruhan**

Hasil perancangan keseluruhan alat robot keseimbangan menggunakan metode PID dapat dilihat pada gambar IV.8 berikut ini :



**Gambar IV.8 Rancangan Alat Pada Saat Pemasangan**

Perancangan robot keseimbangan menggunakan metode PID ini terdiri dari arduino uno, sensor GY-521 (MPU 6050), *bluetooth* HC – 05, *driver motor*, *motor dc*, dan *smartphone Android*.

Langkah awal pengujian adalah dengan menghidupkan Arduino Uno dengan menghubungkannya dengan power supply, selanjutnya buka aplikasi pada perangkat *smartphone android* dan kemudian melakukan pairing antar perangkat dengan menghubungkan perangkat android dengan perangkat minimum sistem. Letakkan robot keseimbangan pada permukaan yang rata.

Selanjutnya kita akan melakukan pengujian terhadap kondisi kemiringan dan kemampuan robot dalam menyeimbangkan dirinya sendiri.

**Tabel IV.1 Hasil Pengujian Pada Robot Keseimbangan**

ax/ay/gx/gy/gz:	0	14	0	-50	-2
SPEED :					
-255					
ax/ay/gx/gy/gz:	0	31	-1	74	-9
SPEED :					
-255					
ax/ay/gx/gy/gz:	1	13	-6	105	4
SPEED :					
-255					
ax/ay/gx/gy/gz:	0	0	0	-40	-5
SPEED :					
-9					
ax/ay/gx/gy/gz:	-1	16	1	-101	-1
SPEED :					
-255					
ax/ay/gx/gy/gz:	-1	21	-5	-46	22
SPEED :					
-255					
ax/ay/gx/gy/gz:	1	-1	2	-65	-14
SPEED :					
12					
ax/ay/gx/gy/gz:	0	-4	1	-131	-5
SPEED :					
106					
ax/ay/gx/gy/gz:	0	-10	11	-131	24
SPEED :					
255					
ax/ay/gx/gy/gz:	2	-9	11	-114	14
SPEED :					
243					

Pengujian diatas dilakukan dengan cara menghidupkan body robot yang terhubung dengan komputer dan data di ambil melalui serial monitor pada Arduino IDE. Pada tabel tersebut tampak robot berusaha menyeimbangkan dirinya dengan mengubah keluaran output PWM sesuai dengan kebutuhan agar robot tidak jatuh.

Hal berikutnya yang dilakukan adalah menentukan nilai kontrol PID. hal ini bertujuan agar robot mampu menyeimbangkan dirinya dengan baik. Diperlukan uji coba yang terus menerus untuk dapat menentukan nilai Kp, Ki, dan Kd yang tepat, dikarenakan bobot robot yang tidak sama dengan yang lain, serta bentuk body robot yang mempengaruhi titik tumpu pada robot keseimbangan.

### **IV.3. Kelebihan dan Kekurangan**

Pada perancangan dan implementasi robot keseimbangan dengan menggunakan metode PID ini masih jauh dari sempurna. Robot keseimbangan dengan menggunakan metode PID ini memiliki beberapa kelebihan dan kekurangan, diantaranya sebagai berikut :

#### Kelebihan

Adapun beberapa kelebihan yang dimiliki dari alat ini adalah :

1. Cara penggunaan dan pemakaian alat yang cukup mudah mudah
  2. Dapat memudahkan kita dalam mengontrol pergerakan robot melalui aplikasi yang telah dirancang pada *smartphone*.
  3. Dapat mengontrol robot dari jarak yang cukup jauh, dikarenakan menggunakan bluetooth sebagai media transfer datanya.
- 
- Kekurangan

Sedangkan beberapa kekurangan yang dimiliki alat ini adalah :

1. Kecepatan motor masih belum stabil. Hal ini dikarenakan motor yang kurang baik dalam menerima perubahan nilai tegangan yang diberikan oleh mikrokontroler.
2. Robot keseimbangan ini masih menggunakan baterai lipo berkapasitas rendah yakni 1300Mah sehingga hanya bertahan sementara saat alat dihidupkan.
3. Masih seringnya robot kehilangan posisi keseimbangannya dikarenakan kurangnya respon motor terhadap nilai yang diberikan oleh mikrokontroler.