IQBAL SANTOSA
6702180049
Kelompok Jari Bengkok

MODUL 4 Sistem Kendali PID Kasus P

1. JUDUL PRAKTIKUM

Sistem Kendali PID Kasus P

2. MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

1. Mahasiswa dapat memahami fungsi dan cara kerja PID pada motor DC
2. Mahasiswa dapat membuat program sistem kendali berbasis PID dengan error yang dihubungkan dengan konstanta proporsional

3. PARAMETER PENILAIAN

No.	Parameter	Persentase (%)
1.	Lembar Penilaian Praktikum	40%
2.	Jurnal/Laporan Praktikum	60%

4. PERALATAN DAN BAHAN

Alat dan Bahan :

1. Robot Kit Line Follower
2. Baterai LiPo 2-Cell 1300 mAh
3. Kabel Mini-USB
4. Arduino Nano
5. Battery Checker
6. Battery Balancer

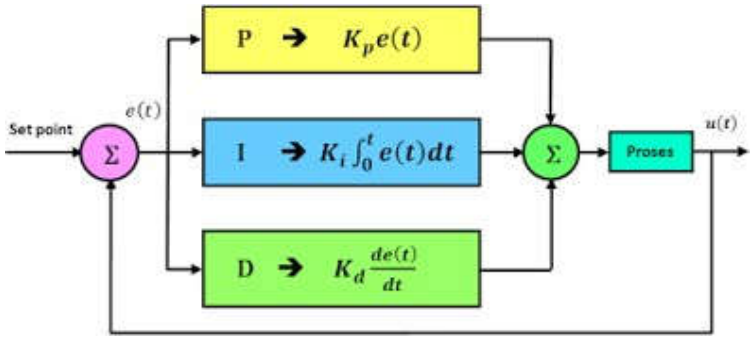
Perangkat Lunak :

1. Software IDE Arduino
2. Software Proteus (untuk simulasi)

5. TEORI DASAR

5.1. Sistem Kendali PID

Teknik kendali PID adalah pengendali yang merupakan gabungan antara aksi kendali proporsional ditambah aksi kendali integral ditambah aksi kendali derivatif/turunan (Ogata, 1996). PID merupakan kependekan dari *proportional integral derivative*. Kombinasi ketiga jenis aksi kendali ini bertujuan untuk saling melengkapi kekurangan-kekurangan dari masing-masing aksi kendali. Untuk memudahkan dalam memahami konsep teknik kendali PID silakan menyermati diagram blok pengendali PID pada gambar 1 di bawah ini.



Gambar 1. Diagram blok pengendali PID

Dalam aksi kendali PID, ada beberapa parameter variabel (dapat diubah/berubah) yang dapat dimanipulasi untuk tujuan menghasilkan aksi kendali terbaik dalam aplikasinya. Cara manipulasi parameter ini sering dinamakan dengan Manipulated Variable (MV). Dalam notasi matematikanya dapat ditulis dengan MV(t) atau u(t). Berikut persamaan matematik kendali PID.

$$u(t) = MV(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t) \quad \ldots\ldots\ldots\ldots \quad (1)$$

$$K_i = \frac{K_p}{T_i} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$

$$K_d = K_p T_d \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

Persamaan (2) dan (3) disubtitusikan ke dalam persamaan (1) makan akan menjadi:

$$u(t) = MV(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau)d\tau + K_p T_d \frac{d}{dt} e(t) \quad \ldots\ldots\ldots \quad (4)$$

$$u(t) = MV(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau)d\tau + T_d \frac{d}{dt} e(t) \right) \quad \ldots\ldots\ldots \quad (5)$$

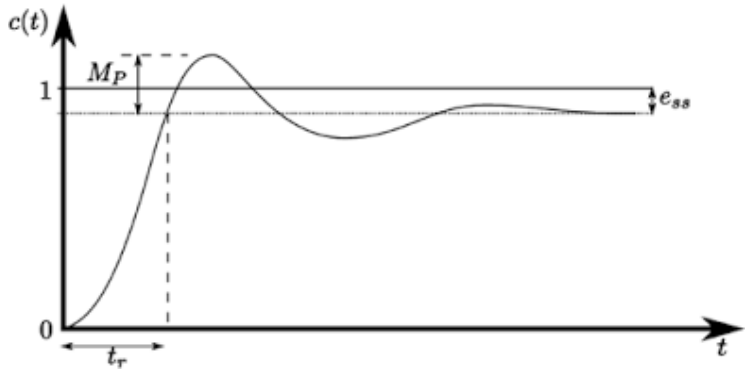
Apabila kita terapkan transformasi Laplace pada persamaan (4) di atas, maka penulisannya adalah sebagai berikut:

$$G(s) = K_p + \frac{K_i}{s} + K_d S \quad \text{.....} \quad (6)$$

$$G(s) = \frac{K_d s^2 + K_p s + K_i}{s} \quad \text{.....} \quad (7)$$

Respon Sistem Kendali PID

Gambar 2 di bawah ini merupakan ilustrasi grafik respon sistem kendali PID.

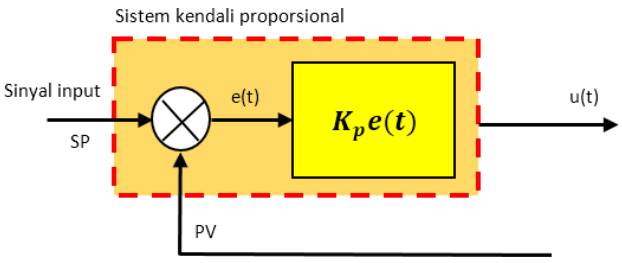


Gambar 2. Sinyal respon sistem kendali PID
MP = maksimum overshoot, ess = steady state error, tr = rise time

Aksi kendali PID memiliki karakter mampu mengurangi rise time (tr), mengurangi overshoot maksimum (MP), dan menghilangkan kesalahan keadaan tunak atau steady-state errors (ess).

5.2. Pengertian Sistem Kendali PID Kasus P (*Proportional*)

Aksi kendali proporsional (P) adalah aksi kendali yang memiliki karakter dapat mengurangi waktu naik (rise time), tetapi tidak menghilangkan kesalahan keadaan tunak (steady satate error).



Gambar 3. Diagram blok sistem kendali proporsional (P)

Persamaan hubungan antara keluaran sistem u(t) dengan sinyal *error* e(t) pada aksi kendali proporsional adalah sebagai berikut.

$$u(t) = K_p e(t)$$ ..... (1)

Sedangkan persamaan sinyal *error* -nya adalah:

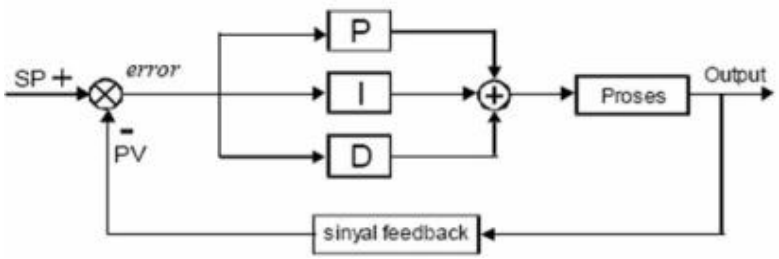
$$e(t) = SP - PV$$ ..... (2)

Pada praktikum ini nilai PV (*process value*) adalah error dengan setpoint (SP) sensor dianggap 0.

Dimana,
$u(t)$ = sinyal keluaran sistem kendali
K_p = Konstanta penguatan proporsional
$e(t)$ = sinyal *error*
SP = *Set Point*
PV = *Process Value* (nilai aktual)
t = waktu

5.3. Aplikasi PID pada Robot Line Follower

Sistem kendali PID ini bertujuan untuk menentukan paramater aksi kendali Proportional, Integratif, Derivatif pada robot line follower. Proses ini dapat dilakukan dengan cara trial and error . Keunggulan cara ini plant tidak perlu diidentifikasi dan membuat model matematis plant. Hanya dengan cara mencoba memberikan konstanta P-I-D pada formula PID ehingga di peroleh hasil yang optimal, dengan mengacu pada karakteristik masing-masing kontrol P-I-D.
Tujuan penggunaan sistem kendali PID adalah untuk mengolah suatu sinyal kesalahan atau error, nilai error tersebut diolah dengan formula PID untuk dijadikan suatu sinyal kendali atau sinyal kontrol yang akan diteruskan ke aktuator. Diagram blok sistem umpan balik loop tertutup pada perancangan sistem kendali PID pada robot line follower dapat dilihat pada gambar berikut ini:

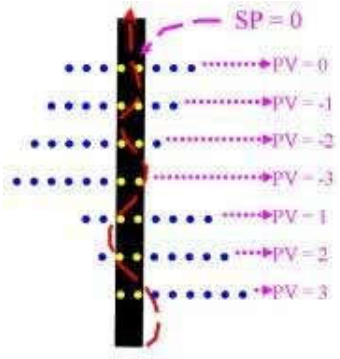


Dari blok diagram di atas dapat dijelaskan sebagai berikut

1. SP = *Set point*, suatu parameter nilai acuan atau nilai yang diinginkan.
2. PV = *Present Value*, nilai bobot pembacaan sensor saat itu atau variabel terukur yang di umpan balik oleh sensor (sinyal *feedback* dari sensor).
3. Error = nilai kesalahan, deviasi atau simpangan antar variabel terukur atau bobot sensor (PV) dengan nilai acuan (SP)

$$error = SP - PV$$

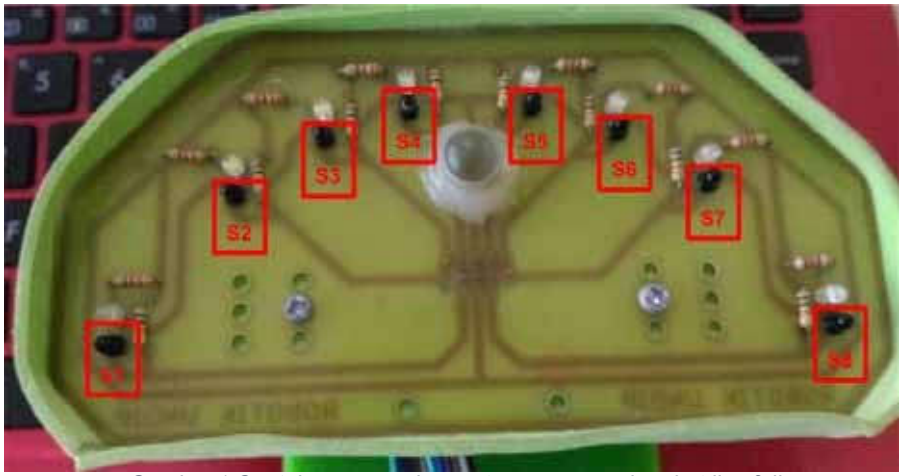
Ilustrasi pemberian bobot sensor (nilai kesalahan pembacaan sensor) pada robot line follower dapat dilihat pada gambar berikut.



PROSEDUR PRAKTIKUM

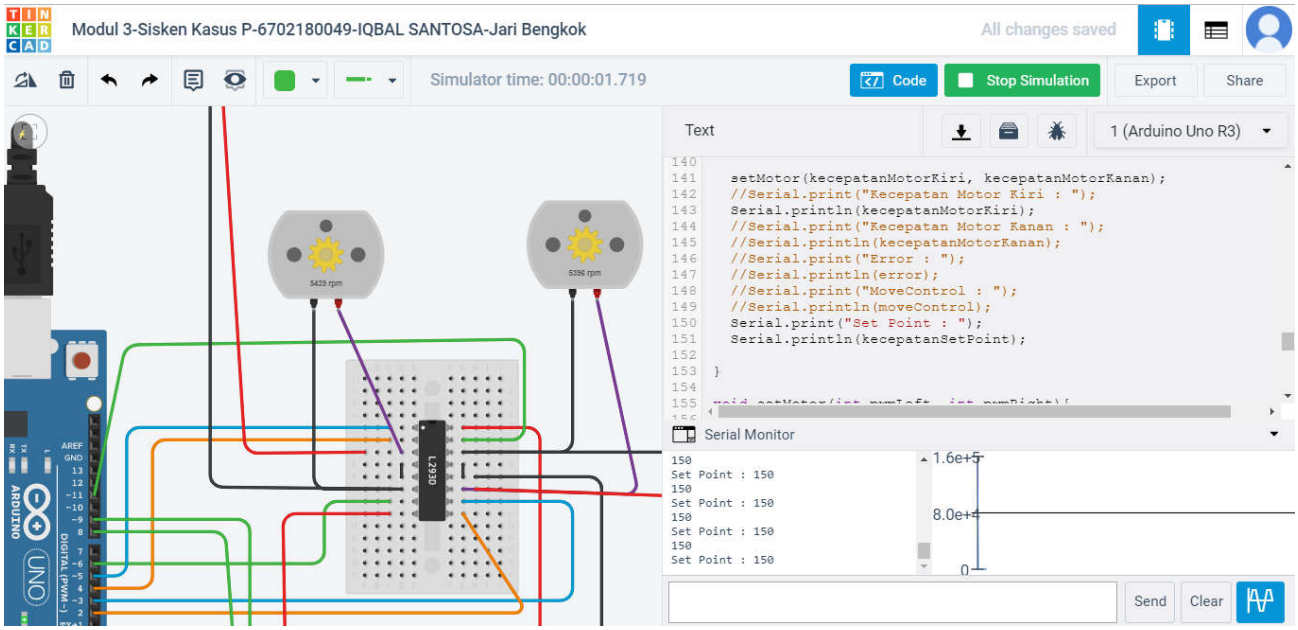
A. Percobaan dalam praktikum

1. Kasus Percobaan



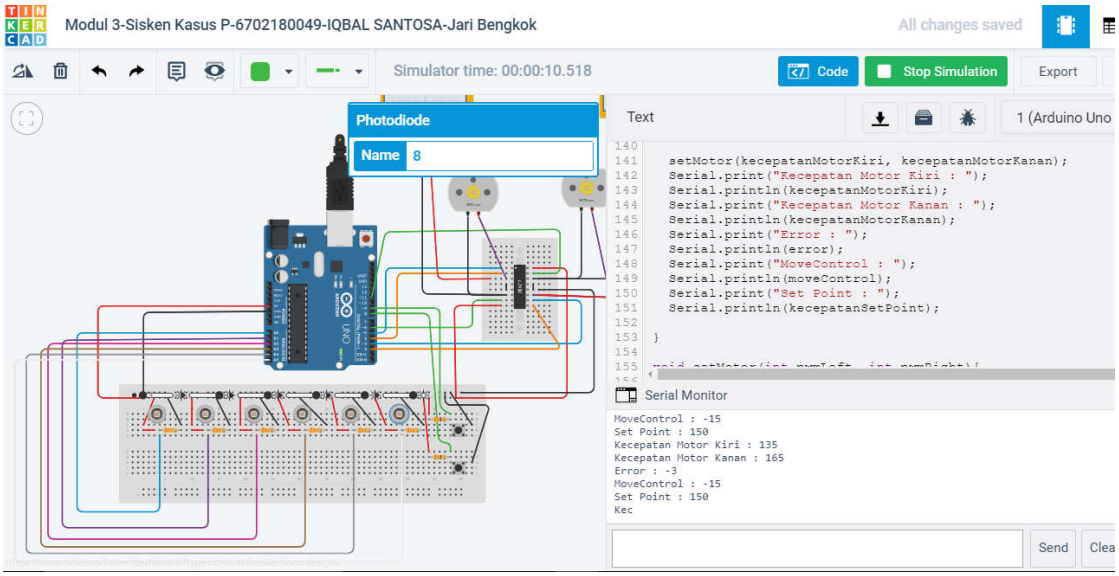
Gambar 1 Contoh susunan dan urutan sensor pada robot line follower.

a. Modifikasi program sistem kendali pada praktikum sebelumnya dengan menambahkan sebuah kondisi string berikut dengan menambahkan sebuah variabel dengan tipe int Kp dengan nilai awal 5, int moveControl dengan nilai awal 0, int error dengan nilai awal 0, int kecepatanMotorKanan dengan nilai awal 0, int kecepatanMotorKiri dengan nilai awal 0, int kecepatanSetPoint dengan nilai awal 150.

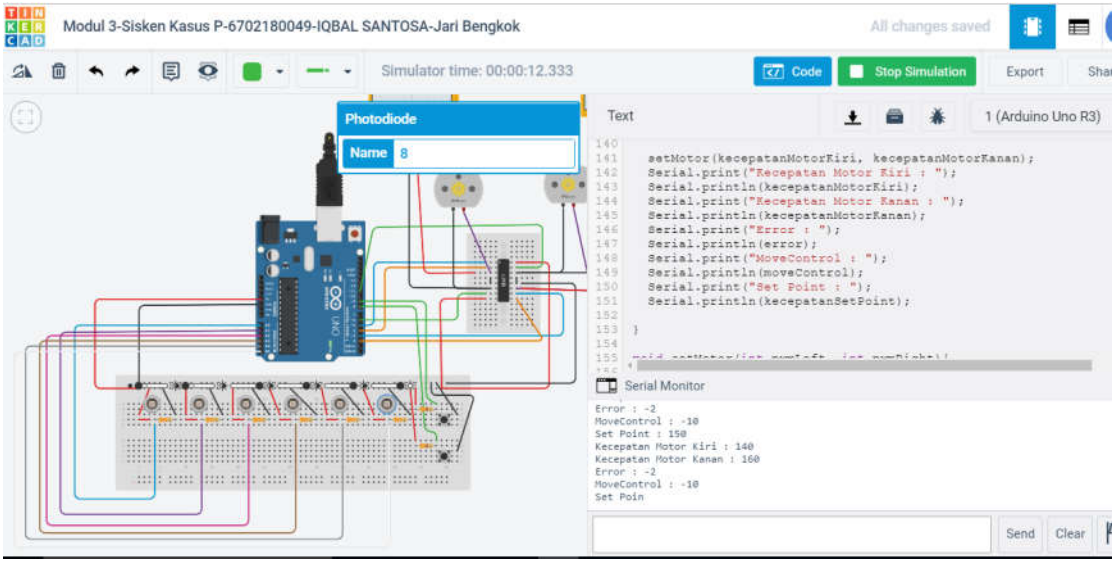


a. *Screenshoot* keluaran serial monitor untuk setiap kondisi. Cetak dan tempelkan pada buku jurnal praktikum.

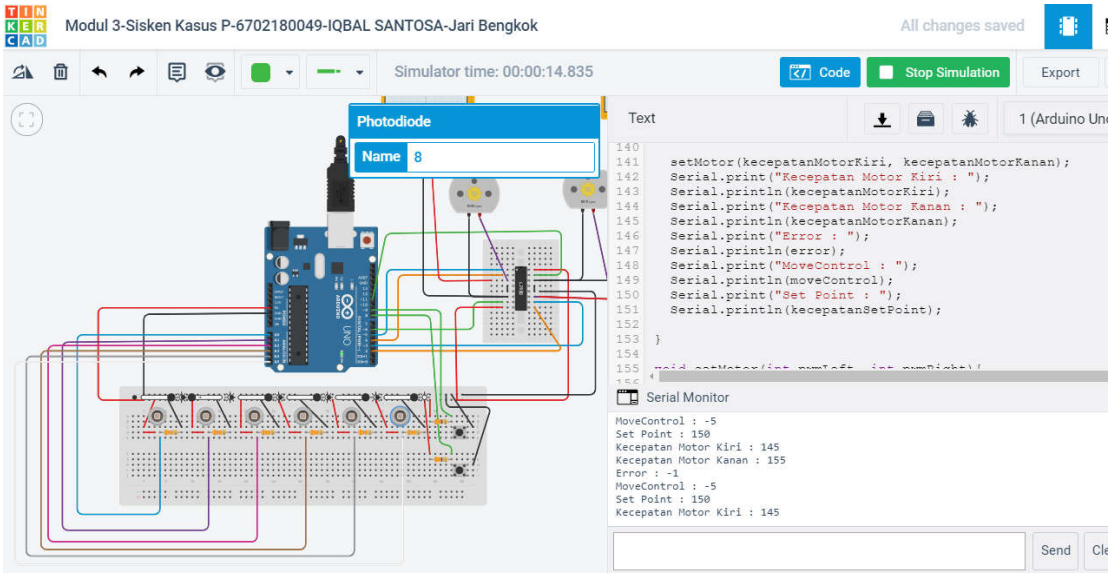
- Jika kondisi sensor "100000", error = -3, print di serial monitor error = -3,



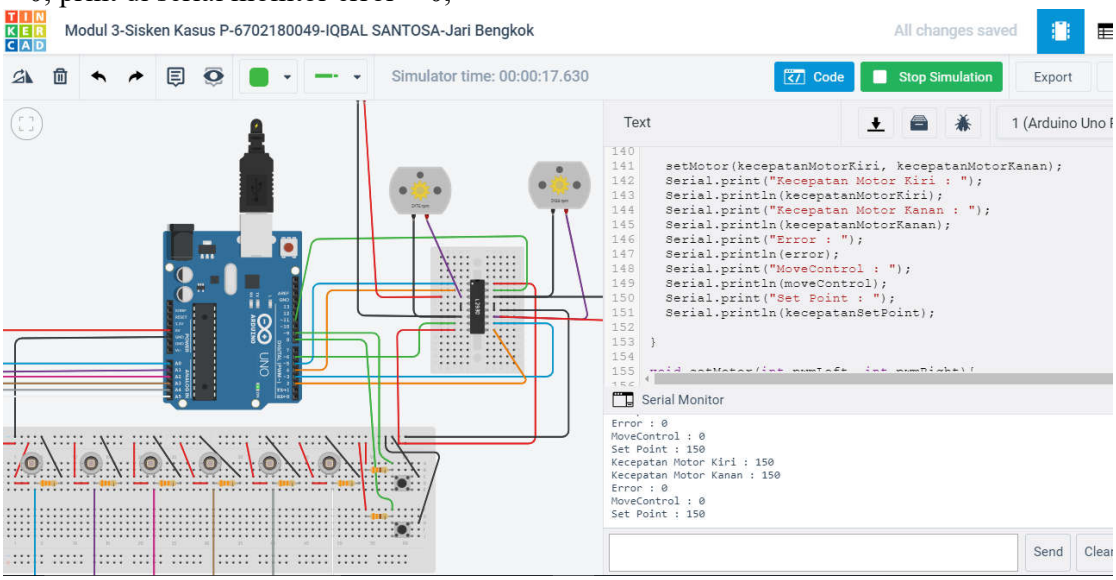
- Jika kondisi sensor "110000", error = -2, print di serial monitor error = -2,



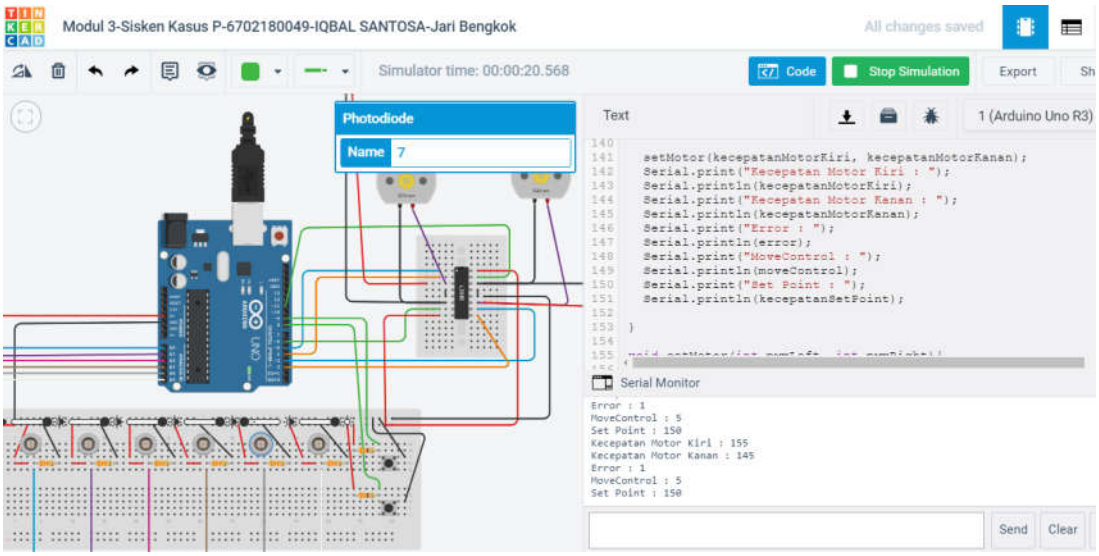
- Jika kondisi sensor "010000", error = -1, print di serial monitor error = -1,



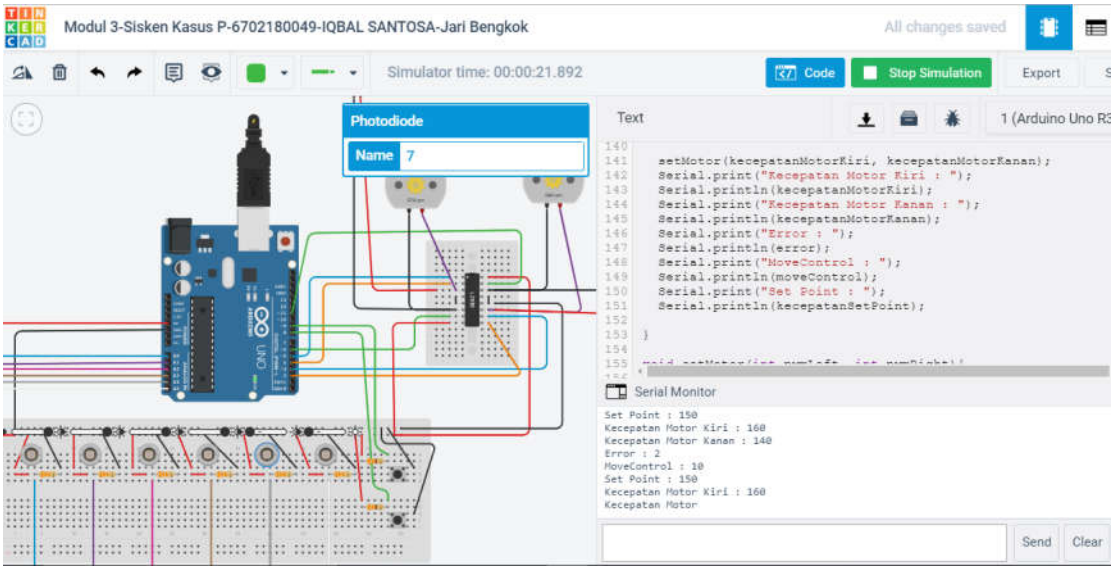
- Jika kondisi sensor "001000" atau "001100" atau "000100" , error = 0, print di serial monitor error = 0,



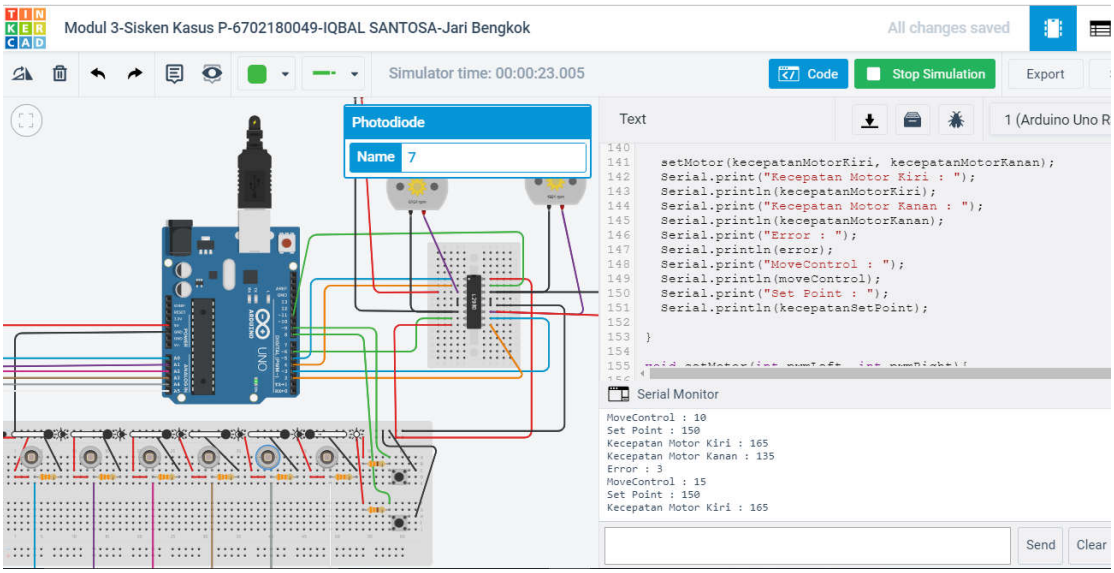
- Jika kondisi sensor "000010", error = 1, print di serial monitor error = 1,



- Jika kondisi sensor "000011", error = 2, print di serial monitor error = 2,



- Jika kondisi sensor "000001", error = 3, print di serial monitor error = 3,



b. *Program*

```
/*************************************************/

//Modul Praktikum 3 - Sistem kendali Kasus P
//Nama          : IQBAL SANTOSA
//Kelompok      : 3
//Nama Tim      : Jari Bengkok
//Anggota 1     : Iqbal Santosa
//Versi Program : 1.0

/*************************************************/
//Deklarasi sensor
int sensor1 = A0;
int sensor2 = A1;
int sensor3 = A2;
int sensor4 = A3;
int sensor5 = A4;
int sensor6 = A5;
int button1 = 8;
int button2 = 9;
int baca_sensor[6];

//Variabel Motor
int pinEnable = 4; //Harus di set high jika ingin mengaktifkan
int pinEnable2 = 2; //Harus di set high jika ingin mengaktifkan
```

IQBAL SANTOSA
6702180049
Kelompok Jari Bengkok

```
//Variabel Motor kiri
int motor_kiri1 = 5; //input motor driver 1 L293D
int motor_kiri2 = 6; //input motor driver 2 L293D
//Variabel motor kanan
int motor_kanan1 = 3; //input motor driver 3 L293D
int motor_kanan2 = 11;//input motor driver 4 L293D

int maxpwm = 255;
int x;

//Variabel dan konstanta untuk persamaan pid
int kp = 5, ki = 0, kd = 0;
int lastError = 0;
int error = 0;
int rate_i, rate_d;
int kecepatanMotorKanan, kecepatanMotorKiri, kecepatanSetPoint = 150;

void setup()
{
 //Pin Mode input sensor
 pinMode(sensor1, INPUT); //set pin A0 sebagai input
 pinMode(sensor2, INPUT); //set pin A1 sebagai input
 pinMode(sensor3, INPUT); //set pin A2 sebagai input
 pinMode(sensor4, INPUT); //set pin A3 sebagai input
 pinMode(sensor5, INPUT); //set pin A4 sebagai input
 pinMode(sensor6, INPUT); //set pin A5 sebagai input

 //pin mode output motor
 pinMode(pinEnable, OUTPUT);
 pinMode(pinEnable2, OUTPUT);
 pinMode(motor_kiri1, OUTPUT);
 pinMode(motor_kiri2, OUTPUT);
 pinMode(motor_kanan1, OUTPUT);
 pinMode(motor_kanan2, OUTPUT);

 //Insialisasi komunikasi serial
 Serial.begin(9600); //default 9600
}
```

```
void readsensor(){ //fungsi untuk membaca sensor dan menyimpan nilai
baca_sensor[0] = analogRead(sensor1);
baca_sensor[1] = analogRead(sensor2);
baca_sensor[2] = analogRead(sensor3);
baca_sensor[3] = analogRead(sensor4);
baca_sensor[4] = analogRead(sensor5);
baca_sensor[5] = analogRead(sensor6);

  delay(50);

  for(x=0; x<=5; x++){
  }
}

void loop()
{
  readsensor();
  //Serial.print("Nilai last error:");
  //Serial.println(lastError);
  //catatan saat semua sensor mendeteksi gelap kedua motor jalan
  if(baca_sensor[0] < 34 && baca_sensor[1] > 34 &&
   baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
   baca_sensor[4] > 34 && baca_sensor[5] > 34) {
    error = -3;
  }
  if(baca_sensor[0] < 34 && baca_sensor[1] < 34 &&
   baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
   baca_sensor[4] > 34 && baca_sensor[5] > 34) {
    error = -2;
  }
   if(baca_sensor[0] > 34 && baca_sensor[1] < 34 &&
   baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
   baca_sensor[4] > 34 && baca_sensor[5] > 34) {
    error = -1;
  }
  if(baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
   baca_sensor[2] < 34 && baca_sensor[3] < 34 ||
   baca_sensor[2] < 34 || baca_sensor[3] < 34 &&
     baca_sensor[4] > 34 && baca_sensor[5] > 34){
     error = 0;
```

```
  }
  if(baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
   baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
   baca_sensor[4] < 34 && baca_sensor[5] > 34) {
    error = 1;
  }
  if(baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
   baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
   baca_sensor[4] < 34 && baca_sensor[5] < 34) {
    error = 2;
  }
  if(baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
   baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
   baca_sensor[4] > 34 && baca_sensor[5] < 34) {
    error = 3;
  }
  //nilai terendah photodioda 33 paling gelap
  //nilai tertinggi = 687 paling terang
  //aktifkan motor
  digitalWrite(pinEnable, HIGH);
  digitalWrite(pinEnable2, HIGH);
  //fungsi deteksi garis
  follow_line();
}

void follow_line(){
readsensor();

//fungsi MV
  rate_d = error - lastError;
  rate_i = error + lastError;
  lastError = error;
  int moveControl = (kp * error) + (ki * rate_i) + (kd * rate_d);

  kecepatanMotorKanan = kecepatanSetPoint - moveControl;
  kecepatanMotorKiri = kecepatanSetPoint + moveControl;

  setMotor(kecepatanMotorKiri, kecepatanMotorKanan);
  Serial.print("Kecepatan Motor Kiri : ");
```

```
  Serial.println(kecepatanMotorKiri);
  Serial.print("Kecepatan Motor Kanan : ");
  Serial.println(kecepatanMotorKanan);
  Serial.print("Error : ");
  Serial.println(error);
  Serial.print("MoveControl : ");
  Serial.println(moveControl);
  Serial.print("Set Point : ");
  Serial.println(kecepatanSetPoint);

}

void setMotor(int pwmLeft, int pwmRight){
  if (pwmLeft > maxpwm){
   pwmLeft = maxpwm;
  }
  else if (pwmLeft < -maxpwm){
   pwmLeft = -maxpwm;
  }

  if (pwmLeft < 0){
    pwmLeft *= -1;
    analogWrite(motor_kiri2, pwmLeft);
    analogWrite(motor_kiri1, 0);
    }
  else{
    analogWrite(motor_kiri2, 0);
    analogWrite(motor_kiri1, pwmLeft);
  }
  if (pwmRight > maxpwm){
    pwmRight = maxpwm;
  }
  else if (pwmRight < -maxpwm){
    pwmRight = -maxpwm;
  }
  if (pwmRight < 0){
    pwmRight *= -1;
    analogWrite(motor_kanan2, pwmRight);
    analogWrite(motor_kanan1, 0);
    }
```

```
  else{
    analogWrite(motor_kanan2, 0);
    analogWrite(motor_kanan1, pwmRight);
  }
}
```

c. Isi tabel kebenaran dari sistem pada Tabel 1 dan tuliskan pada buku jurnal praktikum.

Tabel 1Tabel Kebenaran Sistem Kendali

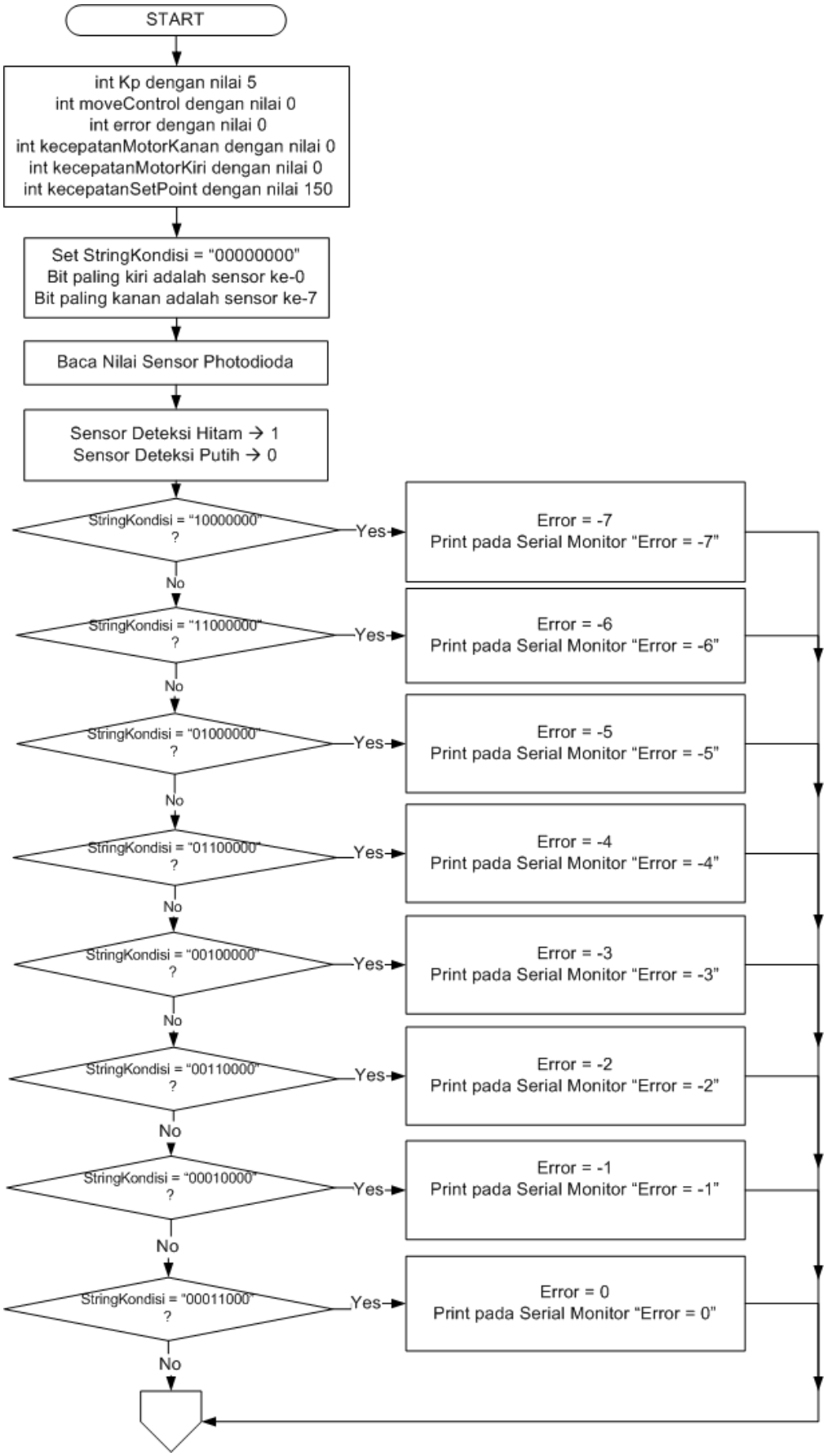
Sensor								Error	Nilai Setpoint	Analog Value	
0	1	2	3	4	5	6	7			Motor Kiri	Motor Kanan
	1	0	0	0	0	0		-3	150	135(4800RPM)	165(6200RPM)
	1	1	0	0	0	0		-2	150	140(5000RPM)	160(6000RPM)
	0	1	0	0	0	0		-1	150	145(5200RPM)	155(5800RPM)
	0	0	1	0	0	0		0	150	150(5600RPM)	150(5600RPM)
	0	0	1	1	0	0		0	150	150(5600RPM)	150(5600RPM)
	0	0	0	1	0	0		0	150	150(5600RPM)	150(5600RPM)
	0	0	0	0	1	0		1	150	155(5800RPM)	145(5200RPM)
	0	0	0	0	1	1		2	150	160(6000RPM)	140(5000RPM)
	0	0	0	0	0	1		3	150	165(6200RPM)	135(4800RPM)

d. Link github

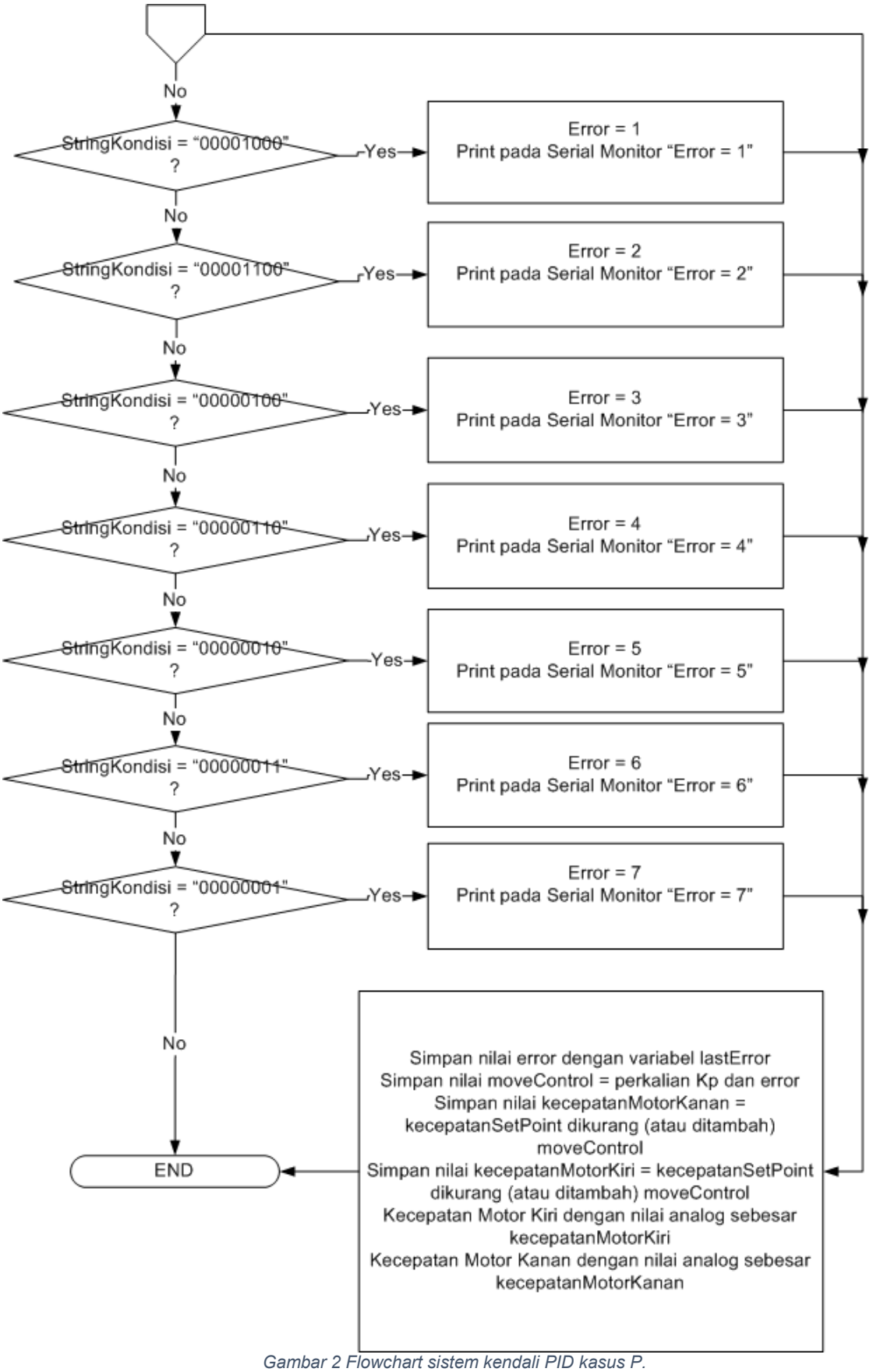
https://github.com/iqbalsantosa/iqbalsantosa-modul_3_sisken_kasus_p_6702180049_iqbal_santosa_jari_bengkok

e. Kesimpulan

Kita dapat membuat program sistem kendali berbasis PID dengan error yang dihubungkan dengan konstanta proporsional


START
int Kp dengan nilai 5
int moveControl dengan nilai 0
int error dengan nilai 0
int kecepatanMotorKanan dengan nilai 0
int kecepatanMotorKiri dengan nilai 0
int kecepatanSetPoint dengan nilai 150
Set StringKondisi = "00000000"
Bit paling kiri adalah sensor ke-0
Bit paling kanan adalah sensor ke-7
Baca Nilai Sensor Photodioda
Sensor Deteksi Hitam → 1
Sensor Deteksi Putih → 0
StringKondisi = "10000000" ?
Yes
Error = -7
Print pada Serial Monitor "Error = -7"
No
StringKondisi = "11000000" ?
Yes
Error = -6
Print pada Serial Monitor "Error = -6"
No
StringKondisi = "01000000" ?
Yes
Error = -5
Print pada Serial Monitor "Error = -5"
No
StringKondisi = "01100000" ?
Yes
Error = -4
Print pada Serial Monitor "Error = -4"
No
StringKondisi = "00100000" ?
Yes
Error = -3
Print pada Serial Monitor "Error = -3"
No
StringKondisi = "00110000" ?
Yes
Error = -2
Print pada Serial Monitor "Error = -2"
No
StringKondisi = "00010000" ?
Yes
Error = -1
Print pada Serial Monitor "Error = -1"
No
StringKondisi = "00011000" ?
Yes
Error = 0
Print pada Serial Monitor "Error = 0"
No

IQBAL SANTOSA
6702180049
Kelompok Jari Bengkok



Gambar 2 Flowchart sistem kendali PID kasus P.

6. Jurnal Praktikum

a. Jurnal pada Buku Praktikum harus memuat konten sebagai berikut :

- Judul Praktikum :
- Maksud dan Tujuan Praktikum :
- Peralatan dan Bahan Praktikum :
- Dasar Teori
- Foto Peralatan dan Bahan Praktikum :
- Hasil Praktikum (Tulis tangan kode program yang telah diberi komentar/penjelasan beserta foto hasil percobaan yang telah diberi nama dan NIM anggota kelompok)
- Kesimpulan Praktikum