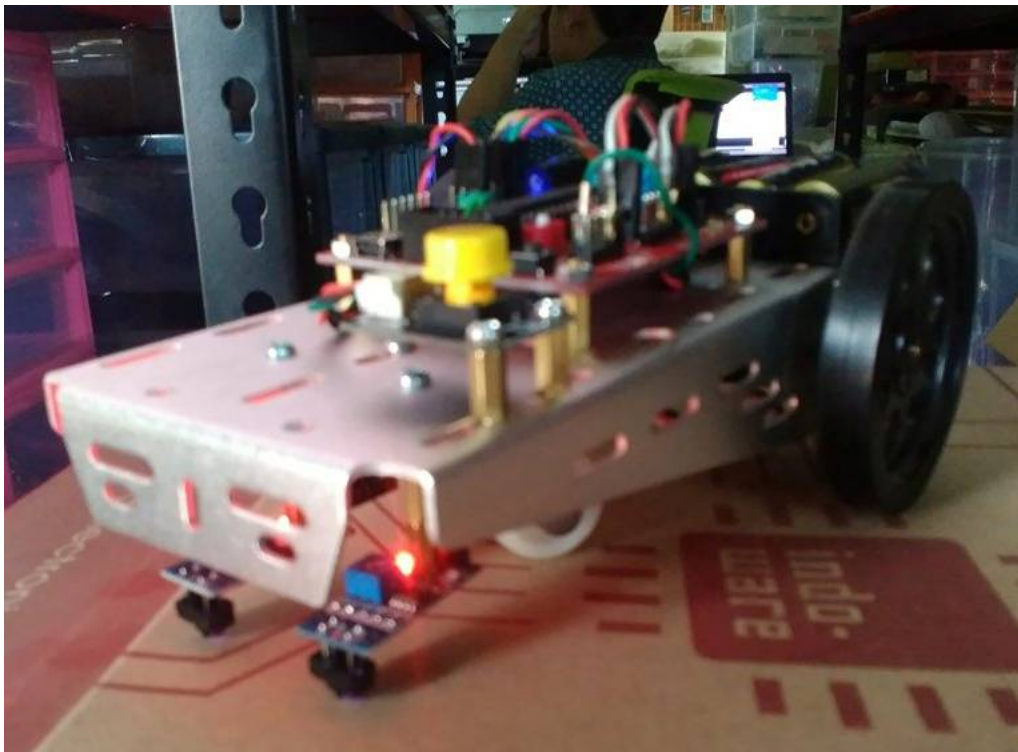


Robot LineFollower dengan Module Arduino Indo-Ware

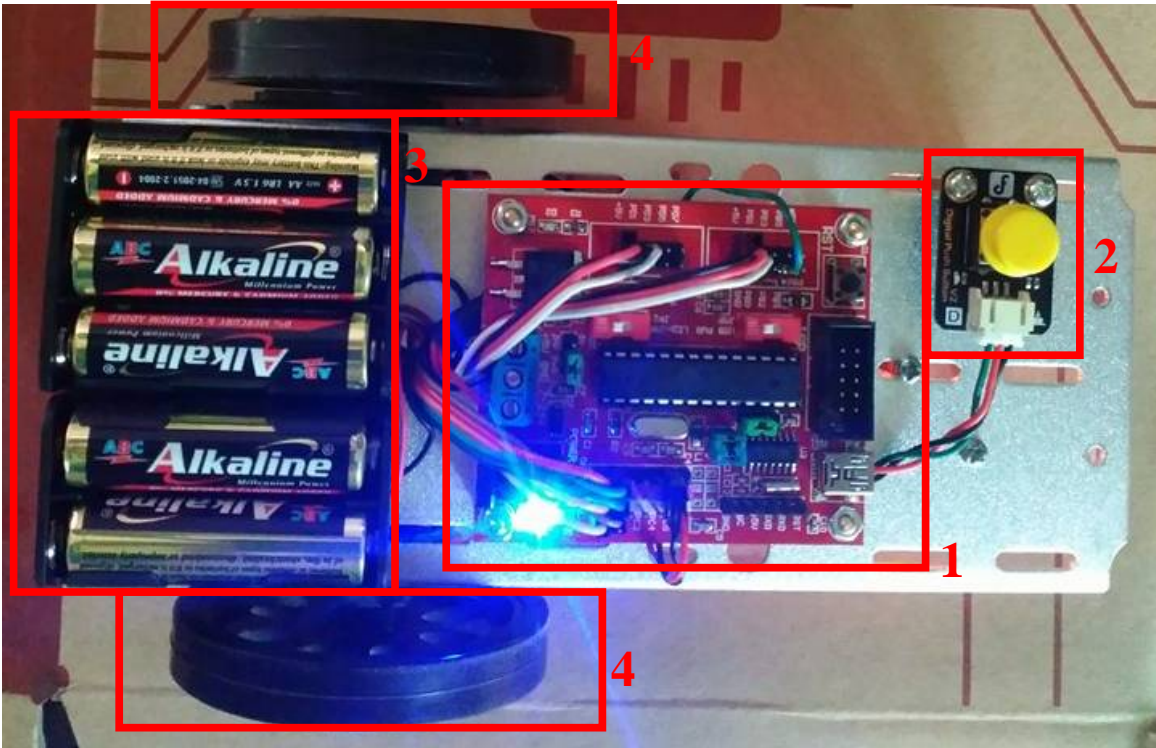
LineFollower adalah jenis robot beroda yang dapat mengikuti garis yang biasa dibuat warna hitam untuk garisnya, dan putih untuk bagian lapangannya. Perbedaan yang sangat kontras itu dijadikan acuan dalam membuat logika waktu memprogramnya.

Di dalam pembahasan kali ini Robot yang dibuat menggunakan Module Arduino produksi dari Indo-Ware, modul sensor garis dan motor sero sebagai penggeraknya. Berikut bentuk dari robot:

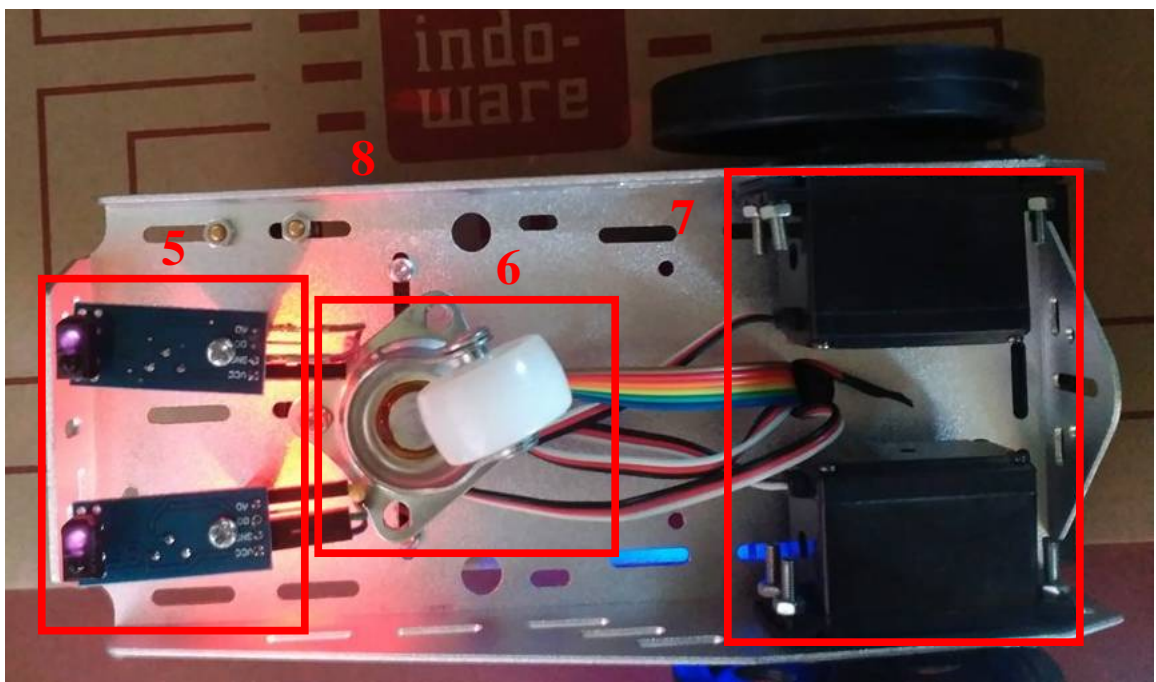


Gambar 1. Robot LineFollower

Berikut asing-masing bagian dari robot:



Gambar 2. Bagian Robot tampak atas



Gambar 3. Bagian Robot tampak bawah

Keterangan bagian robot:

1. ModuleMikrokontroler
2. PushBotton
3. Batteray
4. Roda belakang
5. Sensor Garis
6. Roda bebas depan
7. Motor Servo
8. Mekanik

Agar robot dapat berjalan pertama kita harus dapat membaca data dari sensor, kemudian mengolah data tersebut hingga dapat menjadi sebuah pemicu penggerak dari servo. Untuk dapat membaca sensor berikut ode program menggunakan arduino:

```
const int sensor1 = A0; //mendeklarasikan sensor1 sebagai PIN A0
const int sensor2 = A1; //mendeklarasikan sensor2 sebagai PIN A1

int data1 = 0; //sebuah wadah dengan nama data1 yang bertipe data integer dan diisi dengan nilai 0
int data2 = 0; //sebuah wadah dengan nama data2 yang bertipe data integer dan diisi dengan nilai 0

void setup() {
  Serial.begin(9600); //sebagai pengatur kecepatan baut ratemi krokontroler
}

void loop(){
  data1 = analogRead(sensor1); //baca sensor1 kemudian masukkan datanya ke dalam data1
  data2 = analogRead(sensor2); //baca sensor1 kemudian masukkan datanya ke dalam data2
  Serial.print("data 1 = "); //kirim ke PC "data 1 = "
  Serial.print(data1); //kirim data1 (yang dikirim adalah nilai yang sudah dimasukkan tadi)
  Serial.print(" data 2 = ");
  Serial.println(data2); //kirim data2 kemudian ganti baris
  delay(2); //tunda selama 2mS
}
```

Setelah kita bisa membaca data kemudian kita harus bisa mengkonversinya menjadi sebuah pemicu agar servo dapat bergerak. Servo ini berputar 360 derajat, dengan mengirim nilai 0-90 dia akan

berputar ke kiri, sedangkan jika data yang dikirim antara 91-180 maka dia akan berputar ke arah kanan. Dengan acuan tersebut dapat disimpulkan bahwa agar robot nantinya data bergerak maju mengikuti garis maka servo kanan dan kiri diberi data yang berbeda karena pemasangan mekanik servo kanan berbanding terbalik dengan servo kiri. Dalam pembahasan kali ini servo robot sebelah kanan dihubungkan ke PORT Digital 9, sedangkan servo robot sebelah kiri dihubungkan ke PORT Digital 3. Dengan demikian maka agar robot bergerak maju, servo kanan harus diberi nilai data antara 91-180 sedangkan servo kiri harus diberi data antara 90-0.

Dalam code program di sini digunakan libraryServo.h bawaan dari SoftwareArduino. Berikut code program untuk menggerakkan servo, namun sudah ditambah agar dapat menggerakkan 2 servo sekaligus.

```
#include<Servo.h>
Servo myservo1; // createservoobjecttocontrol a servo
Servo myservo2; // createservoobjecttocontrol a servo
int pos = 0; // variabelstoretheservoposition
void setup() {
  myservo1.attach(9); // attachtheservo on pin 9 totheservoobject
  myservo2.attach(3);
}
void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goesfrom 0 degreesto 180 degrees
    // insteps of 1 degree
    myservo1.write(pos); // tellservotogopositionvariable 'pos'
    myservo2.write(pos);
    delay(100); // waits 15ms for theservotoreachtheposition
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goesfrom 180 degreesto 0 degrees
    myservo1.write(pos); // tellservotogopositionvariable 'pos'
    myservo2.write(pos);
    delay(100); // waits 15ms for theservotoreachtheposition
  }
}
```

Setelah bisa membaca data dari sensor dan menggerakkan servo, maka keduanya kita gabungkan agar robot dapat berjalan mengikuti garis. Berikut contoh code programnya:

```

#include<Servo.h> //libraryservo
Servoservo_kanan; //membuat sebuah nama untuk servo, di sini dinamakan servo_kanan
Servoservo_kiri; // membuat sebuah nama untuk servo, di sini dinamakan servo_kanan
const int sensor1 = A0; //mendeklarasikan nama sensor1 sebagai PIN A0
const int sensor2 = A1; //mendeklarasikan nama sensor2 sebagai PIN A1
const int tombol = 12; //mendeklarasikan nama tombol sebagai PIN Digital 12
const int led = 7; //mendeklarasikan nama led sebagai PORT Digital 7
int data1 = 0; //membuat sebut tempat data dengan nama "data1" bertipe data integer dengan
nilai awal 0.
int data2 = 0; //membuat sebut tempat data dengan nama "data2" bertipe data integer dengan
nilai awal 0.
int kecepatan_kanan = 0; //membuat sebut tempat data dengan nama "kecepatan_kanan" bertipe
data integer dengan nilai awal 0.
int kecepatan_kiri = 0;
int motor_kanan = 0;
int motor_kiri = 0;
int start = 0;
void setup() {
servo_kanan.attach(9); //mendeklarasikan servo kanan terhubung pada PORT Digital 9
servo_kiri.attach(3); //mendeklarasikan servo kanan terhubung pada PORT Digital 3
Serial.begin(9600);
pinMode(led, OUTPUT); //mengkonfigurasikan PORT led sebagai OUTPUT
pinMode(tombol, INPUT); //mengkonfigurasikan PORT tombol sebagai INPUT
}
void baca_sensor(){ //sub bagian bernama baca sensor, di dalamnya terdapat program untuk
membaca sensor
    data1 = analogRead(sensor1); // baca sensor1 kemudian masukkan nilainya dalam data1
    data2 = analogRead(sensor2); // baca sensor2 kemudian masukkan nilainya dalam data2
    if(data1 > 540){ //sebagai pembatas batas atas data1
        data1 = 540;
    }
    if(data2 > 540){ //sebagai pembatas batas atas data2
        data2 = 540;
    }
}
void kondisi(){ // sub bagian bernama kondisi, berisi kondisi sensor yang dibandingkan
dengan motor
    baca_sensor(); //panggil sub bagian "kondisi"
    kecepatan_kanan = data1/6; //data1 dibagi 6 kemudian masukkan ke kecepatan kanan
    kecepatan_kiri = data2/6; //data2 dibagi 6 kemudian masukkan ke kecepatan kiri
    motor_kanan = 180 - kecepatan_kanan; //motor kanan diisi dengan hasil pengurangan nilai 180
dengan kecepatan kanan untuk mendapatkan kecepatan motor kanan
    motor_kiri = 0 + kecepatan_kiri; //motor kiri diisi dengan hasil penjumlahan nilai 0 dengan
kecepatan kiri untuk mendapatkan kecepatan motor kiri
    if(motor_kanan <= 91){ //batas untuk kecepatan motor kanan sehingga tidak menyebabkan motor
berputar terbalik
        motor_kanan = 91;
    }
    if(motor_kiri >= 90){ //batas untuk kecepatan motor kanan sehingga tidak menyebabkan motor
berputar terbalik
        motor_kiri = 90;
    }
    Serial.print("data 1 = ");
    Serial.print(motor_kanan);
    Serial.print(" data 2 = ");
    Serial.println(motor_kiri);
    delay(2);
}

```

```

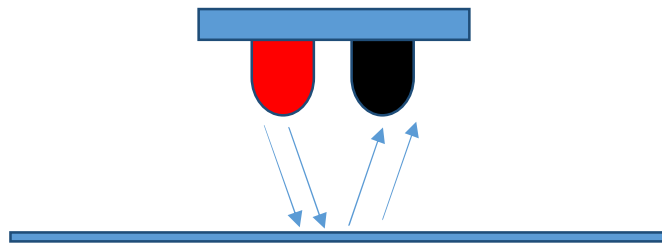
}
void loop() { //sub bagian bernama loop karena di sini program akan berulang terus menerus
  kondisi (); //panggil sub bagian bernama kondisi
  servo_kanan.write(motor_kanan); //motor kanan berputar sesuai kecepatan
  servo_kiri.write(motor_kiri); //motor kiri berputar sesuai kecepatan
}

```

Berikut penjelasan bagaimana code program berjalan:

Pertama baca sensor dengan ADC (Analog to Digital Converter), jika sudah didapatkan data maka dapat diolah untuk acuan kecepatan motor. Dalam pembahasan kali ini nilai hasil pembacaan nilai data antara 30-560. Sedangkan agar robot dapat bergerak maju nilai data yang diberikan ke motor kanan dan kiri berbeda. Jika motor kanan data yang diminta antara 91-180, sedangkan motor sebelah kiri antara 90-0. Dengan demikian rentang data entahlah 90, sehingga ini dijadikan acuan untuk pembandingan nilai dengan sensor.

Dalam hal ini nilai sensor akan semakin kecil nilainya jika terkena garis hitam, sedangkan jika terkena garis putih nilai data akan semakin besar, berikut gambarannya :



Gambar 4. Proses pantulan cahaya

Dari gambar di atas maka dapat dipahami mengapa terjadi perubahan nilai data saat sensor terkena garis hitam, hal itu disebabkan karena intensitas cahaya pantulan berkurang.

Kemudian kita bandingkan dengan nilai data motor, nilai yang harus didapatkan adalah 0-90, untuk itu kita bagi nilai data sensor dengan nilai data motor. Dalam program diatas batas maksimal program dibatasi yaitu 540 guna mendapatkan nilai data yang pas.

Untuk rumusnya pertama kita bagi nilai data sensor dengan nilai data motor.

Nilai pembandingan = nilai max sensor / nilai max motor

Nilai pembandingan = 540 / 90

Maka didapat nilai pembandingnya = 6 sehingga dapat disimpulkan setiap naik 6 data pada sensor maka naik 1 data pada kecepatan motor

Karena motor kanan dan motor kiri berbeda nilainya namun yang kita punya adalah data yang sama maka diperlukan cara agar dapat menjadikannya berbeda. Motor kanan akan semakin cepat jika nilainya 180 sedangkan motor kiri akan semakin cepat jika nilainya 0. Batas minimal motor kanan adalah 91 sedangkan batas minimal kecepatan motor kiri adalah 90, sedangkan nilai yang dihasilkan dari sensor adalah setiap kenaikan 6 data pada sensor maka naik satu data pada kecepatan sehingga dengan demikian,

Untuk motor kanan nilai kecepatan motor harus dikurangi jika sensor kanan semakin mengarah ke tengah garis begitu juga sebaliknya.

kecepatan motor kanan = $180 - (\text{nilai data sensor kanan} / 6)$

kecepatan motor kiri = $0 + (\text{nilai data sensor kiri} / 6)$

kemudian kita tinggal umpankan hasil perhitungan di atas ke masing-masing motor, dan motor akan bergerak maju mengikuti garis.

Dengan cara di atas robot sudah dapat berjalan mengikuti garis, namun robot akan langsung berjalan tanpa adanya sebuah pemicu. Oleh karena itu harus ada penambahan sebuah tombol sebagai pemicu agar robot dapat berjalan. Dalam gambar sudah terdapat sebuah tombol namun di dalam ode program memang belum ada penjelasan tentang ode program tombol tersebut.

Demikian penjelasan singkat tentang robot LineFollower dengan Servosebagia motor penggeraknya.