

# **Sistem Kendali PWM (Pulse Width Modulation) dengan Error Detection Memory**



Mata Kuliah : SISTEM KENDALI

Kode Dosen : AJR

Kelas : D3TK-43-02

Anggota Kelompok :

1. Panggah Danang p (6702190058)
2. Topan Budiargo (6702190013)

**PROGRAM STUDI D3 TEKNOLOGI KOMPUTER  
FAKULTAS ILMU TERAPAN  
UNIVERSITAS TELKOM  
BANDUNG  
2021**

## A. Tujuan

1. Mahasiswa dapat memahami fungsi dan cara kerja PWM pada motor DC
2. Mahasiswa dapat membuat program sistem kendali berbasis PWM dengan error detection memory pada robot *line follower* untuk menyimpan kondisi error terakhir (*last error condition*).

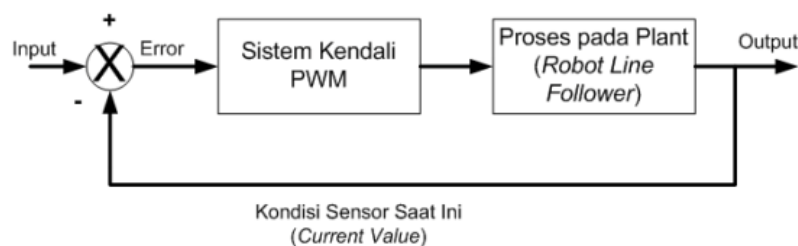
## 3. Alat dan Bahan

- 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)

## 4. Teori dasar

Pengertian PWM (Pulse Width Modulation) Sistem kendali berbasis Pulse Width Modulation (PWM) pada robot line follower secara umum hanya dapat digunakan untuk mengendalikan kecepatan motor kiri dan kanan. Akan tetapi, agar robot dapat selalu dapat mengikuti garis hitam dibutuhkan suatu algoritma atau mekanisme yang dapat menanggulangi keadaan error.

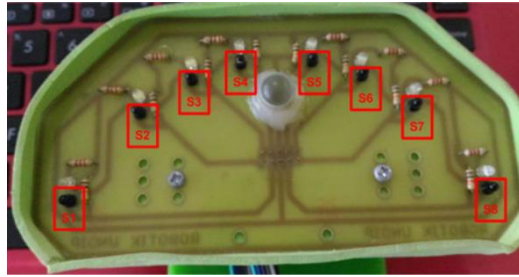
Kondisi tujuan (goal condition) yang menjadi target dari suatu sistem kendali disebut dengan setpoint. Program Studi D3 Teknologi Komputer Fakultas Ilmu Terapan 2019 Hal 1 2 Gambar 1 Control Loop dari Sistem Kendali PWM



Nilai setpoint didapatkan ketika kedua sensor di bagian tengah mendeteksi garis hitam. Kemampuan robot untuk memposisikan diri ke posisi setpoint dipengaruhi oleh kemampuan sistem mendeteksi kondisi error yang merupakan selisih antara setpoint dengan kondisi pembacaan sensor saat ini (Gambar 1)

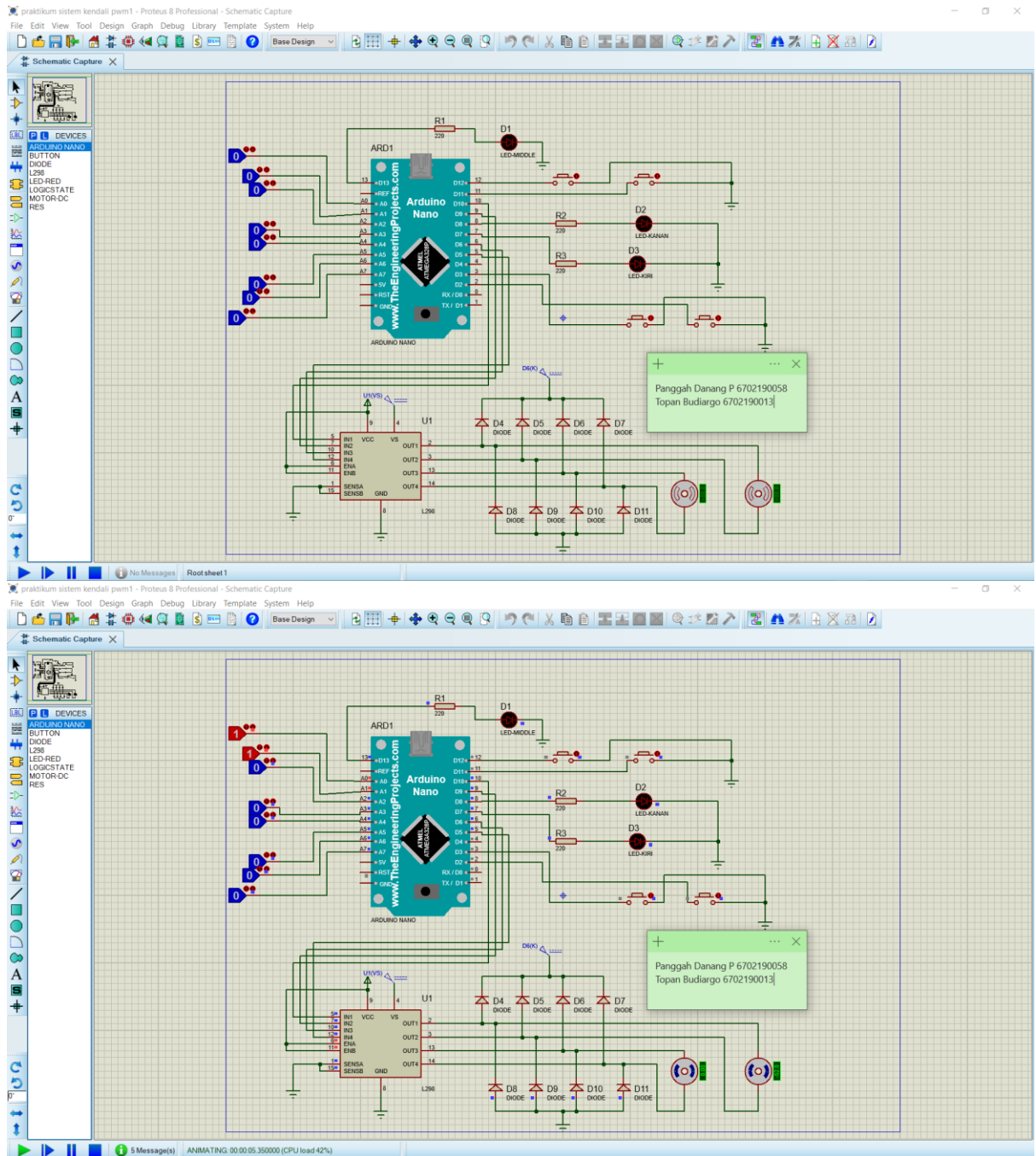
## 5. Hasil Percobaan

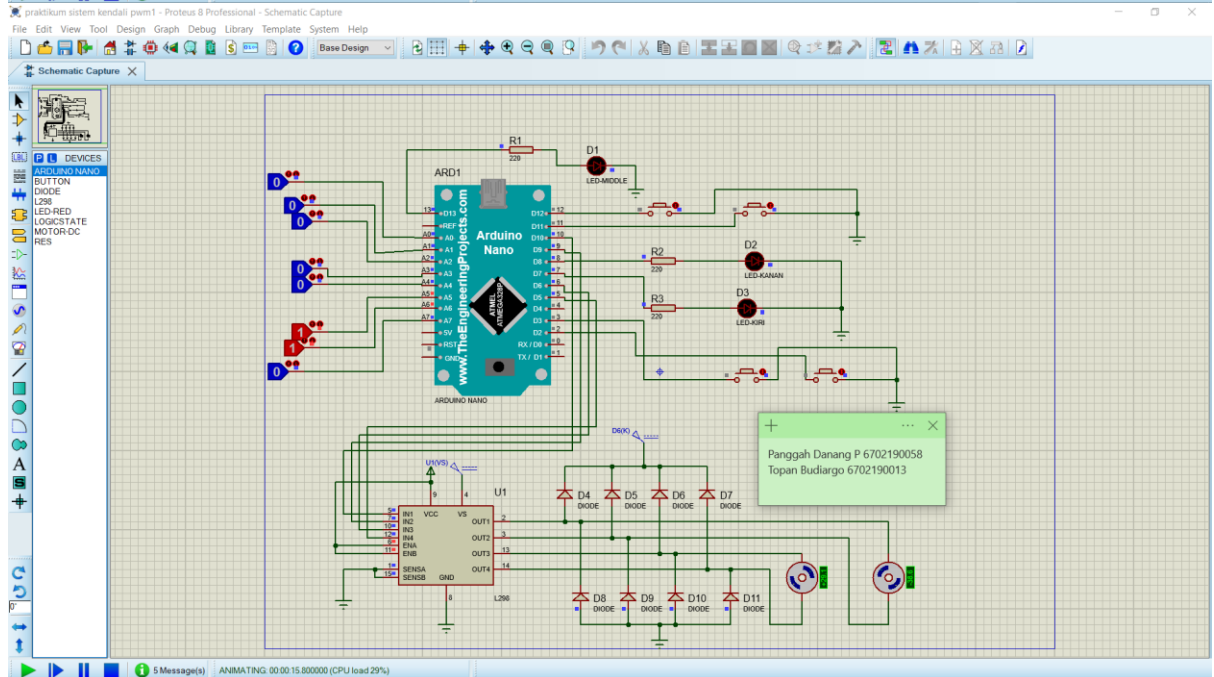
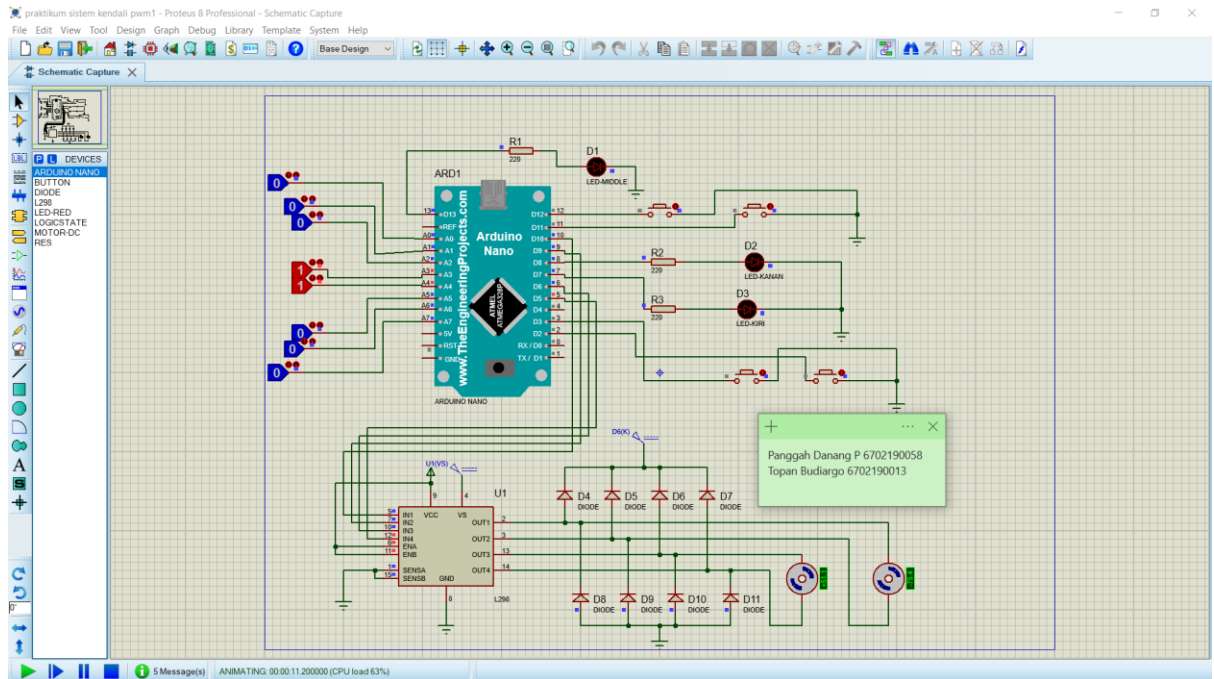
### 1. Kasus Percobaan



Buat sebuah aplikasi sistem kendali PWM pada robot dengan ketentuan sebagai berikut.

- a. Menggunakan seluruh sensor photodiode sebagai input sensor dengan contoh urutan sensor seperti pada Gambar 6. Kemudian, hasil pembacaan sensor akan mempengaruhi *duty cycle* pada motor kiri dan kanan dengan ketentuan sebagai berikut. Flowchart dari program dapat dilihat pada Gambar 7.
  - Sensor 4 dan 5 mendeteksi hitam, sisanya putih à *Duty cycle* 60% pada kedua motor (kedua motor maju)
  - Sensor 1 dan 2 mendeteksi hitam, sisanya putih à *Duty cycle* 50% Motor Kiri, 0% Motor Kanan
  - Sensor 2 dan 3 mendeteksi hitam, sisanya putih à *Duty cycle* 50% Motor Kiri, 20% Motor Kanan
  - Sensor 3 dan 4 mendeteksi hitam, sisanya putih à *Duty cycle* 50% Motor Kiri, 40% Motor Kanan
  - Sensor 7 dan 8 mendeteksi hitam, sisanya putih à *Duty cycle* 50% Motor Kanan, 0% Motor Kiri
  - Sensor 6 dan 7 mendeteksi hitam, sisanya putih à *Duty cycle* 50% Motor Kanan, 20% Motor Kiri
  - Sensor 5 dan 6 mendeteksi hitam, sisanya putih à *Duty cycle* 50% Motor Kanan, 40% Motor Kiri





'program :

```
int mKa1=10;

int mKa2=9;

int mKi1=6;

int mKi2=5;

int led2=8; //led kanan

int led3=7; // led kiri
```

```
int led1=13;  // led tengah

int sensor1=A0;

int sensor2=A1;

int sensor3=A2;

int sensor4=A3;

int sensor5=A4;

int sensor6=A5;

int sensor7=A6;

int sensor8=A7;
```

```
void setup() {

pinMode (led2,OUTPUT);

pinMode (led3,OUTPUT);

pinMode (led1,OUTPUT);

pinMode (mKa1,OUTPUT);

pinMode (mKa2,OUTPUT);

pinMode (mKi1,OUTPUT);

pinMode (mKi2,OUTPUT);

pinMode (sensor1,INPUT);

pinMode (sensor2,INPUT);

pinMode (sensor3,INPUT);

pinMode (sensor4,INPUT);

pinMode (sensor5,INPUT);

pinMode (sensor6,INPUT);

}
```

```

void loop() {

    if(digitalRead(sensor1)==1 && digitalRead(sensor2)==1){

        analogWrite(mKa1,123); // belok kiri

    }

    else if(digitalRead(sensor2)==1 && digitalRead(sensor3)==1){

        analogWrite(mKa1,123); // belok kiri

        analogWrite(mKi2,62); // kanan

    }

    else if(digitalRead(sensor3)==1 && digitalRead(sensor4)==1){

        analogWrite(mKa1,123); // belok kiri

        analogWrite(mKi2,100); // kanan

    }

    else if(digitalRead(sensor4)==1 && digitalRead(sensor5)==1){

        analogWrite(mKa1,150); //

        analogWrite(mKi2,150); //

    }

    else if(digitalRead(sensor5)==1 && digitalRead(sensor6)==1){

        analogWrite(mKa1,100); // belok kiri

        analogWrite(mKi2,123); // kanan

    }

    else if (digitalRead(sensor6)==1 && digitalRead(sensor7)==1 ){

        analogWrite(mKa1,62); // belok kiri

        analogWrite(mKi2,123); // kanan

    }

    else if (digitalRead(sensor7)==1 && digitalRead(sensor8)==1 ){

        analogWrite(mKi2,123); // kanan

    }

}

```

```
else{  
  
    digitalWrite(mKa1,LOW);  
  
    digitalWrite(mKa2,LOW);  
  
    digitalWrite(mKi1,LOW);  
  
    digitalWrite(mKi2,LOW);  
  
}  
  
}
```

## **6. Kesimpulan**

Setelah melakukan praktikum ini kita mengetahui fungsi dan cara kerja PWM pada motor DC dan pengaplikasiannya pada robot line follower.

## **7. Link Video Kegiatan praktikum**

<https://youtu.be/B7JW5OYOB>