

MODUL 2 SISTEM KENDALI PWM



Mata Kuliah : Sistem Kendali

Dosen : MHI

Kelas : D3TK-43-03

Kelompok 17 DuetMaut :

- | | |
|--------------------------|------------|
| 1. Ihsan Maulana Alzidni | 6702194020 |
| 2. Ajie Fauhad | 6702194011 |

PROGRAM STUDI D3 TEKNOLOGI KOMPUTER

FAKULTAS ILMU TERAPAN

UNIVERSITAS TELKOM

BANDUNG

2021

1. JUDUL PRAKTIKUM

Sistem Kendali PWM (*Pulse Width Modulation*)

2. MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

- a. Mahasiswa dapat memahami fungsi dan cara kerja PWM pada motor DC
- b. Mahasiswa dapat membuat program sistem kendali berbasis PWM

3. PERALATAN DAN BAHAN

1. Perangkat Lunak
 - Tinker Cad
 - Software IDE Arduino
2. Perangkat Keras
 - Arduino UNO R3
 - H-bridge Motor Driver
 - 6 Photodiode
 - 6 Resistor 33k Ohm

4. DASAR TEORI

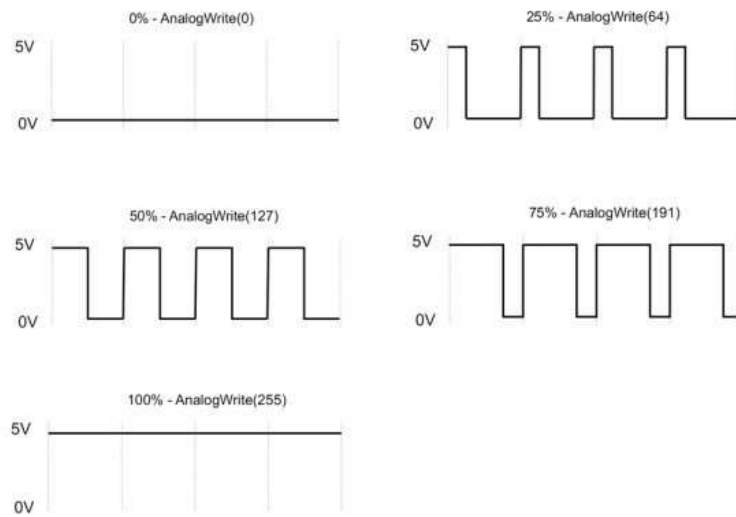
PWM merupakan mekanisme untuk membangkitkan sinyal keluaran yang periodenya berulang antara high dan low, dimana kita dapat mengontrol durasi sinyal high dan low sesuai dengan yang kita inginkan. Duty cycle merupakan prosentase periode sinyal high dan periode sinyal low, prosentase duty cycle akan berbanding lurus dengan tegangan rata-rata yang dihasilkan.

Pada Robot Line Follower sistem kendali on/off tidak dapat digunakan untuk mengendalikan kecepatan motor pada robot *line follower*. Oleh karena itu, dibutuhkan PWM untuk mengatur kecepatan motor. Dengan menggunakan PWM pengaturan kecepatan motor dapat diubah dengan memvariasikan nilai besarnya *duty cycle* pulsa. Pulsa yang nilai *duty cycle*-nya divariasikan inilah yang menentukan kecepatan motor. Besarnya amplitudo dan frekuensi pulsa adalah tetap, sedangkan besarnya *duty cycle* berubah-ubah sesuai dengan kecepatan yang diinginkan.

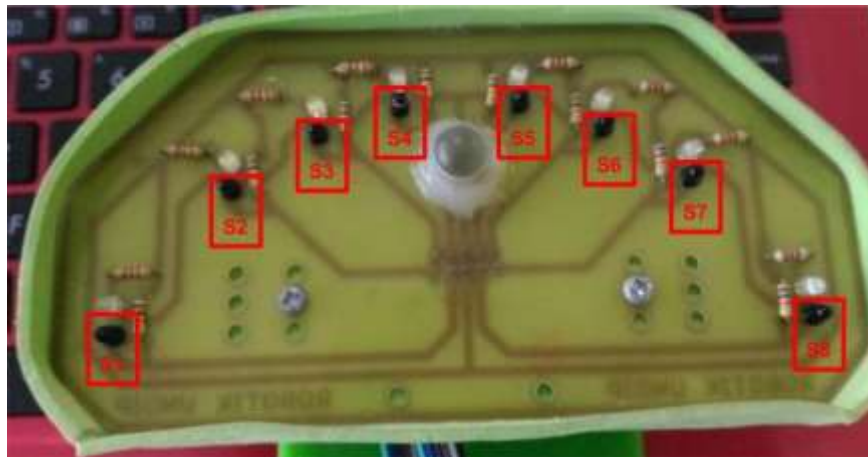
“Semakin besar duty cycle maka semakin cepat pula kecepatan motor, dan sebaliknya semakin kecil duty cycle maka semakin rendah pula kecepatan motor.”

Contoh, Pulsa kotak memiliki *duty cycle* dengan lebar 50%. Pada rangkaian elektronika digital, setiap perubahan PWM dipengaruhi oleh resolusi PWM itu sendiri. Resolusi adalah jumlah variasi perubahan nilai dalam PWM tersebut. Misalnya suatu PWM pada Arduino memiliki resolusi 8 bit, berarti PWM ini memiliki variasi perubahan nilai sebanyak 256 variasi mulai dari 0 – 255 perubahan nilai yang mewakili *duty cycle* 0% – 100% dari keluaran PWM tersebut. Sebagian kaki / pin Arduino telah mendukung fitur PWM. Pin Arduino Nano yang mendukung PWM ditandai dengan adanya tanda tilde (~) di depan angka pinnya, seperti 3, 5, 6, 9, 10, dan 11. Frekuensi yang digunakan dalam Arduino untuk PWM adalah 500Hz (500 siklus dalam 1 detik).

Untuk menggunakan PWM, kita bisa menggunakan fungsi *analogWrite()*. Nilai yang dapat dimasukkan pada fungsi tersebut yaitu antara 0 hingga 255. Nilai 0 berarti pulsa yang diberikan untuk setiap siklus selalu 0 volt, sedangkan nilai 255 berarti pulsa yang diberikan selalu bernilai 5 volt. Ilustrasi fungsi *analogWrite* dapat dilihat pada Gambar dibawah.

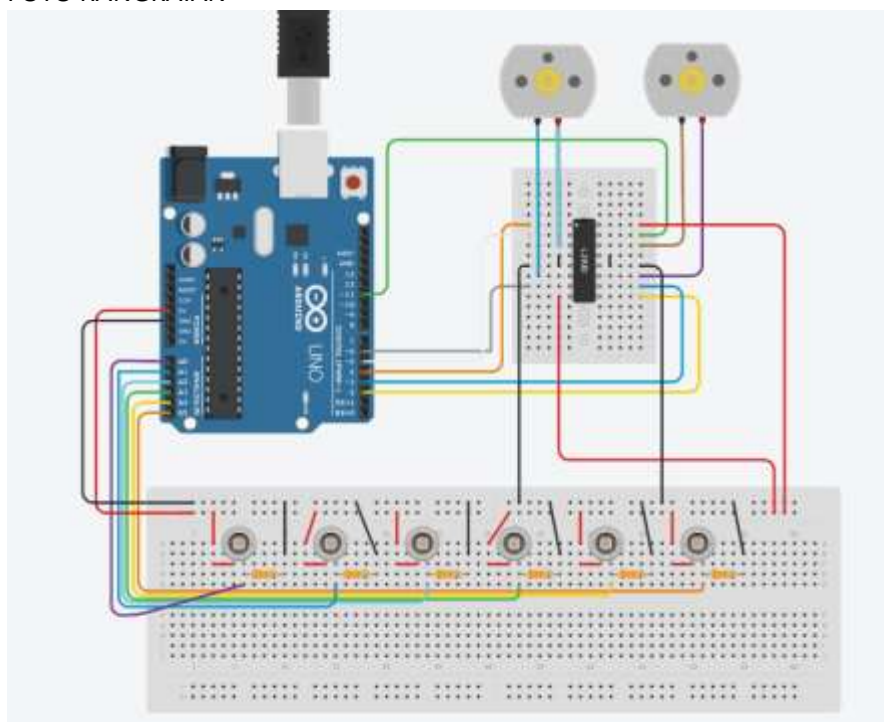


5. FOTO PERALATAN DAN BAHAN PRAKTIKUM



6. HASIL PRAKTIKUM

- FOTO RANGKAIAN



- KODE PROGRAM

```
// PRAKTIKUM 1 SISTEM KENDALI ON/OFF
// KELOMPOK 17 DUET MAUT
// ANGGOTA : AJIE FAUHAD FADHLULLAH (6702194011)
// ANGGOTA : IHSAN MAULANA (6702194020)

//Deklarasi Pin yang digunakan untuk sensor
int sensor1 = A0;
int sensor2 = A1;
int sensor3 = A2;
int sensor4 = A3;
int sensor5 = A4;
int sensor6 = A5;
int baca_sensor[6];

//Variable bantuan untuk menyimpan memory a.k.a Error Detection Memory (EDM)
int LastError;
//pin penggerak Motor
int pinEnable = 4; //Pin 1&2 harus HIGH
int pinEnable2 = 2; //Pin 3&4 harus HIGH

//pin Motorkiri
int motor_kiri1 = 5; //input motor driver
int motor_kiri2 = 6; //input motor driver

//pin Motorkanan
int motor_kanan1 = 3;
int motor_kanan2 = 11;

void setup(){
  // Keenam Sensor Photodiode sebagai INPUT yaitu sensor cahaya
  pinMode(sensor1, INPUT);
  pinMode(sensor2, INPUT);
  pinMode(sensor3, INPUT);
  pinMode(sensor4, INPUT);
  pinMode(sensor5, INPUT);
  pinMode(sensor6, INPUT);

  //Motor sebagai OUTPUT penggerak
  pinMode(pinEnable, OUTPUT);
  pinMode(pinEnable2, OUTPUT);
  pinMode(motor_kiri1, OUTPUT);
  pinMode(motor_kiri2, OUTPUT);
  pinMode(motor_kanan1, OUTPUT);
  pinMode(motor_kanan2, OUTPUT);

  Serial.begin(9600);
}

//Membaca sinyal analog dari sensor
void readsensor(){
  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(100);
  // Menampilkan data sensor ke Serial Monitor
  // Data sensor 1-6
  // Formatting tampilan sensor

  for(int i=0; i<=5; i++){
    Serial.println("Sensor");
    Serial.println(i);
    Serial.println(baca_sensor[i]);
    Serial.println(" ");
    delay(1000);
  }
}

void Sensor12(){
  digitalWrite(pinEnable, HIGH);
  //analogWrite(pin,value 0-255)
  analogWrite(motor_kiri1,0);
  analogWrite(motor_kiri2,0);
}
```

```

    digitalWrite(pinEnable2, HIGH);
    analogWrite(motor_kanan1,127);
    analogWrite(motor_kanan2,0);
}

void Sensor23(){
    digitalWrite(pinEnable, HIGH);
    //analogWrite(pin,value 0-255)
    analogWrite(motor_kiri1,0.2*255);
    analogWrite(motor_kiri2,0);

    digitalWrite(pinEnable2, HIGH);
    analogWrite(motor_kanan1,0.5*255);
    analogWrite(motor_kanan2,0);
}

void Sensor34(){
    digitalWrite(pinEnable, HIGH);
    //analogWrite(pin,value 0-255)
    analogWrite(motor_kiri1,0.6*255);
    analogWrite(motor_kiri2,0);

    digitalWrite(pinEnable2, HIGH);
    analogWrite(motor_kanan1,0.6*255);
    analogWrite(motor_kanan2,0);
}

void Sensor45(){
    digitalWrite(pinEnable, HIGH);
    //analogWrite(pin,value 0-255)
    analogWrite(motor_kiri1,0.5*255);
    analogWrite(motor_kiri2,0);

    digitalWrite(pinEnable2, HIGH);
    analogWrite(motor_kanan1,0.2*255);
    analogWrite(motor_kanan2,0);
}

void Sensor56(){
    digitalWrite(pinEnable, HIGH);
    //analogWrite(pin,value 0-255)
    analogWrite(motor_kiri1,0.5*255);
    analogWrite(motor_kiri2,0);

    digitalWrite(pinEnable2, HIGH);
    analogWrite(motor_kanan1,0);
    analogWrite(motor_kanan2,0);
}

void loop(){
    readsensor();
    //Serial.println("Nilai Last Error : ");

    //Sensor 1 dan 2 mendeteksi gelap, sisanya terang □ Duty cycle 0% motor kiri, 50% motor kanan
    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)
    {
        Sensor12();
    }

    //Sensor 2 dan 3 mendeteksi gelap, sisanya terang □ Duty cycle 20% motor kiri, 50% motor kanan
    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)
    {
        Sensor23();
    }

    //Sensor 3 dan 4 mendeteksi gelap, sisanya terang □ Duty cycle 60% pada kedua motor (kedua motor aktif)
    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)
    {
        Sensor34();
    }
}

```

```

//Sensor 4 dan 5 mendeteksi gelap, sisanya terang □ Duty cycle 50% motor kiri, 20% motor kanan
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)
{
    Sensor45();
}

//Sensor 5 dan 6 mendeteksi gelap, sisanya terang □ Duty cycle 50% Motor kiri, 0% motor kanan
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)
{
    Sensor56();
}

//Semua sensor mendeteksi terang □ Duty cycle kedua motor 0% (semua motor mati)
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)
{
    digitalWrite(pinEnable, HIGH);
    //analogWrite(pin,value 0-255)
    analogWrite(motor_kiri1,0);
    analogWrite(motor_kiri2,0);

    digitalWrite(pinEnable2, HIGH);
    analogWrite(motor_kanan1,0);
    analogWrite(motor_kanan2,0);
}

//Studi Kasus Percobaan 2 : Error Condition --> Hanya 1 buah sensor terbaca
//Simpan kondisi terakhir pada suatu variabel

//Kondisi Sensor 1 saja yang membaca gelap
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
    //Eksekusi kasus Sensor 1 dan 2
    Sensor12();
    Serial.println("Error = -2");
    LastError = -2;
}

//Kondisi Sensor 2 saja yang membaca gelap
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
    //Eksekusi kasus Sensor 2 dan 3
    Sensor23();
    Serial.println("Error = -1");
    LastError = -1;
}

//Kondisi Sensor 3 saja yang membaca gelap
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 = 0
    //Eksekusi kasus Sensor 3 dan 4
    Sensor34();
    Serial.println("Error = 0");
    LastError = 0;
}

//Kondisi Sensor 4 saja yang membaca gelap
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
    //Eksekusi kasus Sensor 4 dan 5
    Sensor45();
}

```

```

Serial.println("Error = -1");
LastError = 1;
}

//Kondisi Sensor 5 saja yang membaca gelap
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
    //Eksekusi kasus Sensor 5 dan 6
    Sensor56();
    Serial.println("Error = 2");
    LastError = 2;
}

}

```

- TABEL

Nilai Sensor 1	Nilai Sensor 2	Nilai Sensor 3	Nilai Sensor 4	Nilai Sensor 5	Nilai Sensor 6	RPM Motor Kiri	RPM Motor Kanan
Sensor 1 dan 2 mendeteksi gelap, sisanya terang → <i>Duty cycle</i> 0% motor kiri, 50% motor kanan							
33	33	686	686	686	686	0	4873
Sensor 2 dan 3 mendeteksi gelap, sisanya terang → <i>Duty cycle</i> 20% motor kiri, 50% motor kanan							
686	33	33	686	686	686	1956	4873
Sensor 3 dan 4 mendeteksi gelap, sisanya terang → <i>Duty cycle</i> 60% pada kedua motor (kedua motor aktif)							
686	686	33	33	686	686	5873	5873
Sensor 4 dan 5 mendeteksi gelap, sisanya terang → <i>Duty cycle</i> 50% motor kiri, 20% motor kanan							
686	686	686	33	33	686	4873	1956
Sensor 5 dan 6 mendeteksi gelap, sisanya terang → <i>Duty cycle</i> 50% Motor kiri, 0% motor kanan							
686	686	686	686	33	33	4873	0
Semua sensor mendeteksi terang → <i>Duty cycle</i> kedua motor 0% (semua motor mati)							
686	686	686	686	686	686	0	0

7. KESIMPULAN PRAKTIKUM

Sistem Kendali PWM ini merupakan system kendali yang sinyal periodenya berulang high dan low dan cocok digunakan pada Robot Line Follower , karena jika hanya menggunakan system kendali ON/OFF tidak bisa mengendalikan kecepatan motor , dengan PWM ini dapat mengatur kecepatan motor dapat diubah dengan memvariasikan nilai besarnya *duty cycle* pulsa. Pulsa yang nilai *duty cycle*-nya divariasikan inilah yang menentukan kecepatan motor.

