## Program Studi D3 Teknik Komputer Fakultas Ilmu Terapan

***2020***

Hal 1

# MODUL 2 Sistem Kendali PWM

# logo telkom

# Mata Kuliah : Sistem Kendali

# Kode Dosen : MHI

# Kelas : D3TK-43-03

# Tim 21: BOBO

# Nama : Dylan Avilla Saeful Azzam 6702190089

# PROGRAM STUDI D3 TEKNOLOGI KOMPUTER FAKULTAS ILMU TERAPAN

# UNIVERSITAS TELKOM

# BANDUNG

# 2021

1. **JUDUL PRAKTIKUM**

Sistem Kendali PWM (*Pulse Width Modulation*)

# MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

1. Mahasiswa dapat memahami fungsi dan cara kerja PWM pada motor DC
2. Mahasiswa dapat membuat program sistem kendali berbasis PWM

# PARAMETER PENILAIAN

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

* 1. **PERALATAN DAN BAHAN**

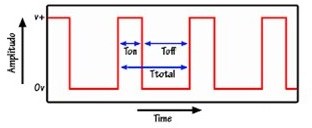
Perangkat Lunak :

* + 1. Software IDE Arduino
    2. Software TinkerCADProteus (untuk simulasi)

# TEORI DASAR

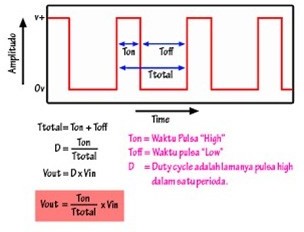
* 1. **Pengertian PWM (*Pulse Width Modulation*)**

*Pulse Width Modulation* (PWM) adalah sebuah metode memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam satu periode, untuk mendapatkan tegangan rata-rata yang berbeda. Bebarapa contoh aplikasi PWM adalah pengontrolan daya atau tegangan yang masuk ke beban, regulator tegangan, pengendalian kecepatan motor, dan lain-lain.



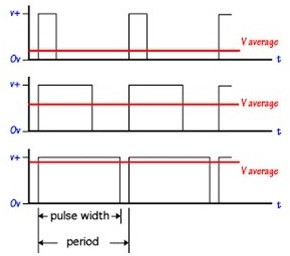
*Gambar 1 Lebar pulsa pada PWM.*

Sinyal PWM pada umumnya memiliki amplitudo dan frekuensi dasar yang tetap tetapi memiliki lebar pulsa yang bervariasi. Lebar pulsa PWM berbanding lurus dengan amplitude sinyal asli yang belum termodulasi. Oleh karena itu, sinyal PWM memiliki frekuensi gelombang yang tetap namun *duty cycle* bervariasi antara 0% hingga 100%.



*Gambar 2 Ilustrasi perhitungan duty-cycle pada PWM.*

Dari persamaan di atas, diketahui bahwa perubahan *duty cycle* akan merubah tegangan output atau tegangan rata-rata seperti gambar dibawah ini.



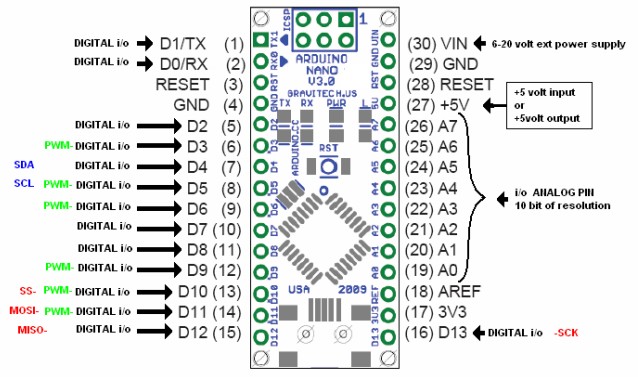
*Gambar 3 Hubungan perubahan duty-cycle terhadap tegangan output.*

# Sistem Kendali PWM pada Robot Line Follower menggunakan Arduino

Seperti yang telah dibahas pada praktikum modul 1, 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 PWMm pengaturan kecepatan motor dapat diubah dengan memvariasikan nilai besarnya *duty cycle* pulsa. Pulsa yang 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

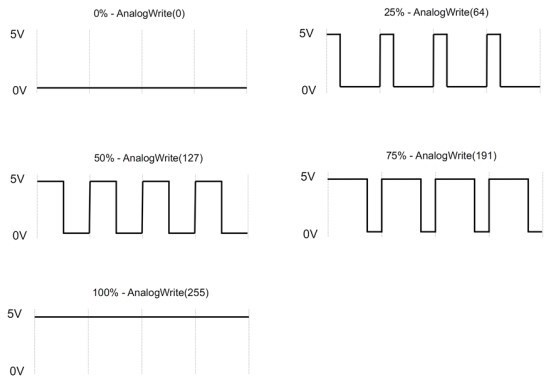
kecepatan yang diinginkan. Semakin besar *duty cycle* maka semakin cepat pula kecepatan motor, dan sebaliknya semakin kecil *duty cycle* maka semakin rendah pula kecepatan motor. Sebagai contoh bentuk pulsa yang dikirimkan adalah seperti pada Gambar 6. Pulsa kotak ini memiliki *duty cycle* dengan lebar 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).



*Gambar 4 Pinout PWM pada Arduino Nano.*

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 5.



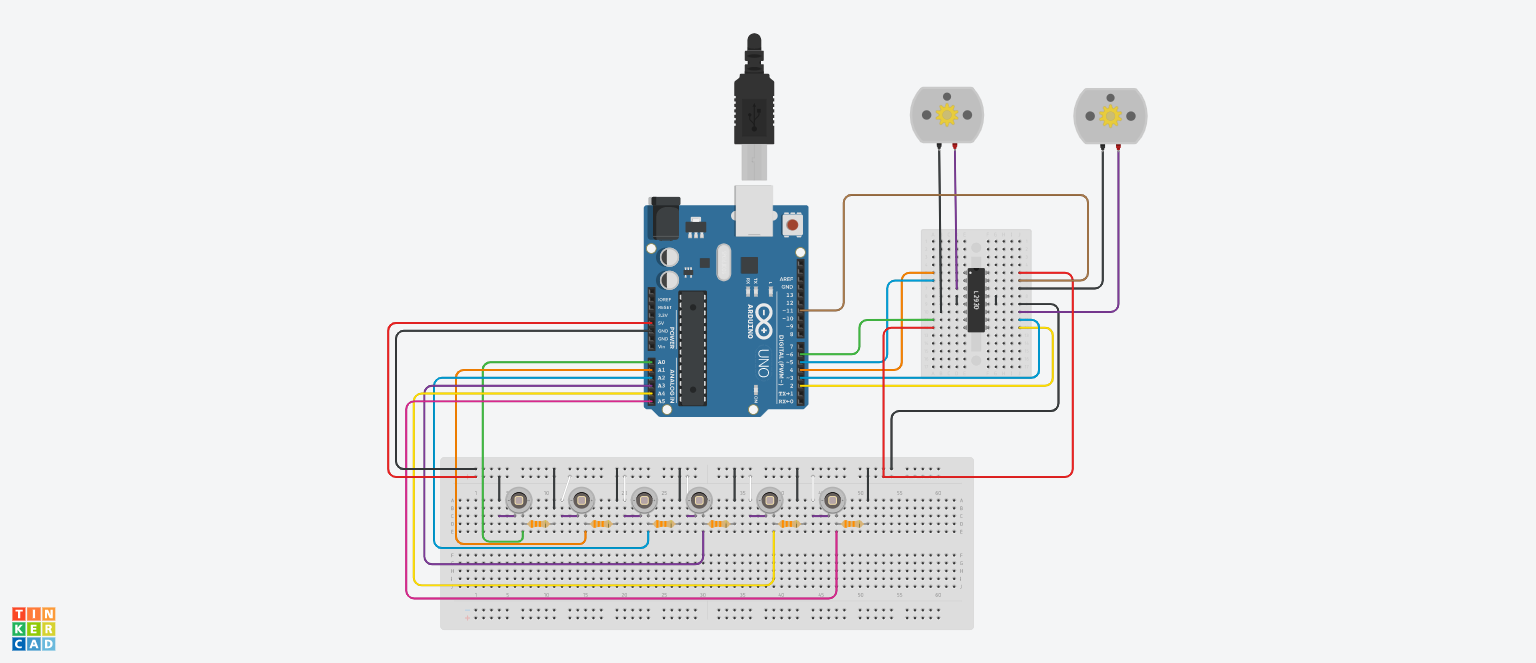
*Gambar 5 Siklus Pulsa PWM.*

Pada Gambar 5, semakin besar *duty cycle* pulsa kotak, maka semakin lama pula posisi logika HIGH. Jika misalnya motor diatur agar berjalan dengan *duty cycle* 50% (analogWrite 127), ketika diberi logika HIGH maka motor akan berada pada kondisi “nyala-mati-nyala-mati” sesuai dengan bentuk pulsa tersebut. Semakin lama motor berada pada kondisi “menyala” maka semakin cepat pula kecepatan motor tersebut. Motor akan berputar dengan kecepatan maksimum apabila mendapat pulsa dengan *duty cycle* 100% (analogWrite 255). Dengan kata lain motor mendapat logika high terus menerus. Pada praktikum ini PWM akan digunakan pada beberapa kondisi. Ketika sensor di bagian tengah mendeteksi garis hitam, maka robot bergerak maju dengan *duty cycle* 60%.

# PROSEDUR PRAKTIKUM

1. Percobaan dalam praktikum

# Kasus Percobaan 1 (Total Nilai 70 poin).



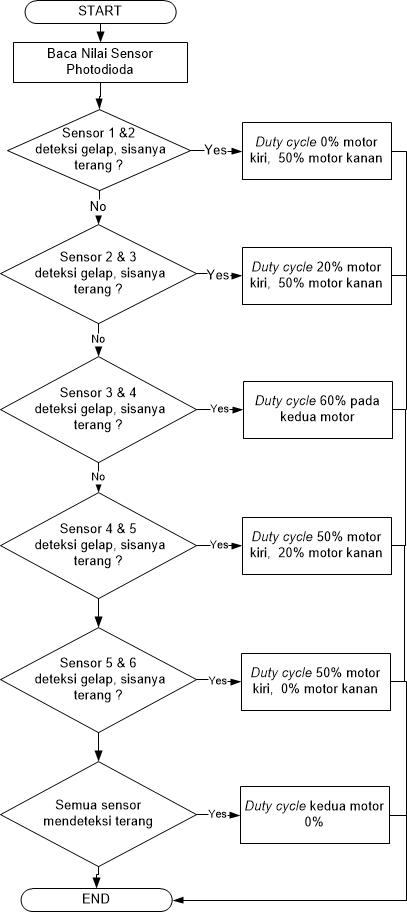
*Gambar 6 Contoh susunan dan urutan sensor pada robot line follower dan TinkerCAD.*

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

1. Buat rangkaian 6 buah sensor photodiode dengan 2 sensor dengan contoh urutan sensor seperti pada [Gambar 6](#_bookmark0) (**nilai 10 poin**). 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](#_bookmark1). Program dapat dimodifikasi dari program sistem kendali on-off dari modul sebelumnya.

* + Sensor 1 dan 2 mendeteksi gelap, sisanya terang  *Duty cycle* 0% motor kiri, 50% motor kanan (**nilai 10 poin**).
  + Sensor 2 dan 3 mendeteksi gelap, sisanya terang  *Duty cycle* 20% motor kiri, 50% motor kanan (**nilai 10 poin**).
  + Sensor 3 dan 4 mendeteksi gelap, sisanya terang  *Duty cycle* 60% pada kedua motor (kedua motor aktif) (**nilai 10 poin**).
  + Sensor 4 dan 5 mendeteksi gelap, sisanya terang  *Duty cycle* 50% motor kiri, 20% motor kanan (**nilai 10 poin**).
  + Sensor 5 dan 6 mendeteksi gelap, sisanya terang  *Duty cycle* 50% Motor kiri, 0% motor kanan (**nilai 10 poin**).
* Semua sensor mendeteksi terang  Duty cycle kedua motor 0% (semua motor mati) (**nilai 10 poin**).

Jelaskan fungsi dari pengubahan berbagai duty cycle pada kondisi-kondisi di atas dan apa perbedaan yang terjadi pada masing-masing perubahan duty cycle!

*Gambar 7 Flowchart sistem kendali PWM pada robot line follower.*

# Kasus Percobaan 2 (Total Nilai 30 poin)

Buatlah sub program yang dapat menyimpan kondisi terakhir dari pembacaan sensor dalam sebuah variabel dan jika hanya terdapat 1 (satu) buah nilai sensor, program harus dapat mengeksekusi kondisi terakhir yang telah disimpan pada variabel. Flowchart dapat dilihat pada [Gambar 8](#_bookmark2). Jika seluruh sensor mendeteksi nilai putih maka seluruh motor harus berhenti.

START

Jalankan program sistem kendali PWM pada robot *line follower*

No

Hanya ada 1 sensor yang terbaca?

Simpan kondisi pembacaan sensor terakhir ke variabel/ EEPROM

Seluruh sensor mendeteksi warna putih

Yes

END

Kedua motor berhenti/

*duty cycle* 0%

Eksekusi kondisi pembacaan sensor terakhir

*Gambar 8 Flowchart program penyimpanan kondisi terakhir.*

# Jurnal Praktikum

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

* Judul Praktikum :

Sistem Kendali PWM (*Pulse Width Modulation*)

* Maksud dan Tujuan Praktikum :
  + 1. Mahasiswa dapat memahami fungsi dan cara kerja PWM pada motor DC
    2. Mahasiswa dapat membuat program sistem kendali berbasis PWM
* Peralatan dan Komponen Praktikum :

Perangkat Lunak :

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

Komponen :

1. 1 buah Arduino Uno R3
2. 2 buah DC Motor
3. 6 buah Resistor
4. 6 buah Photodiode
5. 1 buah Breadboard
6. 1 buah Breadboard Small
7. 1 buah H-bridge Motor Driver 🡪 L293D
   * 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 PWMm pengaturan kecepatan motor dapat diubah dengan memvariasikan nilai besarnya *duty cycle* pulsa. Pulsa yang 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.

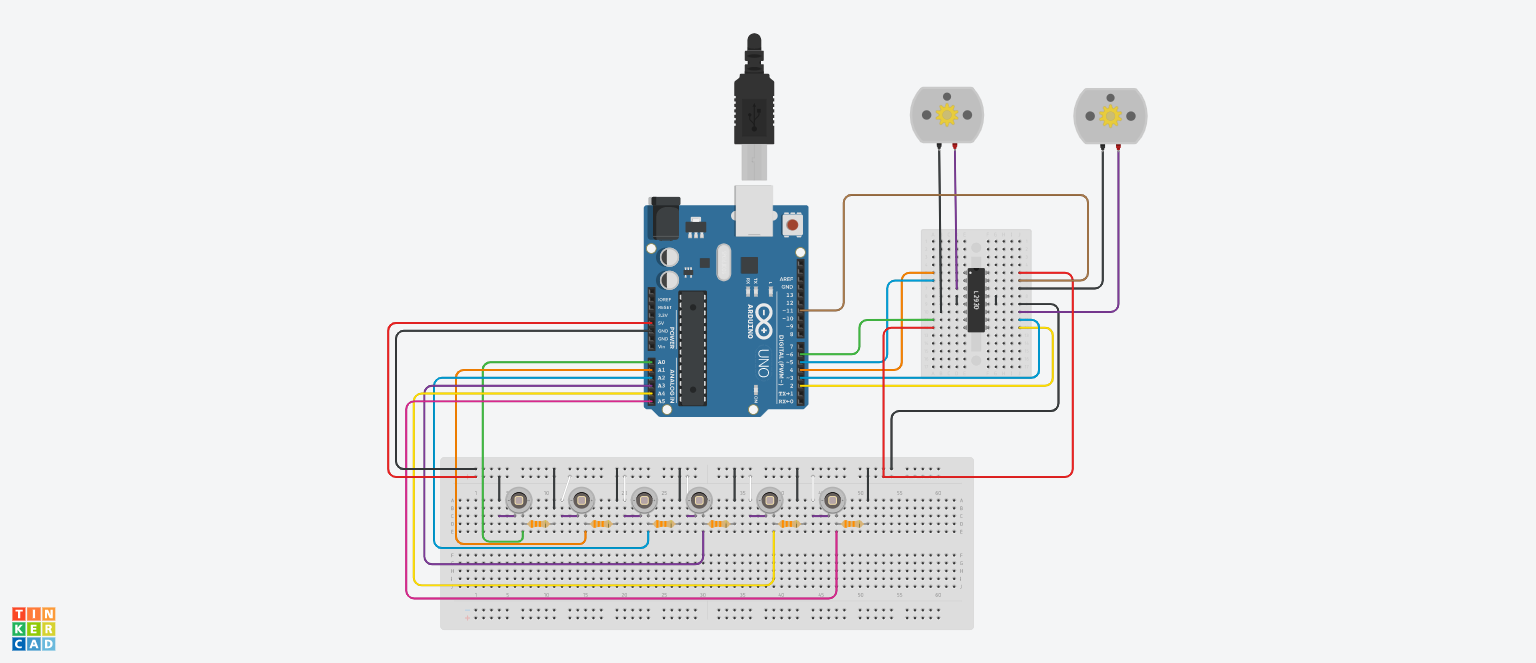
*“Semakin besar duty cycle maka semakin cepat pula kecepatan motor, dan*

*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).

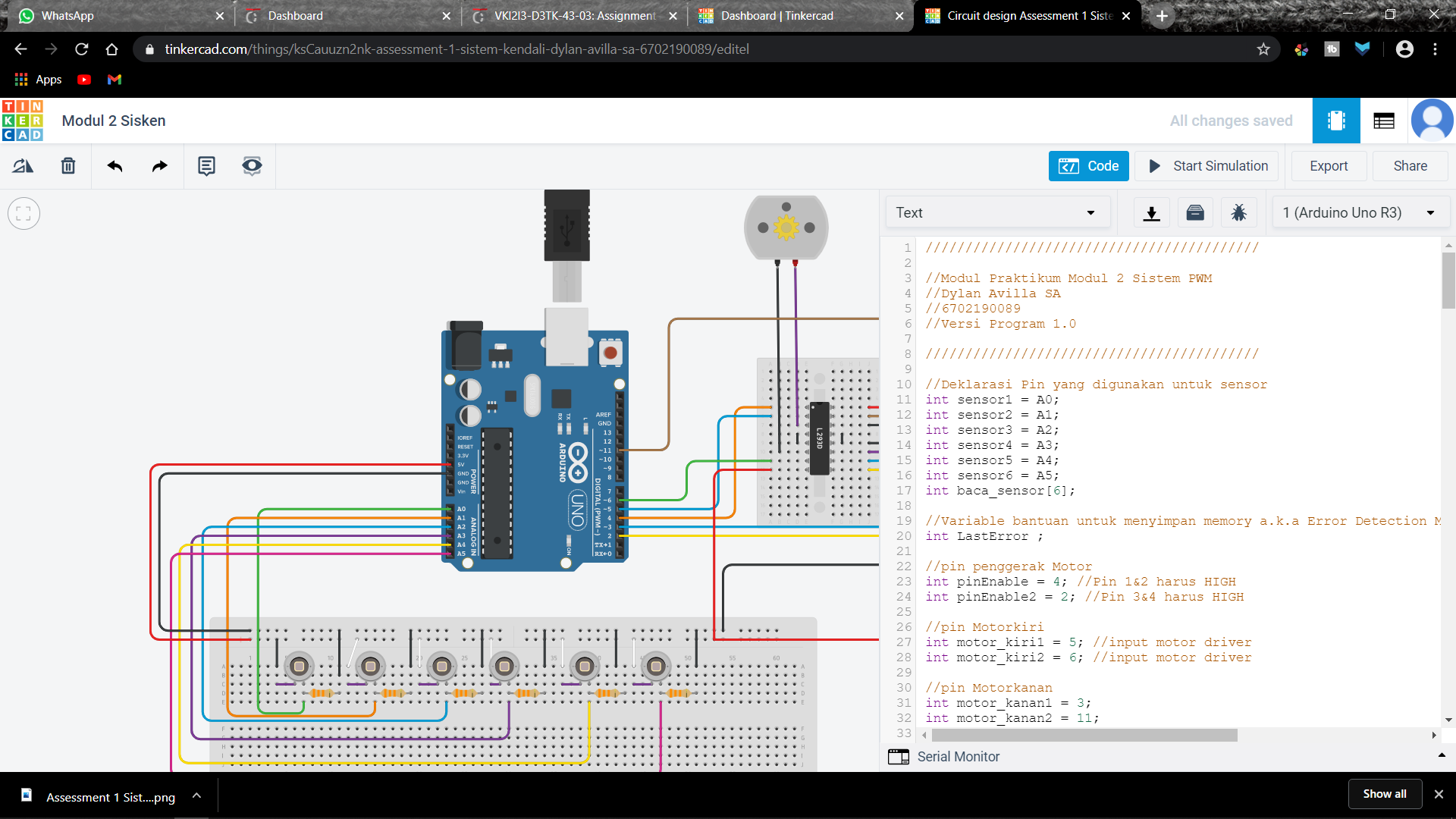
Untuk menggunakan PWM, kita bisa menggunakan fungsi *analogWrite()*. Nilai yang dapat dimasukkan pada fungsi tersebut yaitu antara 0 hingga 255. Nilai 0 berarti pulsa yang diberikan untuk setiap siklus selalu 0 volt, sedangkan nilai 255 berarti pulsa yang diberikan selalu bernilai 5 volt. Ilustrasi fungsi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

* + Foto Peralatan dan Bahan Praktikum :
  + HASIL PRATIKUM



**PERCOBAAN PERTAMA**

* Contoh output program pertama
* KODEPROGRAM



//MODUL PRAKTIKUM 2 SISTEM PWM //

// NAMA ANGGOTA : Dylan Avilla SA (6702190089)

//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 = 0;

//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 OOUTPUT pengerak

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 tampilkan sensor

for(int i=0; i<=5; i++){

Serial.println(baca\_sensor[i]);

}

}

void loop(){

readsensor();

//kondisi 1 jika sensor 1 dan 2 mendeteksi gelap maka motor kanan menyala

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\_kanan1,127);

analogWrite(motor\_kanan2,0);

}

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)

{

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);

}

//kondisi 2 jika sensor 3 dan 4 mendeteksi gelap maka motor kiri dan kanan menyala

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\_kiri1,0.6\*255);

analogWrite(motor\_kiri2,0);

digitalWrite(pinEnable2, HIGH);

analogWrite(motor\_kanan1,0.6\*255);

analogWrite(motor\_kanan2,0);

}

//kondisi 3 jika sensor 5 dan 6 mendeteksi gelap maka motor kiri menyala

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\_kiri1,0.5\*255);

analogWrite(motor\_kiri2,0);

digitalWrite(pinEnable2, HIGH);

analogWrite(motor\_kanan1,0.2\*255);

analogWrite(motor\_kanan2,0);

}

//kondisi 4 jika ke-6 sensor diterangkan semua maka motor tidak akan menyala

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\_kiri1,0.5\*255);

analogWrite(motor\_kiri2,0);

digitalWrite(pinEnable2, HIGH);

analogWrite(motor\_kanan1,0);

analogWrite(motor\_kanan2,0);

}

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)

{

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);

}

}

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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  *Duty cycle* 0% motor kiri, 50% motor kanan | | | | | | | |
| 33 | 33 | 686 | 686 | 686 | 686 | 0 | 4873 |
| Sensor 2 dan 3 mendeteksi gelap, sisanya terang  *Duty cycle* 20% motor kiri, 50% motor kanan | | | | | | | |
| 686 | 33 | 33 | 686 | 686 | 686 | 1956 | 4873 |
| Sensor 3 dan 4 mendeteksi gelap, sisanya terang  *Duty cycle* 60% pada kedua motor (kedua motor aktif) | | | | | | | |
| 686 | 686 | 33 | 33 | 686 | 686 | 5873 | 5873 |
| Sensor 4 dan 5 mendeteksi gelap, sisanya terang  *Duty cycle* 50% motor kiri, 20% motor kanan | | | | | | | |
| 686 | 686 | 686 | 33 | 33 | 686 | 4873 | 1956 |
| Sensor 5 dan 6 mendeteksi gelap, sisanya terang  *Duty cycle* 50% Motor kiri, 0% motor kanan | | | | | | | |
| 686 | 686 | 686 | 686 | 33 | 33 | 4873 | 0 |
| Semua sensor mendeteksi terang  Duty cycle kedua motor 0% (semua motor mati) | | | | | | | |
| 686 | 686 | 686 | 686 | 686 | 686 | 0 | 0 |

* + Hasil Praktikum (Foto rangkaian, kode program, dan isilah tabel berikut ini)  upload semua file pada Github kelompok masing-masing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nilai Sensor 1 | Nilai Sensor 2 | Nilai Sensor 3 | Nilai Sensor 4 | Nilai Sensor 5 | Nilai Sensor 6 | RPM  Motor Kiri | RPM  Motor Kanan |
| 33 | 33 | 687 | 687 | 687 | 687 | 0 | 4851 |
| 687 | 33 | 33 | 687 | 687 | 687 | 1862 | 4851 |
| 687 | 687 | 33 | 33 | 687 | 687 | 5802 | 5802 |
| 687 | 687 | 687 | 33 | 33 | 687 | 4851 | 1862 |
| 687 | 687 | 687 | 687 | 33 | 33 | 4851 | 0 |
|  |  |  |  |  |  |  |  |