

## MODUL 2 Sistem Kendali PWM

### 1. JUDUL PRAKTIKUM

Sistem Kendali PWM (*Pulse Width Modulation*)

### 2. 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

### 3. PARAMETER PENILAIAN

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

### 4. PERALATAN DAN BAHAN

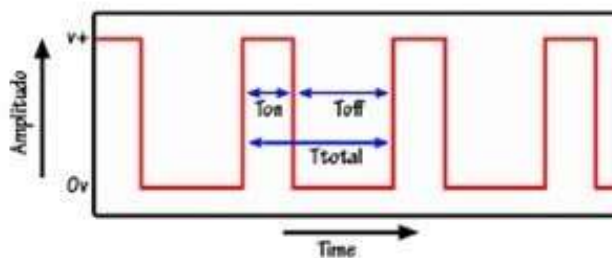
Perangkat Lunak :

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

### 5. TEORI DASAR

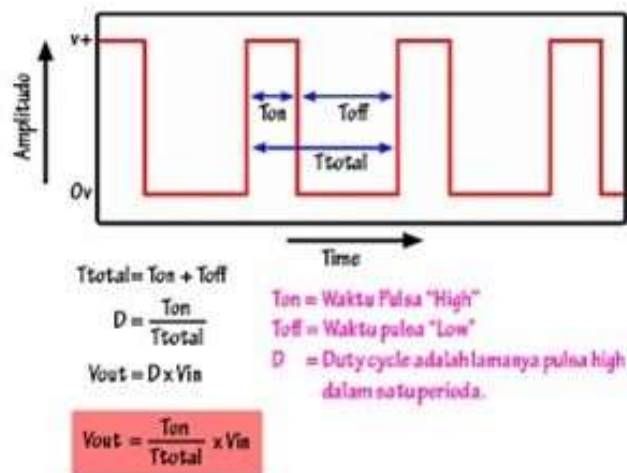
#### 5.1. Pengertian PWM (*Pulse Width Modulation*)

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



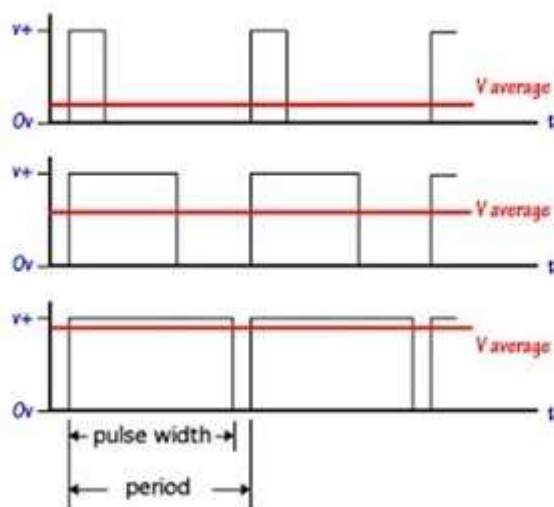
Gambar 1 Lebar pulsa pada PWM.

Sinyal PWM pada umumnya memiliki amplitudo dan frekuensi dasar yang tetap tetapi memiliki lebar pulsa yang bervariasi. Lebar pulsa PWM berbanding lurus dengan amplitudo 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.



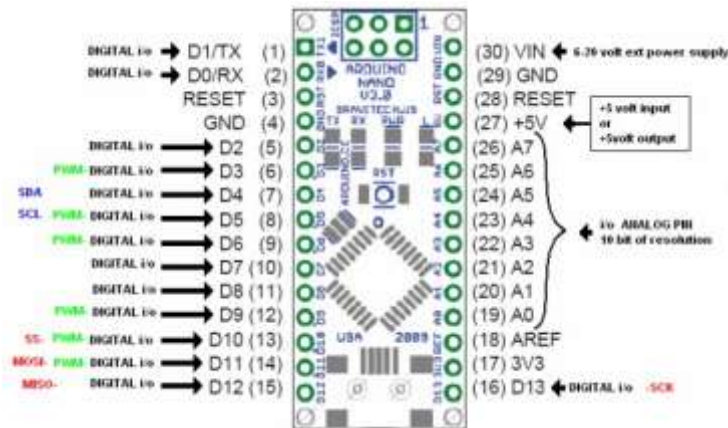
Gambar 3 Hubungan perubahan duty-cycle terhadap tegangan output.

## 5.2. 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 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

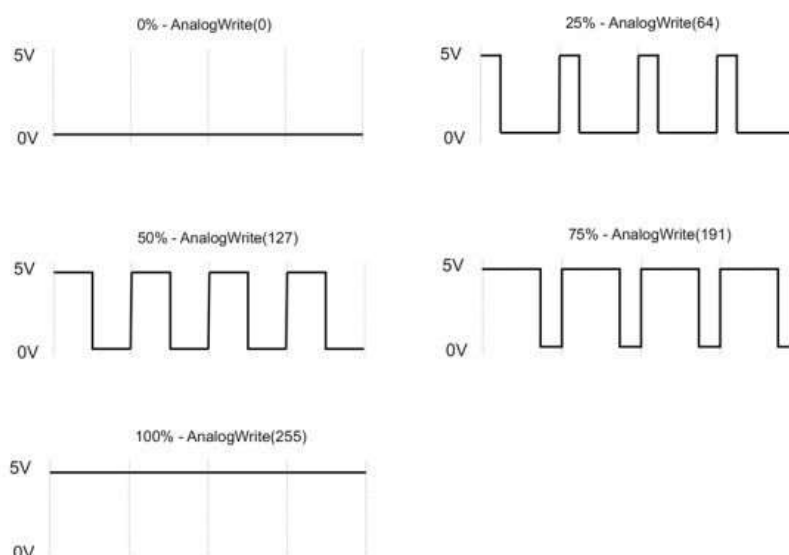
motor, dan sebaliknya semakin kecil *duty cycle* maka semakin rendah pula kecepatan motor. Sebagai contoh bentuk pulsa yang dikirimkan adalah seperti pada Gambar 6. Pulsa kotak ini memiliki *duty cycle* dengan lebar 50%.

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.

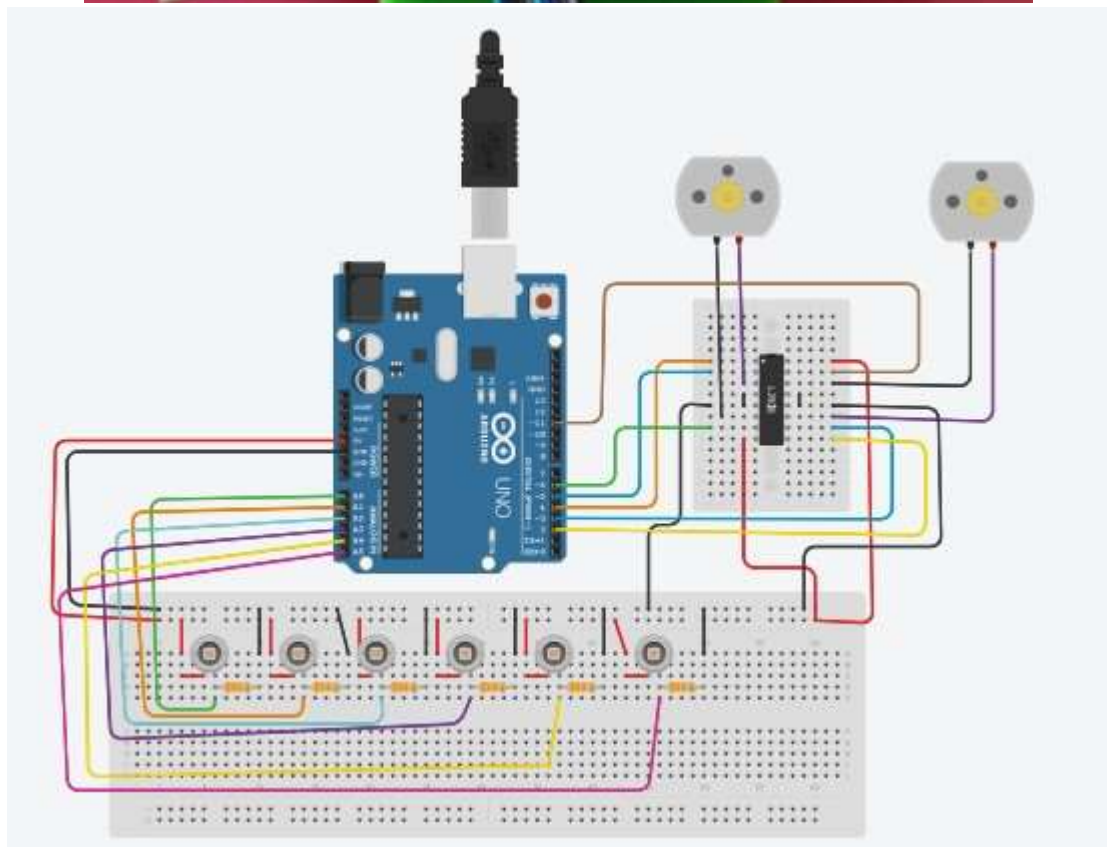
Pada Gambar 5, semakin besar *duty cycle* pulsa kotak, maka semakin lama pula posisi logika HIGH. Jika misalnya motor diatur agar berjalan dengan *duty cycle* 50% (analogWrite 127), ketika diberi logika HIGH maka motor akan berada pada kondisi “nyala-mati-nyala-mati” sesuai dengan bentuk pulsa tersebut. 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%.

## 6. PROSEDUR PRAKTIKUM

### A. Percobaan dalam praktikum

#### 1. Kasus Percobaan 1 (Total Nilai 70 poin).



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

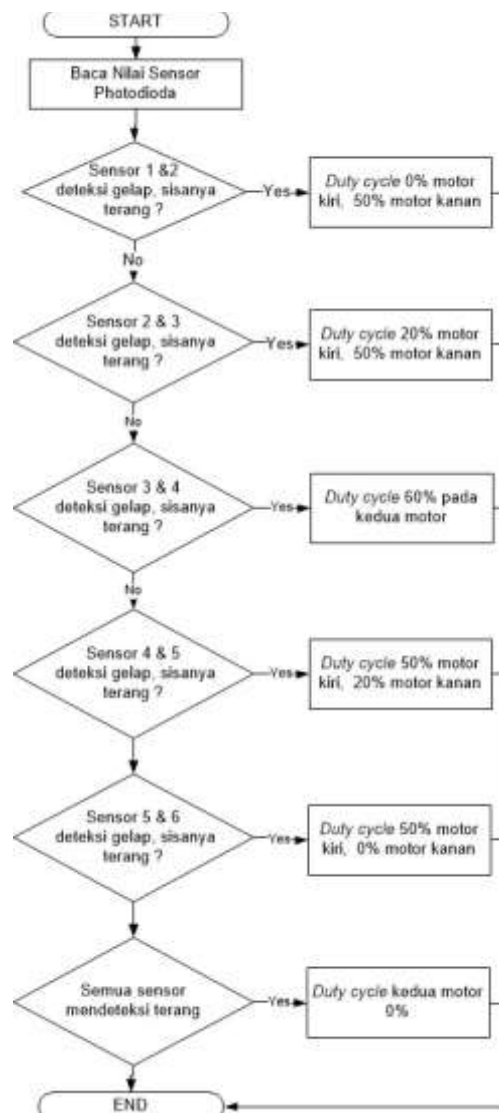
Buat sebuah aplikasi sistem kendali PWM pada robot dengan ketentuan sebagai berikut

- Buat rangkaian 6 buah sensor photodiode dengan 2 sensor dengan contoh urutan sensor seperti pada Gambar 6 (**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. 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**).

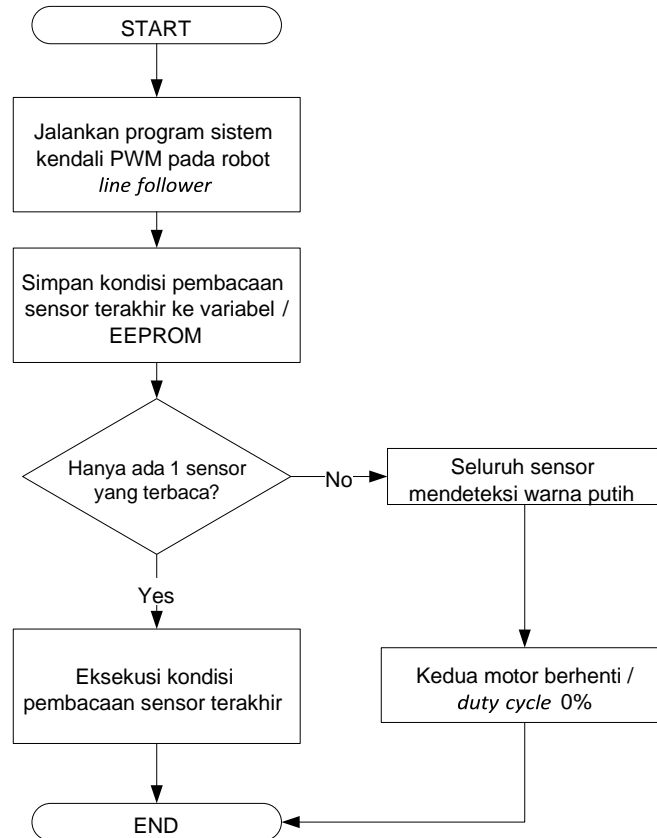
Jelaskan fungsi dari pengubahan berbagai *duty cycle* pada kondisi-kondisi di atas dan apa perbedaan yang terjadi pada masing-masing perubahan *duty cycle*!



Gambar 7 Flowchart sistem kendali PWM pada robot line follower.

## 2. 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. Jika seluruh sensor mendeteksi nilai putih maka seluruh motor harus berhenti.



Gambar 8 Flowchart program penyimpanan kondisi terakhir.

## 7. Jurnal Praktikum

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

- Judul Praktikum :
- Maksud dan Tujuan Praktikum :
- Peralatan dan Komponen Praktikum :
- Dasar Teori
- Foto Peralatan dan Bahan Praktikum :
- 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	689	689	689	689	0	4650
689	33	33	689	689	689	1858	4765
689	689	33	33	689	689	5602	5819
689	689	689	33	33	689	4565	2018

689	689	689	689	33	33	4750	0
689	689	689	689	689	689	0	0

- **Kesimpulan Praktikum**

Dengan menggunakan PWM kita dapat memperhalus gerakan dari motor, yang awalnya Cuma bisa hidup dan mati, sekarang bisa kita jalankan secara perlahan.

Pada tikungan line follower akan berbelok secara halus tanpa harus berhenti berhenti ketika akan berbelok.