

## BAB IV

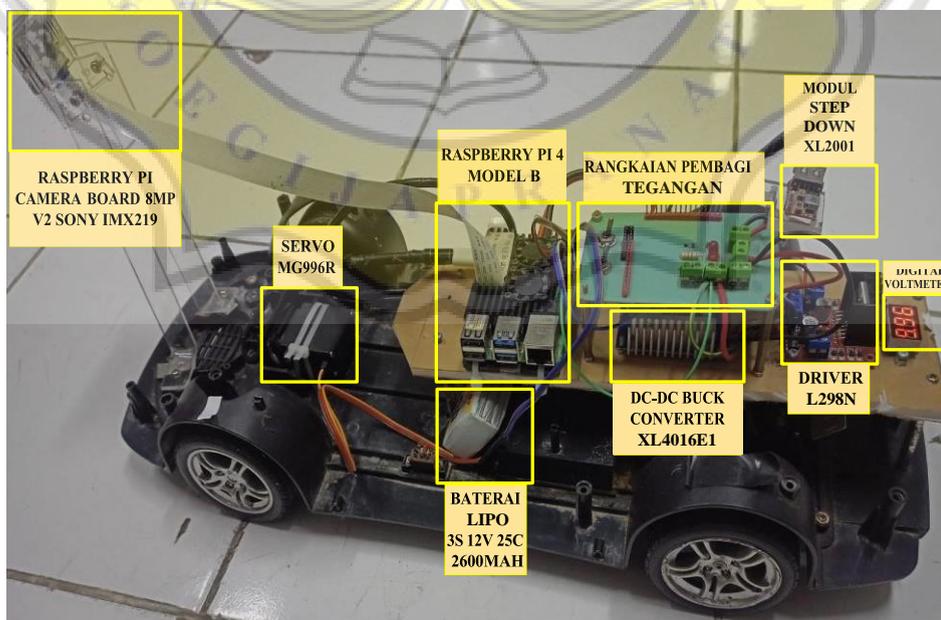
### HASIL DAN PEMBAHASAN

#### 4.1. Pendahuluan

Pada bab ini akan dijelaskan hasil hasil *prototype* robot AGV, program, dan pengujian alat yang dilakukan untuk penelitian pergerakan AGV. Kemudian akan dipaparkan mengenai sudut pergerakan AGV terhadap pola lintasan. Untuk mengatur pergerakan AGV agar dapat berpindah tempat secara presisi dapat dilakukan dengan mengatur *Pulse Width Modulation* (PWM).

#### 4.2. Prototype Alat

Dalam penelitian ini, menghasilkan bentuk *hardware* yang digunakan untuk Tugas Akhir. Hasil *hardware* ini sesuai dengan perancangan alat pada Bab III. Gambar 4.1 menunjukkan *hardware* dari AGV.



Gambar 4.1 Hardware AGV

### 4.3. Program

Berikut ini merupakan pembahasan program untuk sistem pergerakan AGV saat berpindah tempat.

```
import RPi.GPIO as GPIO
import time
from time import sleep
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
Ena, In1, In2 = 22, 17, 27
```

Langkah pertama, import *library* yang dibutuhkan. Untuk **import Rpi.GPIO as GPIO** berfungsi agar *library* dari *Open CV* aktif. Fungsi **import time** berfungsi mengatur jeda waktu. Untuk fungsi **GPIO.setmode(GPIO.BCM)** berfungsi mengubah GPIO menjadi Broadcom (BCM) pada sistem penomoran. Sedangkan **Ena, In1, In2 = 22, 17, 27** merupakan lokasi pin *driver* L298N.

```
def Pola_A():
    GPIO.setup(25, GPIO.OUT)
    pwm=GPIO.PWM(25, 360)
    pwm.start(0)
    pwm.ChangeDutyCycle(42)
    sleep(1)
    pwm.stop(1)
    GPIO.setup(Ena, GPIO.OUT)
    GPIO.setup(In1, GPIO.OUT)
    GPIO.setup(In2, GPIO.OUT)
    pwm = GPIO.PWM(Ena, 25)
    pwm.start(0)
    GPIO.output(In1, GPIO.LOW)
    GPIO.output(In2, GPIO.HIGH)
    pwm.ChangeDutyCycle(100)
    sleep (1.6)
    pwm.stop(1)

    print("BERADA DI TITIK A")
    print("")
```

Kemudian AGV dapat berpindah tempat dari P menuju A. Fungsi **pwm.ChangeDutyCycle** digunakan untuk mengatur PWM. Sedangkan **print**

menunjukkan posisi AGV berhenti. Untuk program perpindahan AGV dari lokasi

A menuju lokasi tujuan B dapat diamati seperti dibawah ini.

```
def Pola_B():
    GPIO.setup(25, GPIO.OUT)
    pwm=GPIO.PWM(25, 360)
    pwm.start(0)
    pwm.ChangeDutyCycle(42)
    sleep(1)
    pwm.stop(1)
    GPIO.setup(Ena, GPIO.OUT)
    GPIO.setup(In1, GPIO.OUT)
    GPIO.setup(In2, GPIO.OUT)
    pwm = GPIO.PWM(Ena, 25)
    pwm.start(0)
    GPIO.output(In1, GPIO.LOW)
    GPIO.output(In2, GPIO.HIGH)
    pwm.ChangeDutyCycle(100)
    sleep (0.8)
    pwm.stop(1)
    GPIO.setup(25, GPIO.OUT)
    pwm=GPIO.PWM(25, 360)
    pwm.start(0)
    pwm.ChangeDutyCycle(53)
    sleep(1)
    pwm.stop(1)
    GPIO.setup(Ena, GPIO.OUT)
    GPIO.setup(In1, GPIO.OUT)
    GPIO.setup(In2, GPIO.OUT)
    pwm = GPIO.PWM(Ena, 25)
    pwm.start(0)
    GPIO.output(In1, GPIO.LOW)
    GPIO.output(In2, GPIO.HIGH)
    pwm.ChangeDutyCycle(100)
    sleep (2.3)
    pwm.stop(1)
    print("BERADA DI TITIK B")
    print("")
```

Untuk program perpindahan AGV dari B menuju C dapat diamati seperti dibawah ini.

```
def Pola_C():
    GPIO.setup(25,GPIO.OUT)
    pwm=GPIO.PWM(25, 360)
    pwm.start(0)
    pwm.ChangeDutyCycle(31)
    sleep(1)
    pwm.stop(1)
    GPIO.setup(Ena,GPIO.OUT)
    GPIO.setup(In1,GPIO.OUT)
    GPIO.setup(In2,GPIO.OUT)
    pwm = GPIO.PWM(Ena,25)
    pwm.start(0)
    GPIO.output(In1,GPIO.LOW)
    GPIO.output(In2,GPIO.HIGH)
    pwm.ChangeDutyCycle(100)
    sleep (1.5)
    pwm.stop(1)
    GPIO.setup(25,GPIO.OUT)
    pwm=GPIO.PWM(25, 360)
    pwm.start(0)
    pwm.ChangeDutyCycle(53)
    sleep(1)
    pwm.stop(1)
    GPIO.setup(Ena,GPIO.OUT)
    GPIO.setup(In1,GPIO.OUT)
    GPIO.setup(In2,GPIO.OUT)
    pwm = GPIO.PWM(Ena,25)
    pwm.start(0)
    GPIO.output(In1,GPIO.LOW)
    GPIO.output(In2,GPIO.HIGH)
    pwm.ChangeDutyCycle(100)
    sleep (1.5)
    pwm.stop(1)
    print("BERADA DI TITIK C")
    print("")
```

Untuk program perpindahan AGV dari C menuju D dapat diamati seperti dibawah ini.

```
def PolA_D():
    GPIO.setup(25, GPIO.OUT)
    pwm=GPIO.PWM(25, 360)
    pwm.start(0)
    pwm.ChangeDutyCycle(42)
    sleep(1)
    pwm.stop(1)
    GPIO.setup(Ena, GPIO.OUT)
    GPIO.setup(In1, GPIO.OUT)
    GPIO.setup(In2, GPIO.OUT)
    pwm = GPIO.PWM(Ena, 25)
    pwm.start(0)
    GPIO.output(In1, GPIO.LOW)
    GPIO.output(In2, GPIO.HIGH)
    pwm.ChangeDutyCycle(100)
    sleep (1)
    pwm.stop(1)
    GPIO.setup(25, GPIO.OUT)
    pwm=GPIO.PWM(25, 360)
    pwm.start(0)
    pwm.ChangeDutyCycle(31)
    sleep(1)
    pwm.stop(1)
    GPIO.setup(Ena, GPIO.OUT)
    GPIO.setup(In1, GPIO.OUT)
    GPIO.setup(In2, GPIO.OUT)
    pwm = GPIO.PWM(Ena, 25)
    pwm.start(0)
    GPIO.output(In1, GPIO.LOW)
    GPIO.output(In2, GPIO.HIGH)
    pwm.ChangeDutyCycle(100)
    sleep (1.2)
    pwm.stop(1)
    print("BERADA DI TITIK D")
    print("")
```

Untuk program perpindahan AGV dari D menuju E dapat diamati seperti dibawah ini.

```
def Pola_E():
    GPIO.setup(25,GPIO.OUT)
    pwm=GPIO.PWM(25, 360)
    pwm.start(0)
    pwm.ChangeDutyCycle(31)
    sleep(1)
    pwm.stop(1)
    GPIO.setup(Ena,GPIO.OUT)
    GPIO.setup(In1,GPIO.OUT)
    GPIO.setup(In2,GPIO.OUT)
    pwm = GPIO.PWM(Ena,25)
    pwm.start(0)
    GPIO.output(In1,GPIO.LOW)
    GPIO.output(In2,GPIO.HIGH)
    pwm.ChangeDutyCycle(100)
    sleep (1.2)
    pwm.stop(1)
    GPIO.setup(25,GPIO.OUT)
    pwm=GPIO.PWM(25, 360)
    pwm.start(0)
    pwm.ChangeDutyCycle(53)
    sleep(1)
    pwm.stop(1)
    GPIO.setup(Ena,GPIO.OUT)
    GPIO.setup(In1,GPIO.OUT)
    GPIO.setup(In2,GPIO.OUT)
    pwm = GPIO.PWM(Ena,25)
    pwm.start(0)
    GPIO.output(In1,GPIO.LOW)
    GPIO.output(In2,GPIO.HIGH)
    pwm.ChangeDutyCycle(100)
    sleep (1.2)
    pwm.stop(1)
    print("BERADA DI TITIK E")
    print("")
```

Untuk program perpindahan AGV dari E menuju F dapat diamati seperti dibawah ini.

```
def Pola_F():
    GPIO.setup(25, GPIO.OUT)
    pwm=GPIO.PWM(25, 360)
    pwm.start(0)
    pwm.ChangeDutyCycle(42)
    sleep(1)
    pwm.stop(1)
    GPIO.setup(Ena, GPIO.OUT)
    GPIO.setup(In1, GPIO.OUT)
    GPIO.setup(In2, GPIO.OUT)
    pwm = GPIO.PWM(Ena, 25)
    pwm.start(0)
    GPIO.output(In1, GPIO.LOW)
    GPIO.output(In2, GPIO.HIGH)
    pwm.ChangeDutyCycle(100)
    sleep (0.8)
    pwm.stop(1)
    GPIO.setup(25, GPIO.OUT)
    pwm=GPIO.PWM(25, 360)
    pwm.start(0)
    pwm.ChangeDutyCycle(48)
    sleep(1)
    pwm.stop(1)
    GPIO.setup(Ena, GPIO.OUT)
    GPIO.setup(In1, GPIO.OUT)
    GPIO.setup(In2, GPIO.OUT)
    pwm = GPIO.PWM(Ena, 25)
    pwm.start(0)
    GPIO.output(In1, GPIO.LOW)
    GPIO.output(In2, GPIO.HIGH)
    pwm.ChangeDutyCycle(100)
    sleep (0.9)
    pwm.stop(1)
    print("BERADA DI TITIK F")
    print("")
```

Untuk program perpindahan AGV dari F menuju G dapat diamati seperti dibawah ini.

```
def Pola_G():
    GPIO.setup(25,GPIO.OUT)
    pwm=GPIO.PWM(25, 360)
    pwm.start(0)
    pwm.ChangeDutyCycle(42)
    sleep(1)
    pwm.stop(1)
    GPIO.setup(Ena,GPIO.OUT)
    GPIO.setup(In1,GPIO.OUT)
    GPIO.setup(In2,GPIO.OUT)
    pwm = GPIO.PWM(Ena,25)
    pwm.start(0)
    GPIO.output(In1,GPIO.LOW)
    GPIO.output(In2,GPIO.HIGH)
    pwm.ChangeDutyCycle(100)
    sleep (1.3)
    pwm.stop(1)
    GPIO.setup(25,GPIO.OUT)
    pwm=GPIO.PWM(25, 360)
    pwm.start(0)
    pwm.ChangeDutyCycle(53)
    sleep(1)
    pwm.stop(1)
    GPIO.setup(Ena,GPIO.OUT)
    GPIO.setup(In1,GPIO.OUT)
    GPIO.setup(In2,GPIO.OUT)
    pwm = GPIO.PWM(Ena,25)
    pwm.start(0)
    GPIO.output(In1,GPIO.LOW)
    GPIO.output(In2,GPIO.HIGH)
    pwm.ChangeDutyCycle(100)
    sleep (1.9)
    pwm.stop(1)
    print("BERADA DI TITIK G")
    print("")
```

Kemudian untuk program perpindahan AGV dari G kembali ke P dapat diamati seperti dibawah ini.

```
def Pola_P():
    GPIO.setup(25, GPIO.OUT)
    pwm=GPIO.PWM(25, 360)
    pwm.start(0)
    pwm.ChangeDutyCycle(42)
    sleep(1)
    pwm.stop(1)
    GPIO.setup(Ena, GPIO.OUT)
    GPIO.setup(In1, GPIO.OUT)
    GPIO.setup(In2, GPIO.OUT)
    pwm = GPIO.PWM(Ena, 25)
    pwm.start(0)
    GPIO.output(In1, GPIO.LOW)
    GPIO.output(In2, GPIO.HIGH)
    pwm.ChangeDutyCycle(100)
    sleep (1)
    pwm.stop(1)
    GPIO.setup(25, GPIO.OUT)
    pwm=GPIO.PWM(25, 360)
    pwm.start(0)
    pwm.ChangeDutyCycle(31)
    sleep(1)
    pwm.stop(1)
    GPIO.setup(Ena, GPIO.OUT)
    GPIO.setup(In1, GPIO.OUT)
    GPIO.setup(In2, GPIO.OUT)
    pwm = GPIO.PWM(Ena, 25)
    pwm.start(0)
    GPIO.output(In1, GPIO.LOW)
    GPIO.output(In2, GPIO.HIGH)
    pwm.ChangeDutyCycle(100)
    sleep (1.2)
    pwm.stop(1)
    print("BERADA DI TITIK P")
    print("")
```

#### 4.4. Hasil Pengukuran Kestabilan AGV terhadap Kecepatan

Hasil pengukuran berikut ini dilaksanakan melalui beberapa pengukuran pada pola lintasan AGV. Pengukuran kestabilan AGV dapat diamati pada Tabel 1. Kemudian untuk mengukur *Pulse Width Modulation* (PWM) memakai osiloskop. Parameter pengukuran berupa sinyal PWM, dan RPM menggunakan *tachometer digital*. Dilihat pada Tabel 1, yang mempengaruhi kecepatan dan kestabilan AGV ditentukan oleh *Pulse Width Modulation* (PWM). Oleh karena itu, untuk mencapai hasil yang perpindahan AGV yang tepat diperlukan pengaturan PWM sesuai program yang dibuat. Semakin kecil nilai PWM maka akan semakin lambat kecepatan AGV serta memiliki kondisi yang stabil. Sedangkan jika mengatur PWM dengan nilai yang tinggi maka AGV akan melaju dengan cepat namun memiliki kondisi yang tidak stabil. Jika membutuhkan AGV dengan kondisi yang stabil dan tidak terlalu cepat PWM dapat diatur antara 50 % sampai dengan 60% sehingga AGV pada kondisi kecepatan sedang.

Tabel 1. Tabel Kestabilan AGV terhadap Kecepatan

PWM Duty Cycle (%)	Kestabilan AGV	Kondisi Kecepatan AGV
30	stabil	lambat
40	stabil	lambat
50	stabil	sedang
60	stabil	sedang
70	stabil	cepat
80	stabil	cepat
90	tidak stabil	cepat

#### 4.5. Hasil Pengukuran Selisih Sudut antara AGV dengan Pola Lintasan

Dalam penelitian ini, yang difokuskan adalah sistem pergerakan AGV dari suatu tempat ke tempat lainnya dan mengukur selisih sudut antara AGV dengan pola lintasan. Hal ini dilakukan supaya mengetahui hasil yang sebenarnya melalui tiga kali percobaan. Langkah yang perlu diperhatikan adalah proses peletakan AGV sangat mempengaruhi posisi akhir AGV hal ini dapat disesuaikan melalui kalibrasi AGV. Hasil pengujian pada Tabel 2 menyajikan selisih sudut rata-rata antara sudut pergerakan AGV dengan acuan sudut sebesar  $3.09^\circ$ . Dengan selisih sudut tersebut menandakan *error* kecil sehingga AGV dapat beroperasi secara optimal.

Tabel 2. Tabel Selisih Sudut antara AGV dengan Pola Lintasan

Perpindahan posisi AGV	Sudut Acuan	Percobaan ke-			Selisih Sudut
		1	2	3	
P ke A	$90^\circ$	$92,5^\circ$	$92,9^\circ$	$93^\circ$	$2,8^\circ$
A ke B	$110^\circ$	$112,8^\circ$	$112,9^\circ$	$112,8^\circ$	$2,83^\circ$
B ke C	$90^\circ$	$93,1^\circ$	$93,2^\circ$	$93,1^\circ$	$3,13^\circ$
C ke D	$110^\circ$	$113,2^\circ$	$113,3^\circ$	$113,4^\circ$	$3,3^\circ$
D ke E	$90^\circ$	$93,3^\circ$	$92,8^\circ$	$93,1^\circ$	$3,06^\circ$
E ke F	$110^\circ$	$113,2^\circ$	$113,3^\circ$	$113,1^\circ$	$3,2^\circ$
F ke G	$100^\circ$	$103,2^\circ$	$103,1^\circ$	$102,8^\circ$	$3,03^\circ$
G ke P	$70^\circ$	$72,9^\circ$	$73,3^\circ$	$74,1^\circ$	$3,43^\circ$
Selisih sudut rata-rata					$3,09^\circ$

## 4.6. Hasil Pengujian

Berikut ini akan dijelaskan beberapa hasil pengujian Hasil pengukuran berikut ini dilaksanakan melalui beberapa pengukuran pada pola lintasan AGV. untuk mengukur *Pulse Width Modulation* (PWM) memakai osiloskop. Parameter pengukuran berupa sinyal PWM, dan RPM menggunakan tachometer digital.

### 4.6.1. Pengukuran AGV Belok Kanan

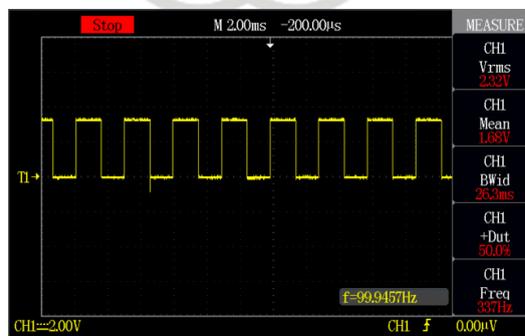
Pengukuran dilaksanakan dengan cara mengatur *duty cycle* 31%. Hasil pengukuran dapat diamati seperti pada Gambar 4.2.



Gambar 4.2 AGV Belok Kanan dengan *Duty Cycle* 31%

### 4.6.2. Pengukuran AGV Belok Kiri

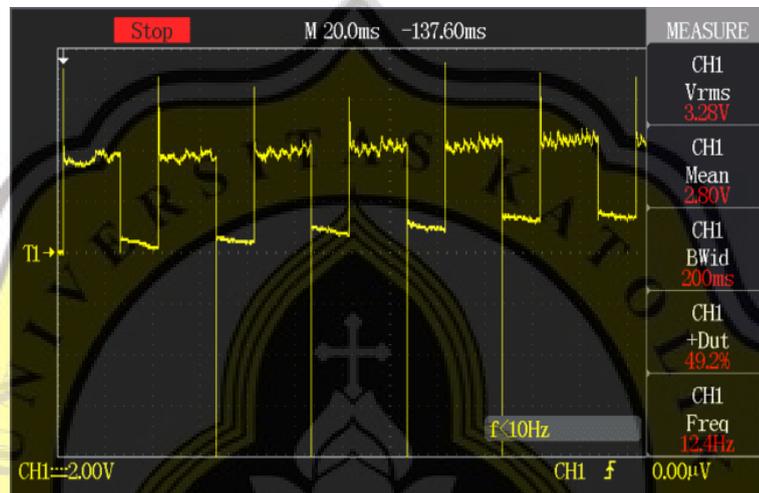
Pengukuran dilaksanakan dengan cara mengatur *duty cycle* 50%. Hasil pengukuran dapat dilihat pada Gambar 4.3.



Gambar 4.3 AGV Belok Kiri dengan *Duty Cycle* 50%

### 4.6.3. Pengukuran AGV Kecepatan Sedang

Pengukuran dilaksanakan dengan cara mengatur *duty cycle* 49.2%. Hasil pengukuran dapat dilihat pada Gambar 4.4. Dan untuk kecepatan sedang kecepataannya adalah 539.5 RPM seperti pada Gambar 4.5.



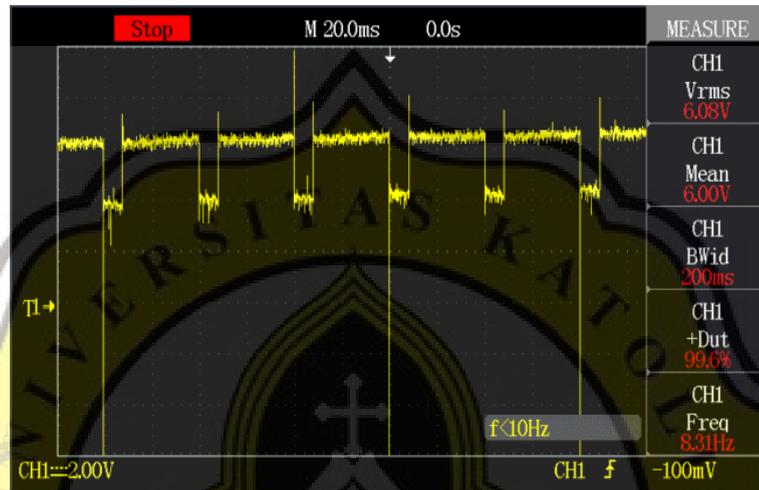
Gambar 4.4 PWM pada Motor DC dengan *Duty Cycle* 49.2%



Gambar 4.5 AGV dengan Kecepatan Sedang 539.5 RPM

#### 4.6.4. Pengukuran AGV Kecepatan Cepat

Pengukuran dilaksanakan dengan cara mengatur *duty cycle* 99,6%. Hasil pengukuran dapat diamati pada Gambar 4.6. Dan untuk kecepatan cepat kecepataannya adalah 889.7 RPM yang dapat diamati pada Gambar 4.7.



Gambar 4.6 PWM pada Motor DC dengan *Duty Cycle* 99.6%



Gambar 4.7 AGV dengan Kecepatan Cepat 889.7 RPM