

## BAB IV

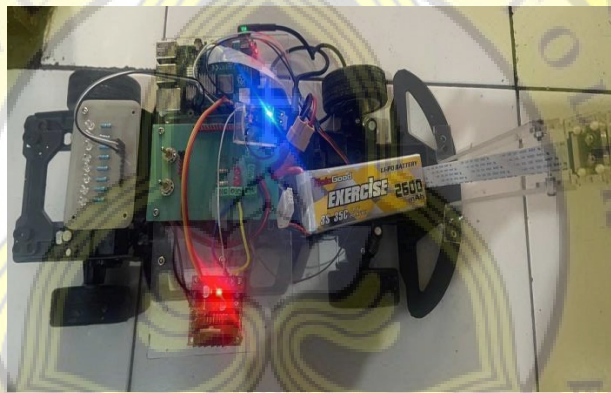
### HASIL DAN PEMBAHASAN

#### 4.1. Pendahuluan

Pada bab ini akan dibahas mengenai hal-hal tentang *prototype* alat, program, Sebelumnya, membuat robot dapat berpindah tempat yang akan ditentukan dengan baik

#### 4.2. *Prototype* Alat

Dalam penelitian ini, dapat dihasilkan bentuk *prototype* alat untuk laporan tugas akhir:



Gambar 4.1 Hardware AGV

#### 4.3. Program

Program dibagi menjadi beberapa pembahasan, yaitu :

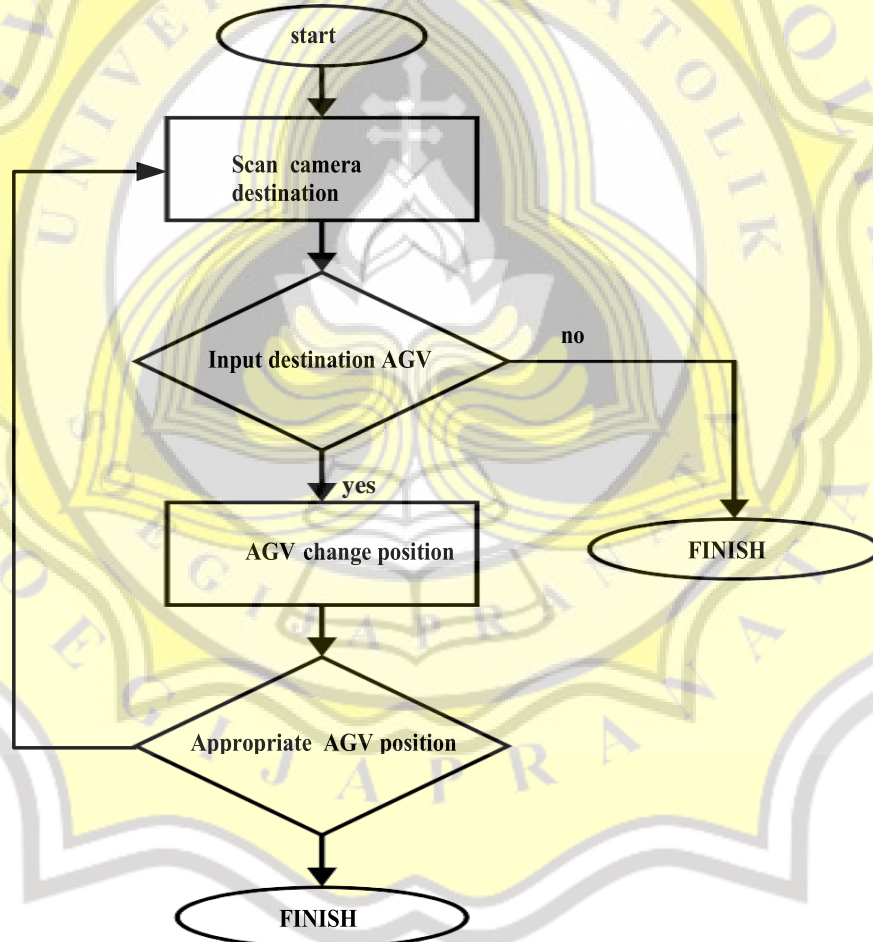
##### 4.3.1 Deskripsi

Prioritas project ini adalah membahas sistem mobilitas robot AGV. Lingkaran Penggerak robot AGV menggunakan roda konvensional agar lebih mudah gerakan manuver. Bagian roda menggunakan roda kendaraan konvensional sebagai kendaraannya Ini mendukung dan memaksimalkan sistem 2WD. Robot AGV menggunakan motor DC. Oleh karena itu, seorang pengemudi motor diperlukan L298N, digunakan untuk memudahkan pengaturan pergerakan roda digunakan di robot AGV. Jadi L298N

membutuhkan voltase masukan 10 volt arus searah. Duty cycle control adalah salah satu cara untuk mengontrol roda sehingga kecepatan memenuhi persyaratan. Ketika roda berputar dan tidak berputar diberikan batas berhenti, itu tidak bergerak ke posisi yang dimaksud. Lalu tambahkan Fungsi pengatur waktu diperlukan. Ini untuk membatasi putaran roda saat robot AGV mengubah posisinya. Dengan fungsi penghitung waktu ini, robot akan turun rute yang diinginkan dan berhenti tepat di tempat tujuan.

#### 4.3.2 Flowchart Program

Berikut ini adalah pembahasan program sistem pergerakan AGV saat berpindah tempat seperti pada Gambar 4.2.



Gambar 4.2 Flowchart Program Jalan Saat Deteksi Pola

#### 4.3.3 Penjelasan Program

Berikut ini adalah pembahasan program sistem pergerakan AGV saat berpindah tempat.

```
TA1.py X
1 import cv2
2 import pytesseract
3 from pytesseract import Output
4 from TA2 import *
5 import RPi.GPIO as GPIO # importing GPIO library
6 import time # importing time library for delay
7
8
9 cap = cv2.VideoCapture(0)
10 cap.set(cv2.CAP_PROP_BUFFERSIZE, 1)
11
```

Gambar 4.3 Program awal

Langkah awal adalah mengimpor perpustakaan yang diperlukan. Pengiriman Sebuah fungsi untuk memasukkan perpustakaan termasuk cv2, oleh tesseract, dan impor adalah fungsi untuk memasukkan program terpisah yang dibuat sebelumnya.

```
14 while True:
15     # Capture frame-by-frame
16     ret, frame = cap.read()
17     d = pytesseract.image_to_data(frame, output_type=Output.DICT)
18     n_boxes = len(d['text'])
19     for i in range(n_boxes):
20         if int(d['conf'][i]) > 60:
21             (text, x, y, w, h) = (d['text'][i], d['left'][i], d['top'][i])
22             # don't show empty text
23             if text and text.strip() != "":
24                 frame = cv2.rectangle(frame, (x, y), (x + w, y + h), (0,
25                 frame = cv2.putText(frame, text, (x, y - 10), cv2.FONT_HE
26                 #print('capture point Hasil: ', text)
27                 key=cv2.waitKey(1)
28                 #key = cv2.waitKey(0)
29                 if key == ord('q'):
```

Gambar 4.4 Program Utama

Seperti terlihat pada gambar di atas, ini merupakan bagian dari program utama membaca teks yang akan diimplementasikan dalam AGV ini. Program memiliki beberapa pengolahan dengan menambahkan tampilan hasil pembacaan yang akan ditampilkan pada monitor. Agar pembacaan prediksi ditampilkan langsung di monitor.



```
28     #key = cv2.waitKey(0)
29     if key == ord('q'):
30         print("Capture pola! Hasil=", text )
31
32         if text == 'titikA':
33             time.sleep(1)
34             #print("AGV parkir di pola", text)
35             print("")
36             time.sleep(1)
37             tujuan = input ("Pergi ke mana ? ")
38             print("")
39
40             if tujuan == "A": # Tujuan A
41                 Pola_A()
42             if tujuan == "B": # Tujuan B
43                 Pola_B()
44             if tujuan == "C": # Tujuan C
```

Gambar 4.5 Program Pergerakan AGV dari Titik A ke B

Program yang digambarkan pada gambar di atas dapat memindahkan AGV ketujuannya dengan program pergerakan AGV yang telah dibuat sebelumnya. Program ini merupakan program pendeteksi titik A dan program tujuan titik B.

```
bismillah loss (raspberrypi) - VNC Viewer
TA1.py ✕
53     if text == 'titikB':
54         time.sleep(1)
55         #print("AGV parkir di pola", text)
56         print("")
57         time.sleep(1)
58         tujuan = input ("Pergi ke mana ? ")
59         print("")
60         if tujuan == "A": # Tujuan A
61             Pola_A()
62         if tujuan == "B": # Tujuan B
63             Pola_B()
64         if tujuan == "C": # Tujuan C
65             Pola_C()
66         if tujuan == "D": # Tujuan D
```

Gambar 4.6 Program Pergerakan AGV dari Titik B ke C

Program diatas adalah program deteksi titik B dan tujuan ketitik C

```
milah loss (raspberrypi) - VNC Viewer
A1.py x
72         if text == 'titikC':
73             time.sleep(1)
74             #print("AGV parkir di pola", text)
75             print("")
76             time.sleep(1)
77             tujuan = input ("Pergi ke mana ? ")
78             print("")
79             if tujuan == "A": # Tujuan A
80                 Pola_A()
81             if tujuan == "B": # Tujuan B
82                 Pola_B()
83             if tujuan == "C": # Tujuan C
84                 Pola_C()
85             if tujuan == "D": # Tujuan D
86                 Pola_D()
```

Gambar 4.7 Program Pergerakan AGV dari Titik C ke D

Program diatas adalah program deteksi titik C dan tujuan ketitik D

```
259
260     # Display the resulting frame
261     cv2.imshow('frame', frame)
262     cv2.waitKey(1)
263     #key = cv2.waitKey(0)
264
265     #key=cv2.waitKey(1)
266     #key = cv2.waitKey(0)
267     #if key == ord('q'):
268         # print("Capture pola!Hasil=",text )
269
270
271
272     # When everything done, release the capture
273     cap.release()
274     cv2.destroyAllWindows()
```

Gambar 4.8 Program Akhir

Gambar diatas ini adalah program akhir yang digunakan pada program pembacaan teks

#### 4.4 Hasil pengukuran

Berikut adalah hasil dari berbagai pengukuran terkait AGV yang telah dilakukan. Anda dapat melihat informasi stabilitas AGV pada Tabel 1. Sebaliknya, Perbedaan Sudut Tabel 2 antara AGV dan Pole Track dapat dilihat. Selanjutnya, osiloskop digunakan untuk memonitor Pulse Width Modulation (PWM). Pengukuran parameter menggunakan PWM tunggal.

Tabel 1. Tabel stabilitas AGV terhadap kecepatan

PWM duty cycle(%)	Stabilitas AGV	Kondisi kecepatan AGV
30	Stabil	lambat
40	Stabil	lambat
50	Stabil	sedang
60	Stabil	sedang
70	Stabil	cepat
80	Stabil	cepat

Tabel 2. Perbedaan sudut antara AGV dengan pola lintasan

Transfer posisi AGV	Referensi sudut	Percobaan ke -			Perbedaan sudut
		1	2	3	
P ke A	90°	92,5°	92,9°	93°	2,8°
A ke B	110°	112,8°	112,9°	112,8°	2,83°
B ke C	90°	93,1°	93,2°	93,1°	3,13°
C ke D	110°	113,2°	113,3°	113,4°	3,3°
<u>Perbedaan sudut rata-rata</u>					<u>3,015°</u>

Menurut hasil pengujian pada Tabel 2, terdapat perbedaan sudut rata-rata 3,09° antara sudut referensi dan sudut pergerakan AGV. Perbedaan sudut ini menunjukkan adanya sedikit kesalahan, sehingga AGV dapat berfungsi dengan baik.

Hasil pengukuran Modulasi Lebar Pulsa (PWM) yang dilakukan dengan osiloskop ditunjukkan di bawah ini. Sinyal PWM mewakili parameter pengukuran.

#### 1. Pengukuran AGV belok kanan

Pengukuran dilakukan dengan mengatur duty cycle sebesar 31%. Hasil pengukuran dapat diamati seperti pada Gambar 4.9

2. Pengukuran AGV belok kiri Pengukuran dilakukan dengan mengatur duty cycle sebesar 50%. Hasil pengukuran dapat dilihat pada Gambar 4.10.

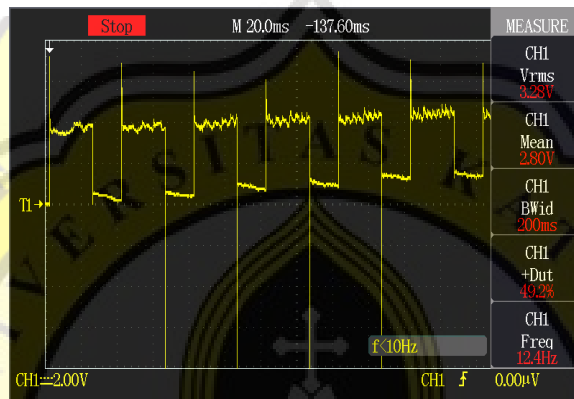
#### 3. Pengukuran AGV kecepatan sedang



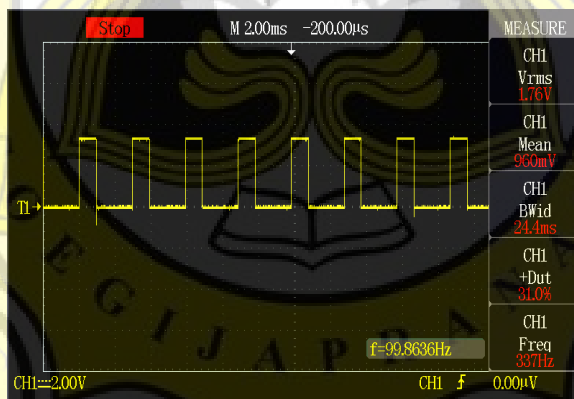
Pengukuran dilakukan dengan mengatur duty cycle sebesar 49,2%. Hasil pengukuran dapat dilihat pada Gambar 4.11.

#### 4. Pengukuran AGV kecepatan cepat

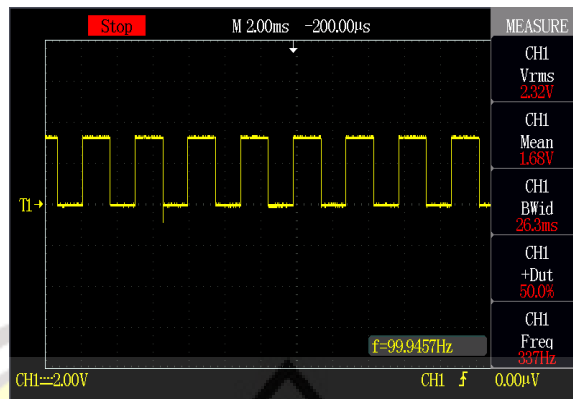
Pengukuran dilakukan dengan mengatur duty cycle sebesar 99,6%. Hasil pengukuran dapat diamati pada Gambar 4.12.



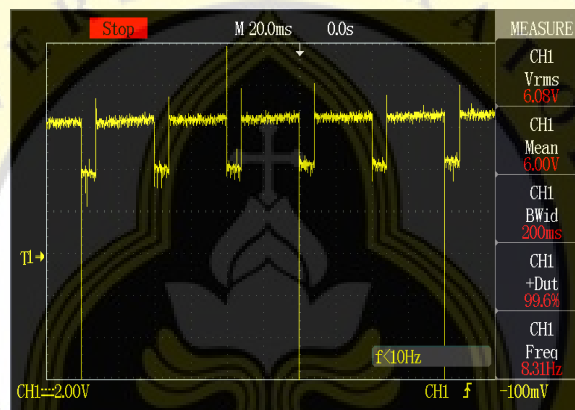
Gambar 4.9. AGV belok kanan dengan duty cycle 31%



Gambar 4.10. AGV belok kiri dengan duty cycle 50%.



Gambar 4.11. PWM pada motor DC dengan duty cycle 49,2%



Gambar 4.12. PWM pada motor DC dengan duty cycle 99,6%