

BAB IV

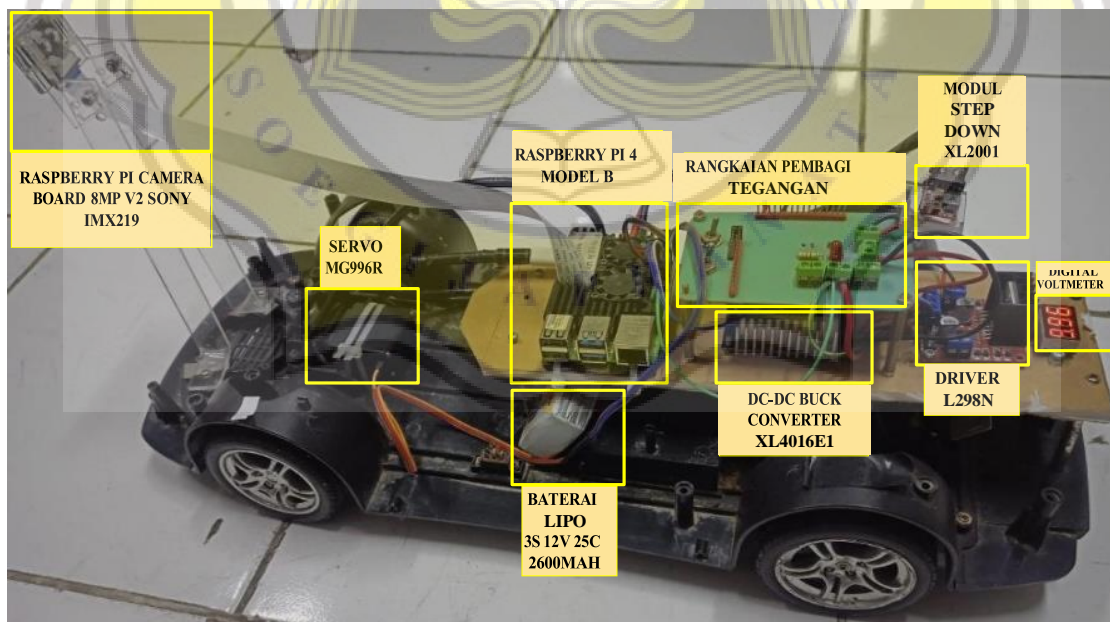
HASIL DAN PEMBAHASAN

4.1. Pendahuluan

Pada bab ini akan dijelaskan hasil hasil *prototype* robot AGV, program, dan pengujian alat yang dilakukan untuk penelitian pembacaan teks yang digunakan pada AGV . Uji hasil dan pembahasan ini akan menyajikan hasil prototype Alat, Program yang digunakan, dan perbandingan sistem Default OCR dengan OCR yang sudah dimodifikasi atau disempurnakan.

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.

```
untitled> x LIVE.py x
1 import cv2
2 import pytesseract
3 from pytesseract import Output
4 from PBU import *
5
6
7 cap = cv2.VideoCapture(0)
8 cap.set(cv2.CAP_PROP_BUFFERSIZE, 1)
9
```

Langkah pertama, import *library* yang dibutuhkan. Fungsi *import* Adalah untuk memasukkan library diantaranya ada cv2 , pytesseract , import sendiri adalah fungsi untuk memasukkan program tersendiri yang sebelumnya dibuat.

```
<untitled> x LIVE.py x
--
11 while True:
12     # Capture frame-by-frame
13     ret, frame = cap.read()
14
15     d = pytesseract.image_to_data(frame, output_type=Output.TEXT)
16     n_boxes = len(d['text'])
17     for i in range(n_boxes):
18         if int(d['conf'][i]) > 60:
19             (text, x, y, w, h) = (d['text'][i], d['left'][i],
20                                 d['top'][i], d['width'][i], d['height'][i])
21             # don't show empty text
22             if text and text.strip() != "":
23                 frame = cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
24                 frame = cv2.putText(frame, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0))
25                 #print('capture pola!Hasil',text )
26                 key=cv2.waitKey(1)
27                 #key = cv2.waitKey(0)
28                 if key == ord('q'):
29                     print("Capture pola!Hasil=",text )
```

Dilihat dari gambar diatas adalah bagian program utama untuk pembacaan teks yang diimplementasikan dalam AGV ini dimana program diatas ini berisi penambahan prosesing dengan menambahkan tampilan hasil pembacaan yang akan ditampilkan dalam layar monitor. Sehingga prediksi pembacaan akan langsung ditampilkan pada monitor .

```
<untitled> x LIVE.py x
30         if text == 'titika': # Deteksi Pola P
31             time.sleep(1)
32             #print("AGV parkir di pola", text)
33             print("")
34             time.sleep(1)
35             tujuan = input ("Pergi ke mana ? ")
36             print("")
37
38         if tujuan == "A": # Tujuan A
39             Pola_A()
40         if tujuan == "B": # Tujuan B
41             Pola_B()
42         if tujuan == "C": # Tujuan C
43             Pola_C()
44         if tujuan == "D": # Tujuan D
45             Pola_D()
46         if tujuan == "E": # Tujuan B
47             Pola_E()
48         if tujuan == "A2": # Tujuan C
49             Pola_A2()
50         if tujuan == "F": # Tujuan D
51             Pola_F()
52         if tujuan == "G": # Tujuan C
```

Program pada gambar diatas ini adalah program yang memiliki fungsi untuk menggerakkan AGV ke tujuan sesuai dengan program pergerakan AGV yang sebelumnya sudah dibuat. Program diatas adalah program deteksi titik A dan tujuan ke titik B.

```
<untitled> x LIVE.py x
60     if text == 'titikB': # Deteksi Pola P
61         time.sleep(1)
62         #print("AGV parkir di pola", text)
63         print("")
64         time.sleep(1)
65         tujuan = input ("Pergi ke mana ? ")
66         print("")
67         if tujuan == "A": # Tujuan A
68             Pola_A()
69         if tujuan == "B": # Tujuan B
70             Pola_B()
71         if tujuan == "C": # Tujuan C
72             Pola_C()
73         if tujuan == "D": # Tujuan D
74             Pola_D()
75         if tujuan == "E": # Tujuan B
76             Pola_E()
77         if tujuan == "A2": # Tujuan C
78             Pola_A2()
79         if tujuan == "F": # Tujuan D
80             Pola_F()
81         if tujuan == "G": # Tujuan C
82             Pola_G()
```

Program diatas adalah program deteksi titik B dan tujuan ke titik C

```
<untitled> x LIVE.py x
88
89     if text == 'titikC': # Deteksi Pola P
90         time.sleep(1)
91         #print("AGV parkir di pola", text)
92         print("")
93         time.sleep(1)
94         tujuan = input ("Pergi ke mana ? ")
95         print("")
96         if tujuan == "A": # Tujuan A
97             Pola_A()
98         if tujuan == "B": # Tujuan B
99             Pola_B()
100        if tujuan == "C": # Tujuan C
101            Pola_C()
102        if tujuan == "D": # Tujuan D
103            Pola_D()
104        if tujuan == "E": # Tujuan B
105            Pola_E()
106        if tujuan == "A2": # Tujuan C
107            Pola_A2()
108        if tujuan == "F": # Tujuan D
109            Pola_F()
110        if tujuan == "G": # Tujuan C
111            Pola_G()
```

Program diatas adalah program deteksi titik C dan tujuan ke titik D

```
<untitled> x LIVE.py x
116
117     if text == 'titikD': # Deteksi Pola P
118         time.sleep(1)
119         #print("AGV parkir di pola", text)
120         print("")
121         time.sleep(1)
122         tujuan = input ("Pergi ke mana ? ")
123         print("")
124         if tujuan == "A": # Tujuan A
125             Pola_A()
126         if tujuan == "B": # Tujuan B
127             Pola_B()
128         if tujuan == "C": # Tujuan C
129             Pola_C()
130         if tujuan == "D": # Tujuan D
131             Pola_D()
132         if tujuan == "E": # Tujuan B
133             Pola_E()
134         if tujuan == "A2": # Tujuan C
135             Pola_A2()
136         if tujuan == "F": # Tujuan D
137             Pola_F()
138         if tujuan == "G": # Tujuan C
```

Program diatas adalah program deteksi titik D dan tujuan ke titik E

```
<untitled> x LIVE.py x
145
146     if text == 'titikE': # Deteksi Pola P
147         time.sleep(1)
148         #print("AGV parkir di pola", text)
149         print("")
150         time.sleep(1)
151         tujuan = input ("Pergi ke mana ? ")
152         print("")
153         if tujuan == "A": # Tujuan A
154             Pola_A()
155         if tujuan == "B": # Tujuan B
156             Pola_B()
157         if tujuan == "C": # Tujuan C
158             Pola_C()
159         if tujuan == "D": # Tujuan D
160             Pola_D()
161         if tujuan == "E": # Tujuan B
162             Pola_E()
163         if tujuan == "A2": # Tujuan C
164             Pola_A2()
165         if tujuan == "F": # Tujuan D
166             Pola_F()
167         if tujuan == "G": # Tujuan C
```

Program diatas adalah program deteksi titik E dan tujuan ke titik F

```
<untitled> x LIVE.py x
173
174     if text == 'titikF': # Deteksi Pola P
175         time.sleep(1)
176         #print("AGV parkir di pola", text)
177         print("")
178         time.sleep(1)
179         tujuan = input ("Pergi ke mana ? ")
180         print("")
181         if tujuan == "A": # Tujuan A
182             Pola_A()
183         if tujuan == "B": # Tujuan B
184             Pola_B()
185         if tujuan == "C": # Tujuan C
186             Pola_C()
187         if tujuan == "D": # Tujuan D
188             Pola_D()
189         if tujuan == "E": # Tujuan B
190             Pola_E()
191         if tujuan == "A2": # Tujuan C
192             Pola_A2()
193         if tujuan == "F": # Tujuan D
194             Pola_F()
195         if tujuan == "G": # Tujuan C
```

Program diatas adalah program deteksi titik F dan tujuan ke titik G

```
<untitled> x LIVE.py x
203
204     if text == 'titikG': # Deteksi Pola P
205         time.sleep(1)
206         #print("AGV parkir di pola", text)
207         print("")
208         time.sleep(1)
209         tujuan = input ("Pergi ke mana ? ")
210         print("")
211         if tujuan == "A": # Tujuan A
212             Pola_A()
213         if tujuan == "B": # Tujuan B
214             Pola_B()
215         if tujuan == "C": # Tujuan C
216             Pola_C()
217         if tujuan == "D": # Tujuan D
218             Pola_D()
219         if tujuan == "E": # Tujuan B
220             Pola_E()
221         if tujuan == "A2": # Tujuan C
222             Pola_A2()
223         if tujuan == "F": # Tujuan D
224             Pola_F()
225         if tujuan == "G": # Tujuan C
226             Pola_G()
```

Program diatas adalah program deteksi titik F dan tujuan ke titik G

```

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

4.4. Perbandingan Default OCR dan Enhanced OCR

Pada pembahasan kali ini, saya akan membandingkan pembacaan teks dengan OCR bawaan operation engine mode (OEM) dan page segmentation mode (PSM) yang kemudian akan saya bandingkan dengan OCR yang saya modifikasi untuk meminimalisir kesalahan pembacaan. Dalam perbandingan ini, saya menggunakan teks tunggal yang bertuliskan "titikA" dan "titikB" pada OCR default saya menggunakan pengaturan OEM-3 dan PSM-7 yang memiliki fungsi untuk membaca satu teks.



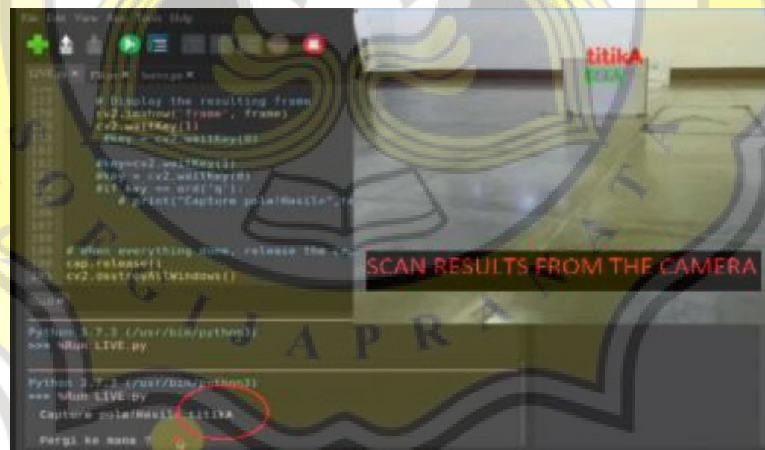
(a)



(b)

Gambar 4.2 Pembacaan Teks dengan Default OCR

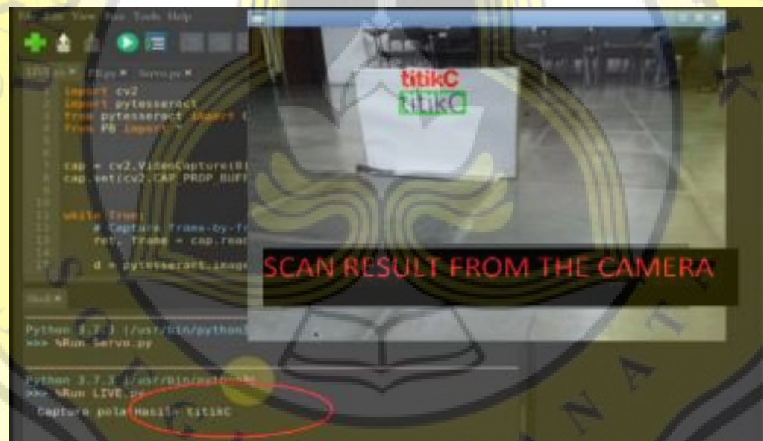
Seperti terlihat pada Gambar 8, pembacaan teks pada bagian (a) posisi teks diambil dari jarak yang jauh dan menghasilkan kata-kata yang tidak tepat, sedangkan pada Gambar 8 bagian (b) menunjukkan teks diambil pada jarak yang sangat dekat, jika gambar diambil pada jarak yang sangat dekat, maka akan menghasilkan kata-kata yang sesuai dengan teks yang diambil.



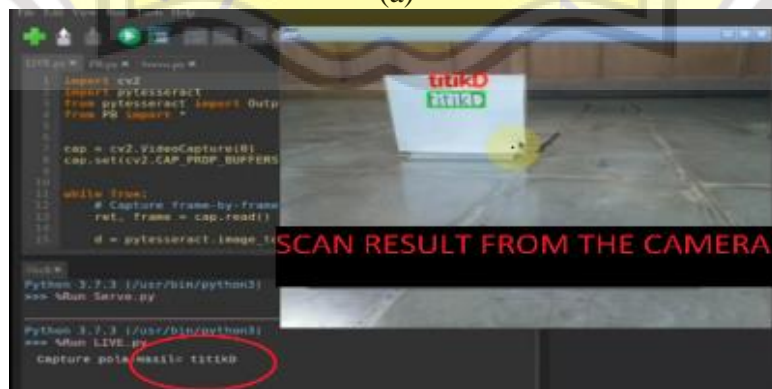
Gambar 4.3 Pembacaan dengan OCR yang disempurnakan

pembacaan yang kita dapatkan dengan OCR yang ditingkatkan ini, dapat membaca teks dengan benar dan dapat digunakan pada jarak kurang lebih 1 - 2 meter. OCR yang ditingkatkan ini juga meminimalkan kesalahan membaca hingga 95% jika kata-kata yang dibaca tidak sesuai dengan teks, tidak akan ditampilkan pada layar keluaran kamera.

Pada Gambar 4.4 bagian (a) adalah pendeteksian teks menggunakan OCR default dimana hasil pada bagian (a) tidak dapat menghasilkan kata-kata yang sesuai dengan yang ditangkap oleh kamera. Pembacaan teks pada bagian (a) kurang tepat karena jarak deteksi teks dapat dikatakan cukup jauh sehingga menyebabkan kesalahan pembacaan. Contohnya dapat dilihat pada bagian (c) menggunakan OCR yang disempurnakan dan menghasilkan pembacaan yang sesuai yang dapat dideteksi dalam jarak kurang lebih 1 meter. Pada gambar 4.4 bagian (b) ketika mendeteksi teks "titikB" terjadi kesalahan, program di hasilkan menghasilkan "-" karena tidak cocok dengan teks di depannya. Jika terjadi error seperti ini maka akan mengulang pembacaan sampai benar sehingga AGV tidak akan error reading saat mendeteksi lokasi dengan pembacaan teks. Selanjutnya, saya akan menyajikan pengenalan teks menggunakan OCR yang telah saya tingkatkan atau modifikasi.



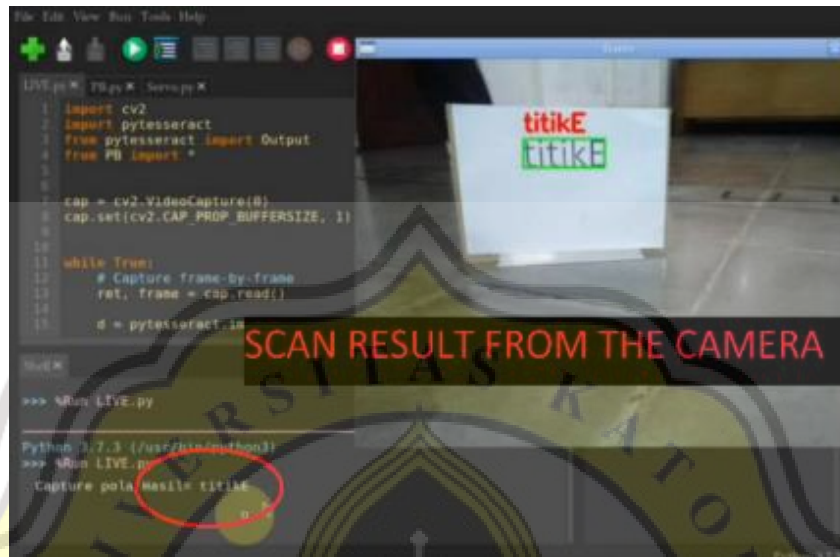
(a)



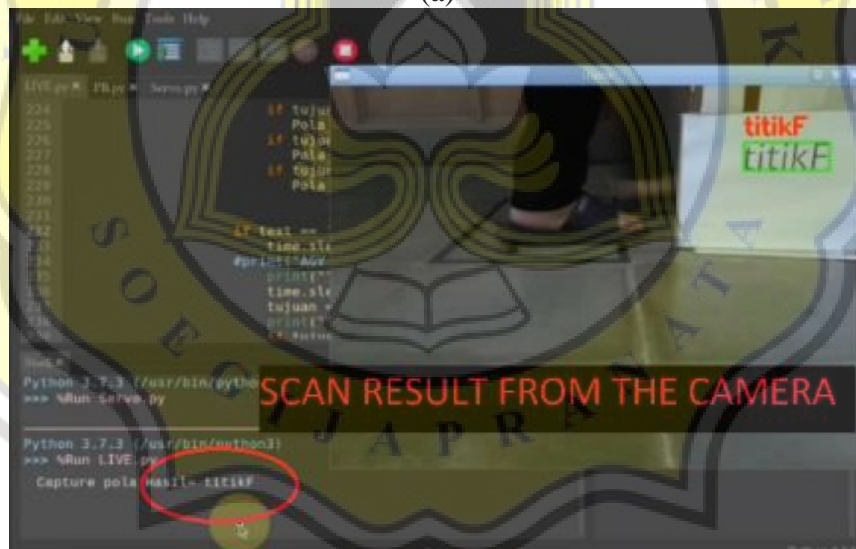
(b)

Gambar 4.5 Pembacaan OCR Pada Titik C dan D

Titik tersebut ditunjukkan pada Gambar 4.5 kamera robot AGV mendeteksi teks “TitikC” dan “TitikD” yang kemudian akan membaca dan mengolah gambar yang kemudian akan menghasilkan teks yang akan ditampilkan di monitor.



(a)



(b)

Gambar 4.6 Pembacaan OCR Pada Titik E dan F

Titik tersebut ditunjukkan pada Gambar 4.6 kamera robot AGV mendeteksi teks “TitikE” dan “TitikF” yang kemudian akan membaca dan mengolah gambar yang kemudian akan menghasilkan teks yang akan ditampilkan di monitor.

Table 1. Test Jarak Deteksi Teks

Jarak Pendeteksian Teks	Tingkat Keberhasilan
0 meter – 0.5 meter	95 %
0.5 meter- 1 meter	95 %
1 meter – 1.5 meter	95 %
1.5 meter – 2 meter	95 %
> 2 meter	Tergantung dari Kondisi cahaya dan ukuran teks

Tabel 1 menunjukkan tingkat keberhasilan membaca teks berdasarkan jarak. Pembacaan berdasarkan tabel di atas memiliki tingkat keberhasilan 95% jika dalam jarak kurang dari 2 meter. jika membaca teks dari jarak lebih dari 2 meter ini bisa berhasil tetapi harus memenuhi beberapa syarat yaitu pencahayaan di dalam ruangan, ukuran teks yang disarankan lebih luas, dan warna dalam ruangan juga mempengaruhi pengenalan teks.