

BABIV

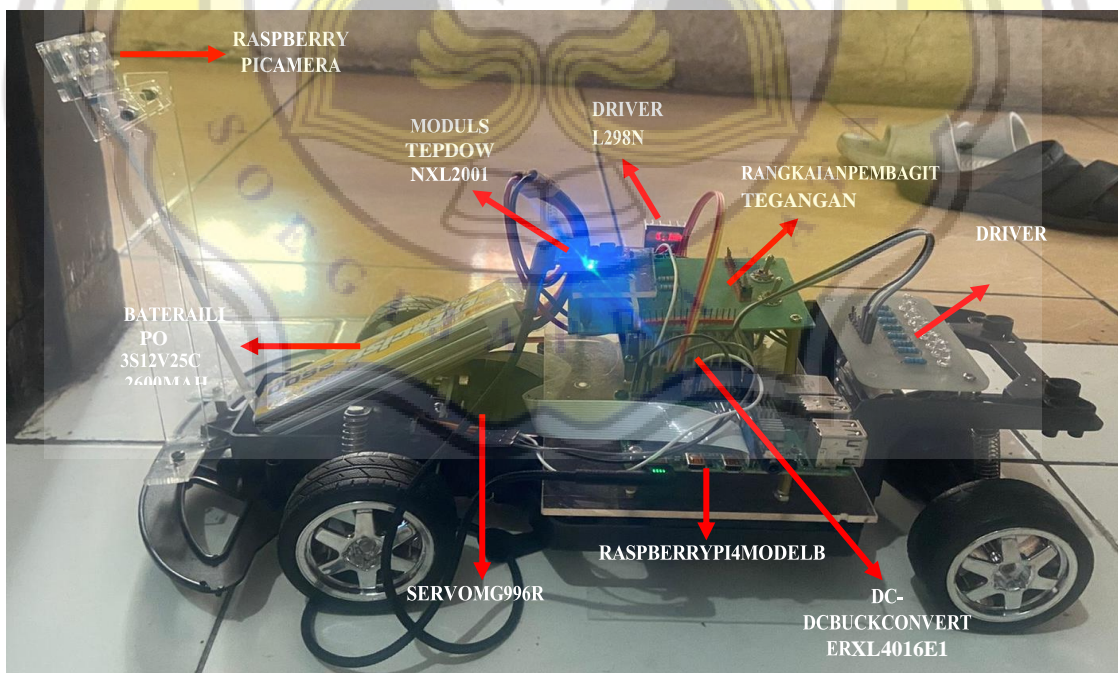
HASIL DAN PEMBAHASAN

4.1. Pendahuluan

Bab ini akan menjelaskan hasil dari prototipe robot AGV, program, dan pengujian alat yang dilakukan untuk studi pembacaan teks AGV. Hasil pengujian dan diskusi selanjutnya akan memberikan hasil dari alat prototipe, program yang digunakan, dan perbandingan antara sistem OCR Default dan OCR yang dimodifikasi atau ditingkatkan.

4.2. Prototype Alat

Menghasilkan bentuk perangkat keras yang akan digunakan dalam Tugas Akhir sebagai bagian dari penelitian ini. Hasil dari perangkat keras ini konsisten dengan desain alat Bab III. Gambar 4.1 menggambarkan perangkat keras AGV.



Gambar4.1Hardware AGV

4.3. 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
```

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.

```
13 while True:
14     # Capture frame-by-frame
15     ret, frame = cap.read()
16
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 pola! Hasil',text )
27                 key=cv2.waitKey(1)
28                 #key = cv2.waitKey(0)
29                 if key == ord('q'):
```

Seperti terlihat pada gambar di atas, ini merupakan bagian dari program utama membaca teks yang akan diimplementasikan dalam AGV ini. Program memiliki

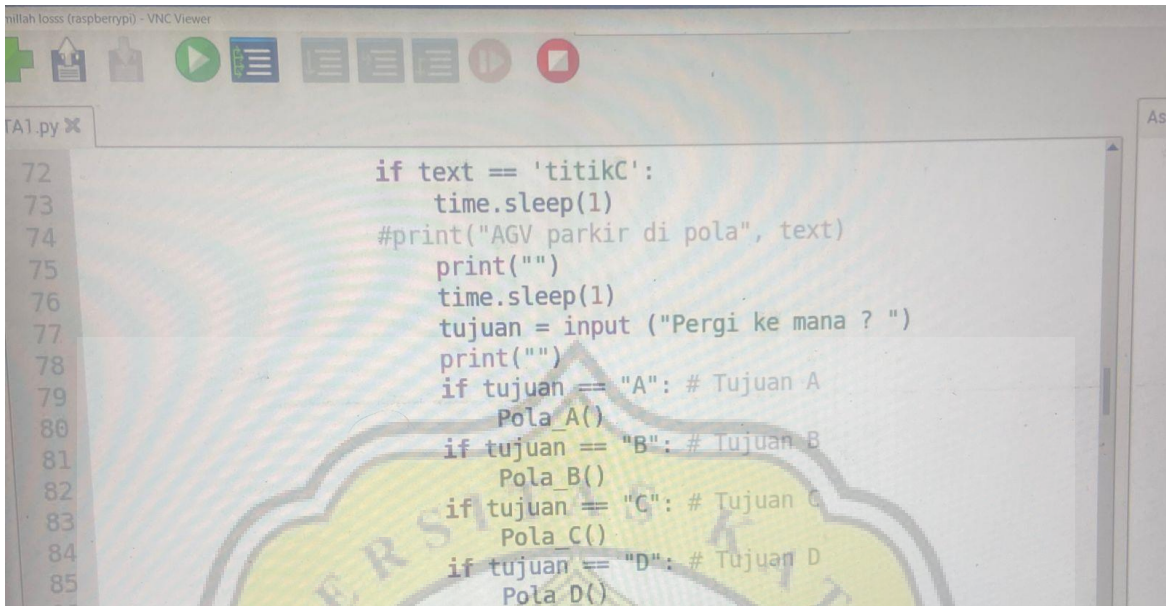
beberapa pengolahan dengan menambahkan tampilan hasil pembacaan yang akan ditampilkan pada monitor. Agar pembacaan prediksi ditampilkan langsung di monitor.

```
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
```

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.

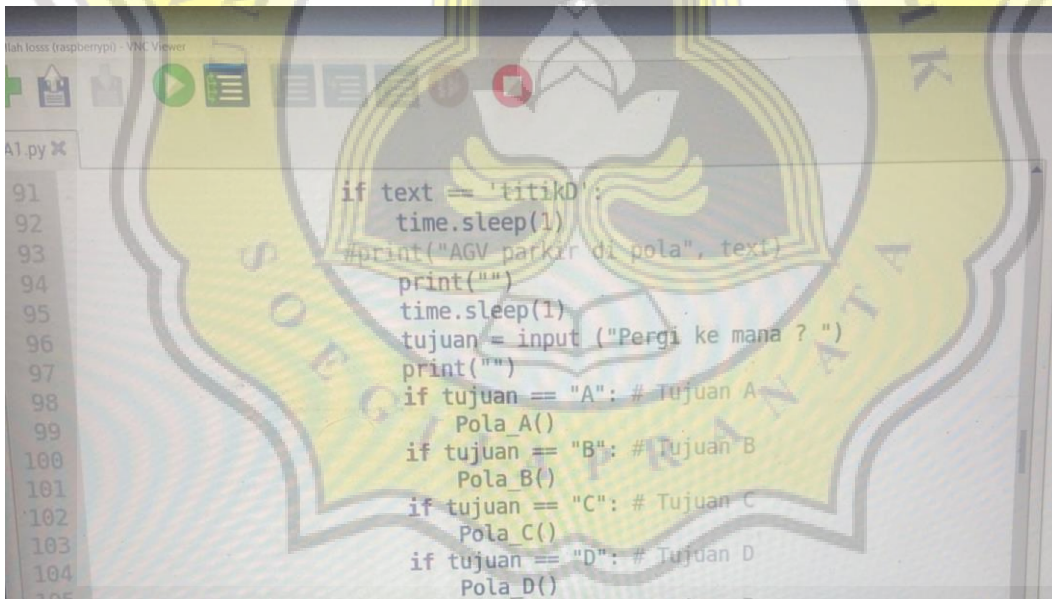
```
bismillah loss (raspberrypi): VNC Viewer
TA1.py x
53         if text == 'titikB':
54             time.sleep(1)
55             #print("AGV parkir di pola", text)
56             print("")
57             time.sleep(1)
58             tujuan = input ("Pergi ke mana ? ")
59             print("")
60             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
```

Program di atas adalah program deteksi titik B dan tujuan ketitik C



```
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()
```

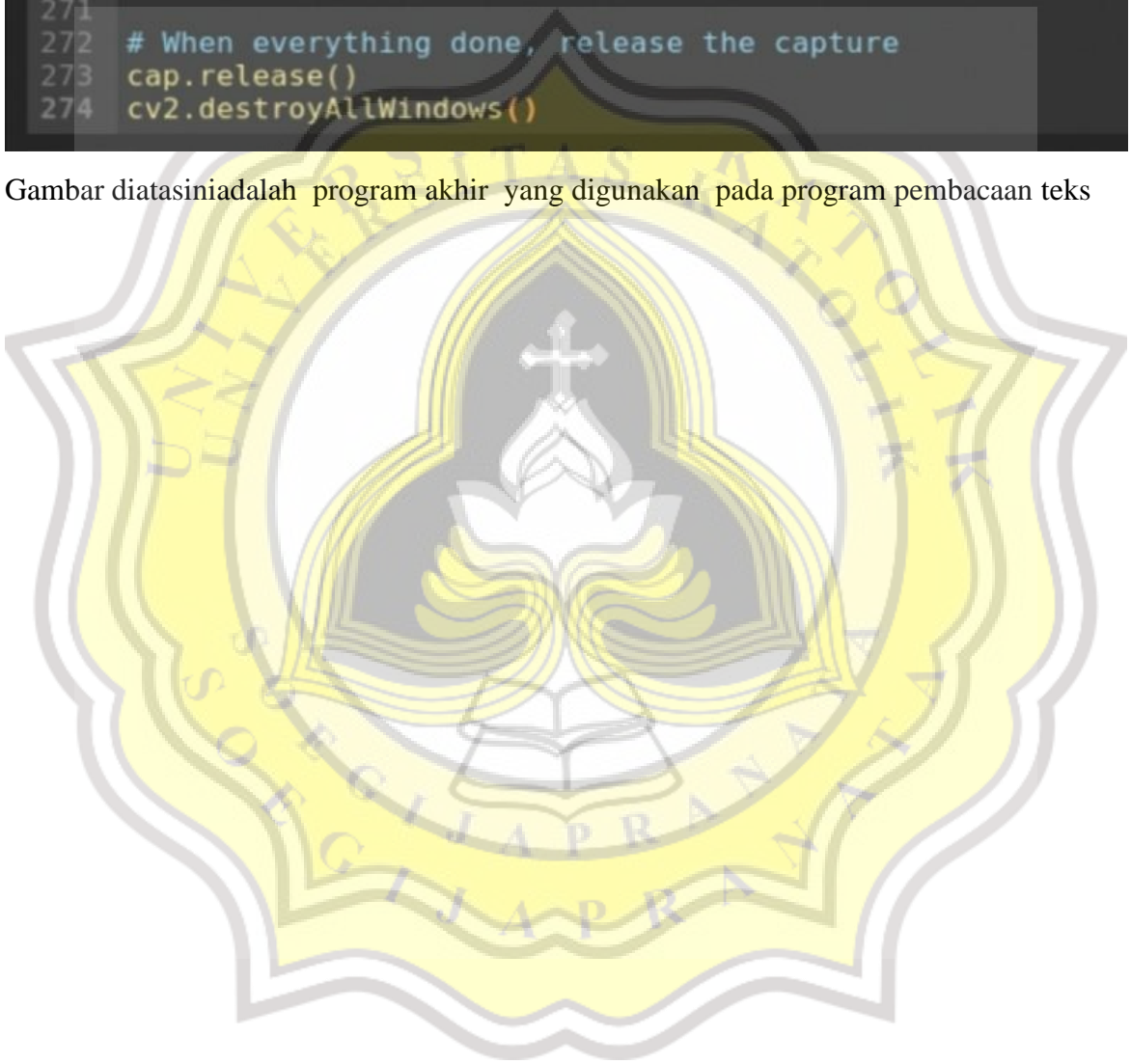
Program di atas adalah program deteksi titik C dan tujuan ketitik D



```
91         if text == 'titikD':
92             time.sleep(1)
93             #print("AGV parkir di pola", text)
94             print("")
95             time.sleep(1)
96             tujuan = input ("Pergi ke mana ? ")
97             print("")
98             if tujuan == "A": # Tujuan A
99                 Pola_A()
100            if tujuan == "B": # Tujuan B
101                Pola_B()
102            if tujuan == "C": # Tujuan C
103                Pola_C()
104            if tujuan == "D": # Tujuan D
105                Pola_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 di atas ini adalah program akhir yang digunakan pada program pembacaan teks



4.4. Perbandingan Default OCR dan Enhanced OCR

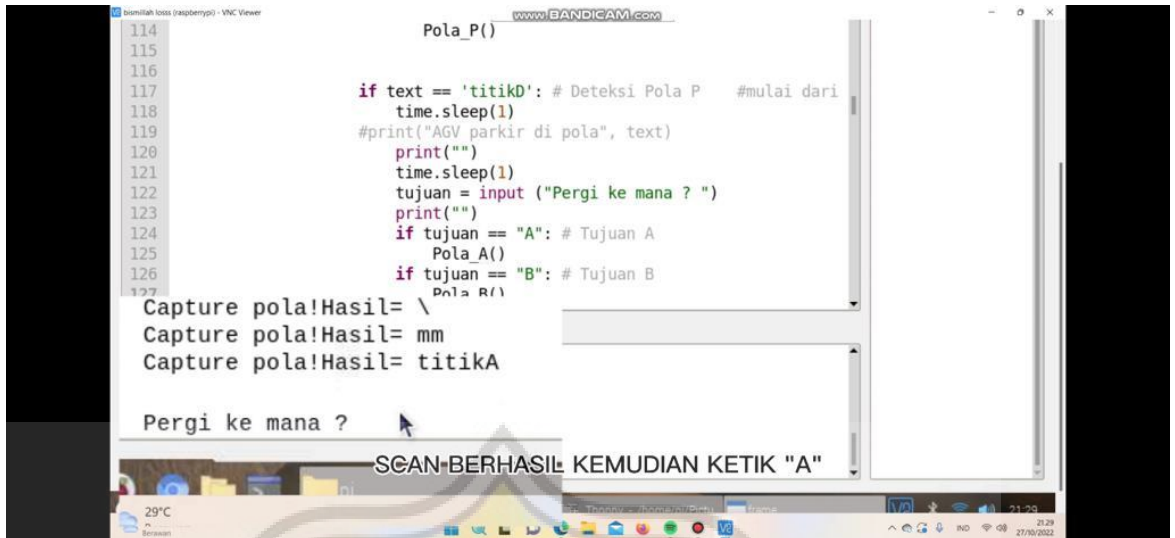
Dalam pembicaraan ini, saya akan membandingkan pembacaan teks dengan mode mesin operasi OCR (OEM) default dan mode segmentasi halaman (PSM) dengan OCR yang saya ubah untuk mengurangi kesalahan pembacaan. Dalam perbandingan ini, saya menggunakan satu teks yang membaca "pointA" dan "pointB" pada OCR default saya menggunakan pengaturan OEM-3 dan PSM-7 yang dapat membaca satu teks.



(a)

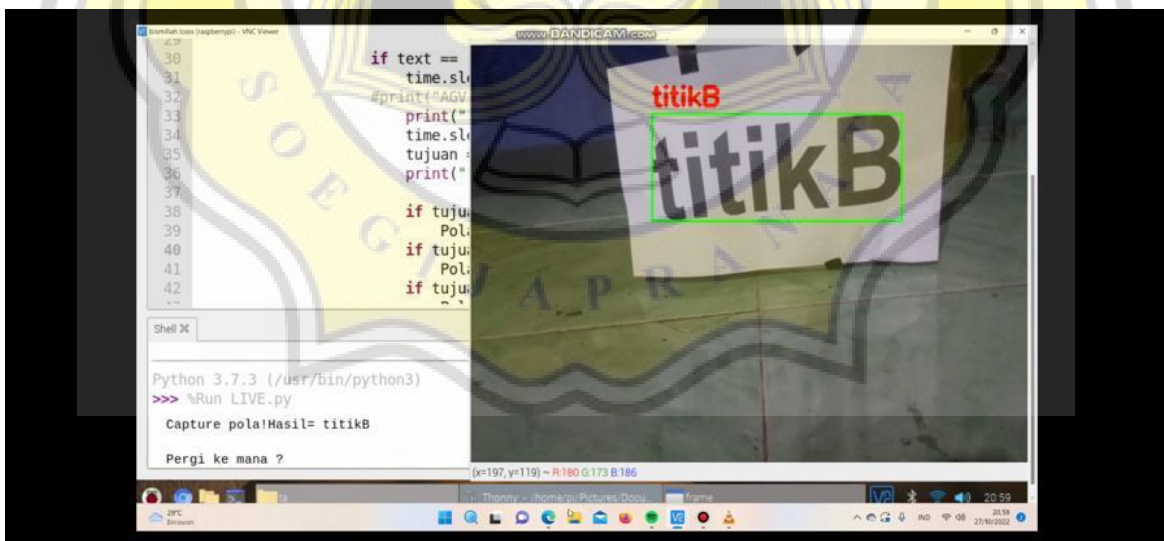
Gambar4.2 Pembacaan Teks dengan Default OCR

Seperti terlihat pada Gambar 8, pembacaan teks pada bagian (a) diambil dari jarak jauh dan menghasilkan kata-kata yang salah, sedangkan pada bagian (b), teks diperoleh dari jarak yang sangat dekat dan menghasilkan kata-kata yang sesuai dengan teks yang diambil.

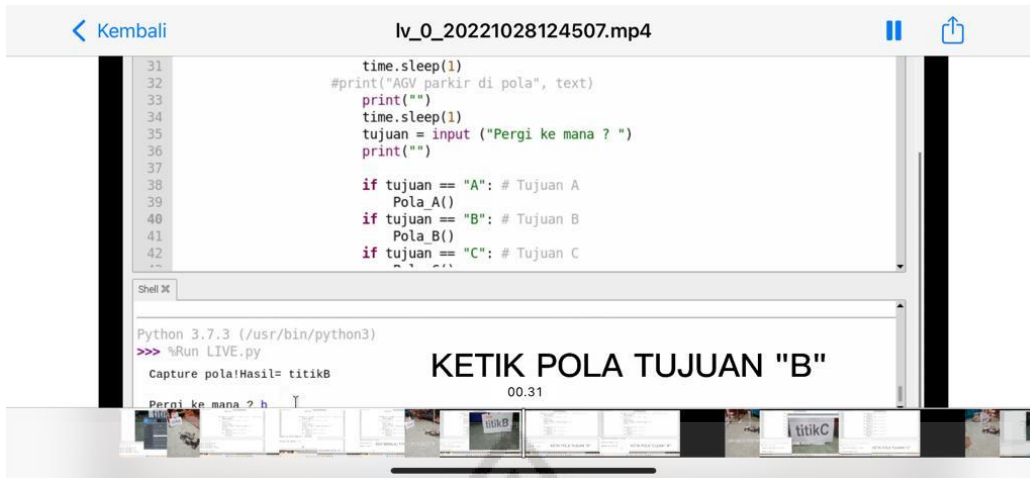


Gambar4.3 Pembacaan dengan OCR yang disempurnakan

OCR yang ditingkatkan ini menghasilkan pembacaan teks yang akurat dan dapat digunakan dari jarak sekitar 1 hingga 2 meter. OCR yang ditingkatkan ini mengurangi kesalahan membaca hingga 95%. Jika kata yang dibaca tidak sesuai dengan teks, kata tersebut tidak akan muncul di layar keluaran kamera.



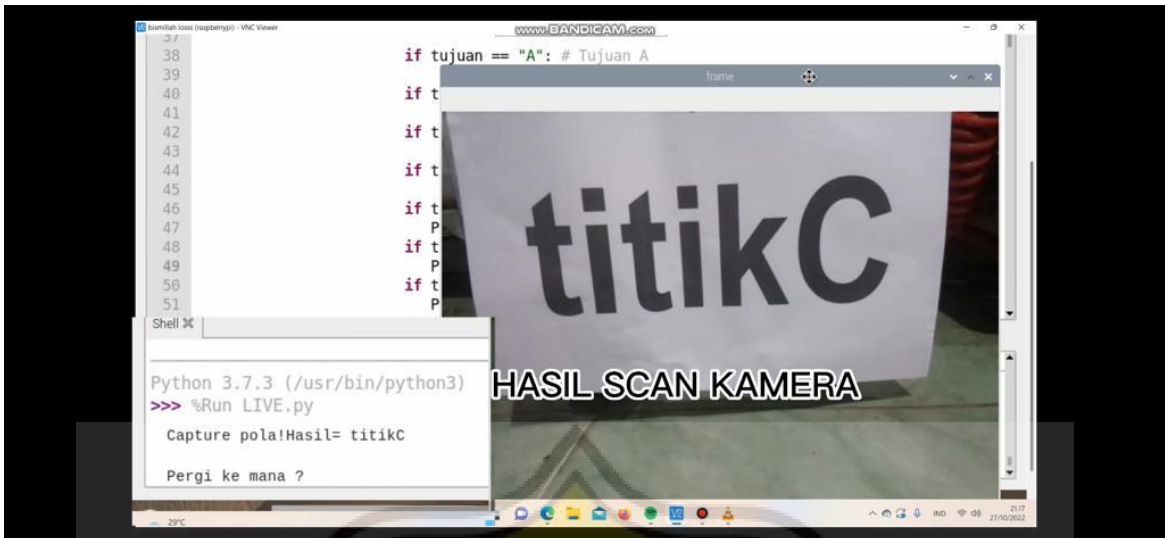
(a)



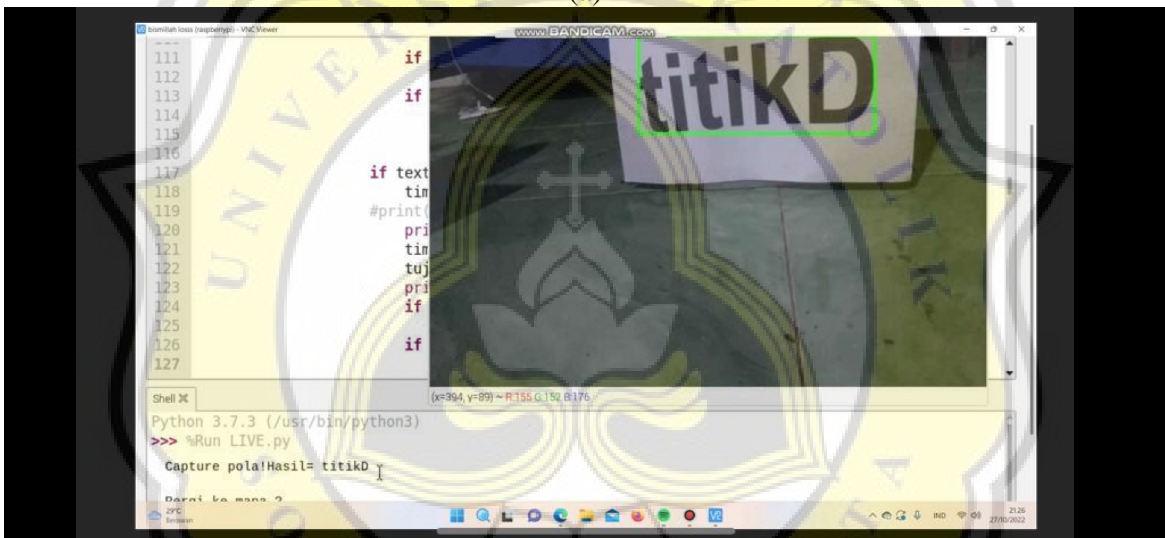
(b)

Gambar 4.4 Perbandingan OCR yang Disempurnakan Dengan OCR Bawaan

Bagian (a) dari Gambar 4.4 menggambarkan deteksi teks menggunakan OCR default, di mana hasilnya tidak dapat menghasilkan kata-kata yang sesuai dengan yang diperoleh kamera. Pembacaan teks pada bagian (a) salah karena jarak deteksi teks sangat jauh sehingga menimbulkan masalah pembacaan. Menggunakan OCR yang ditingkatkan dan membuat pembacaan yang sesuai yang dapat diidentifikasi dalam jarak kira-kira 1 meter diilustrasikan pada bagian (c). Pada Gambar 4.4, bagian (b), Ketika teks "titik B" terdeteksi, terjadi kesalahan, dan program berikut mengeluarkan "-" karena tidak cocok dengan teks sebelumnya. Jika kesalahan tersebut terjadi, pembacaan akan diulang sampai benar, mencegah AGV melakukan kesalahan pembacaan saat mendeteksi lokasi dengan membaca teks. Selanjutnya, saya akan menyajikan pengenalan teks berbasis OCR yang telah saya tingkatkan atau modifikasi.



(a)



(b)

Gambar4.5 Pembacaan OCR Pada Titik C dan D

Gambar 4.5 menggambarkan titik di mana kamera robot AGV mengenali kata "TitikC" dan "TitikD", yang kemudian dibaca dan diproses untuk menghasilkan teks yang 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 menampilkan persentase keberhasilan pembacaan teks berdasarkan jarak. Persentase keberhasilan pembacaan berdasarkan tabel di atas adalah 95% jika jaraknya kurang dari 2 meter. Dimungkinkan untuk membaca teks dari jarak lebih dari 2 meter. Namun, banyak keadaan harus dipenuhi, termasuk pencahayaan ruangan yang memadai, ukuran teks yang lebih besar dari yang direkomendasikan, dan warna dalam ruangan yang memengaruhi pengenalan teks.