

## LAMPIRAN

### Program Utama

```
import cv2
import pytesseract
from pytesseract import Output
from PB import *

cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_BUFFERSIZE, 1)

while True:
    # Capture frame-by-frame
    ret, frame = cap.read()

    d = pytesseract.image_to_data(frame, output_type=Output.DICT)
    n_boxes = len(d['text'])
    for i in range(n_boxes):
        if int(d['conf'][i]) > 60:
            (text, x, y, w, h) = (d['text'][i], d['left'][i], d['top'][i], d['width'][i], d['height'][i])
            # don't show empty text
            if text and text.strip() != "":
                frame = cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
                frame = cv2.putText(frame, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0, 255), 3)
                #print('capture pola!Hasil',text )
                key=cv2.waitKey(1)
                #key = cv2.waitKey(0)
                if key == ord('q'):
                    print("Capture pola!Hasil=",text )

                if text == 'titikA': # Deteksi Pola A
                    time.sleep(1)
                    #print("AGV parkir di pola", text)
                    print("")
                    time.sleep(1)
                    tujuan = input ("Pergi ke mana ? ")
                    print("")

                    if tujuan == "A": # Tujuan A
                        Pola_A()
                    if tujuan == "B": # Tujuan B
                        Pola_B()
                    if tujuan == "C": # Tujuan C
                        Pola_C()
                    if tujuan == "D": # Tujuan D
                        Pola_D()
                    if tujuan == "E": # Tujuan E
                        Pola_E()
                    if tujuan == "A2": # Tujuan A
                        Pola_A2()
                    if tujuan == "F": # Tujuan F
```

```

    Pola_F()
    if tujuan == "G": # Tujuan G
        Pola_G()
    if tujuan == "H": # Tujuan H
        Pola_H()
    if tujuan == "P": # Tujuan P
        Pola_P()

if text == 'titikB': # Deteksi Pola B
    time.sleep(1)
    #print("AGV parkir di pola", text)
    print("")
    time.sleep(1)
    tujuan = input ("Pergi ke mana ? ")
    print("")
    if tujuan == "A": # Tujuan A
        Pola_A()
    if tujuan == "B": # Tujuan B
        Pola_B()
    if tujuan == "C": # Tujuan C
        Pola_C()
    if tujuan == "D": # Tujuan D
        Pola_D()
    if tujuan == "E": # Tujuan E
        Pola_E()
    if tujuan == "A2": # Tujuan A
        Pola_A2()
    if tujuan == "F": # Tujuan F
        Pola_F()
    if tujuan == "G": # Tujuan G
        Pola_G()
    if tujuan == "H": # Tujuan H
        Pola_H()
    if tujuan == "P": # Tujuan P
        Pola_P()

if text == 'titikC': # Deteksi Pola C
    time.sleep(1)
    #print("AGV parkir di pola", text)
    print("")
    time.sleep(1)
    tujuan = input ("Pergi ke mana ? ")
    print("")
    if tujuan == "A": # Tujuan A
        Pola_A()
    if tujuan == "B": # Tujuan B
        Pola_B()
    if tujuan == "C": # Tujuan C
        Pola_C()
    if tujuan == "D": # Tujuan D
        Pola_D()
    if tujuan == "E": # Tujuan E
        Pola_E()

```

```

if tujuan == "A2": # Tujuan A
    Pola_A2()
if tujuan == "F": # Tujuan F
    Pola_F()
if tujuan == "G": # Tujuan G
    Pola_G()
if tujuan == "H": # Tujuan H
    Pola_H()
if tujuan == "P": # Tujuan P
    Pola_P()

if text == 'titikD': # Deteksi Pola D
    time.sleep(1)
#print("AGV parkir di pola", text)
print("")
time.sleep(1)
tujuan = input ("Pergi ke mana ? ")
print("")
if tujuan == "A": # Tujuan A
    Pola_A()
if tujuan == "B": # Tujuan B
    Pola_B()
if tujuan == "C": # Tujuan C
    Pola_C()
if tujuan == "D": # Tujuan D
    Pola_D()
if tujuan == "E": # Tujuan E
    Pola_E()
if tujuan == "A2": # Tujuan A
    Pola_A2()
if tujuan == "F": # Tujuan F
    Pola_F()
if tujuan == "G": # Tujuan G
    Pola_G()
if tujuan == "H": # Tujuan H
    Pola_H()
if tujuan == "P": # Tujuan P
    Pola_P()

if text == 'titikE': # Deteksi Pola E
    time.sleep(1)
#print("AGV parkir di pola", text)
print("")
time.sleep(1)
tujuan = input ("Pergi ke mana ? ")
print("")
if tujuan == "A": # Tujuan A
    Pola_A()
if tujuan == "B": # Tujuan B
    Pola_B()
if tujuan == "C": # Tujuan C
    Pola_C()
if tujuan == "D": # Tujuan D
    Pola_D()

```

```

if tujuan == "E": # Tujuan E
    Pola_E()
if tujuan == "A2": # Tujuan A
    Pola_A2()
if tujuan == "F": # Tujuan F
    Pola_F()
if tujuan == "G": # Tujuan G
    Pola_G()
if tujuan == "H": # Tujuan H
    Pola_H()
if tujuan == "P": # Tujuan P
    Pola_P()

if text == 'titikF': # Deteksi Pola F
    time.sleep(1)
#print("AGV parkir di pola", text)
print("")
time.sleep(1)
tujuan = input ("Pergi ke mana ? ")
print("")
if tujuan == "A": # Tujuan A
    Pola_A()
if tujuan == "B": # Tujuan B
    Pola_B()
if tujuan == "C": # Tujuan C
    Pola_C()
if tujuan == "D": # Tujuan D
    Pola_D()
if tujuan == "E": # Tujuan E
    Pola_E()
if tujuan == "A2": # Tujuan A
    Pola_A2()
if tujuan == "F": # Tujuan F
    Pola_F()
if tujuan == "G": # Tujuan G
    Pola_G()
if tujuan == "H": # Tujuan H
    Pola_H()
if tujuan == "P": # Tujuan P
    Pola_P()

if text == 'titikG': # Deteksi Pola G
    time.sleep(1)
#print("AGV parkir di pola", text)
print("")
time.sleep(1)
tujuan = input ("Pergi ke mana ? ")
print("")
if tujuan == "A": # Tujuan A
    Pola_A()
if tujuan == "B": # Tujuan B
    Pola_B()
if tujuan == "C": # Tujuan C
    Pola_C()

```

```

if tujuan == "D": # Tujuan D
    Pola_D()
if tujuan == "E": # Tujuan E
    Pola_E()
if tujuan == "A2": # Tujuan A
    Pola_A2()
if tujuan == "F": # Tujuan F
    Pola_F()
if tujuan == "G": # Tujuan G
    Pola_G()
if tujuan == "H": # Tujuan H
    Pola_H()
if tujuan == "P": # Tujuan P
    Pola_P()

if text == 'titikP': # Deteksi Pola P
    time.sleep(1)
    #print("AGV parkir di pola", text)
    print("")
    time.sleep(1)
    tujuan = input ("Pergi ke mana ? ")
    print("")
    if tujuan == "A": # Tujuan A
        Pola_A()
    if tujuan == "B": # Tujuan B
        Pola_B()
    if tujuan == "C": # Tujuan C
        Pola_C()
    if tujuan == "D": # Tujuan D
        Pola_D()
    if tujuan == "E": # Tujuan E
        Pola_E()
    if tujuan == "A2": # Tujuan A
        Pola_A2()
    if tujuan == "F": # Tujuan F
        Pola_F()
    if tujuan == "G": # Tujuan G
        Pola_G()
    if tujuan == "H": # Tujuan H
        Pola_H()
    if tujuan == "P": # Tujuan P
        Pola_P()

# Display the resulting frame
cv2.imshow('frame', frame)
cv2.waitKey(1)
#key = cv2.waitKey(0)
#key=cv2.waitKey(1)
#key = cv2.waitKey(0)
#if key == ord('q'):
    # print("Capture pola!Hasil=",text )
# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()

```



## Program Motor DC dan Motor Servo

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

#=====
#      mulai dari titik P
#=====
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("")
    print("")
#####
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("")
print("")

```

```
#####
```

```

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("")
    print("")

```



```
#####
```

```
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("")  
    print("")
```

```
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("")
print("")

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("")
print("")

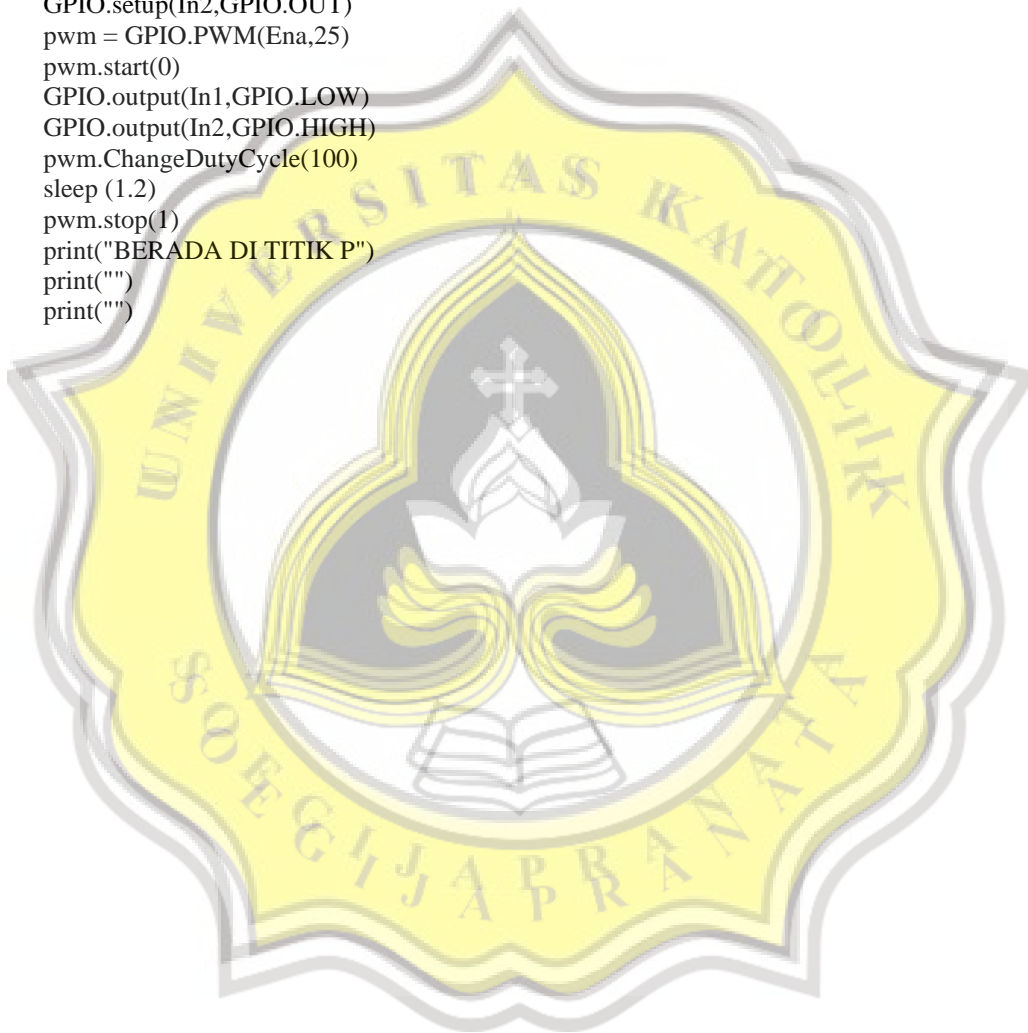
def Pola_G():
    GPIO.setup(25,GPIO.OUT)
    pwm=GPIO.PWM(25, 360)
    pwm.start(0)
    pwm.ChangeDutyCycle(42)
    sleep(1)
    pwm.stop(1)
    GPIO.setup(Ena,GPIO.OUT)
    GPIO.setup(In1,GPIO.OUT)
    GPIO.setup(In2,GPIO.OUT)
    pwm = GPIO.PWM(Ena,25)
    pwm.start(0)
    GPIO.output(In1,GPIO.LOW)
    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("")
    print("")

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("")
print("")
```





ABOUT THE  
JOURNAL

HOME ABOUT USER HOME CATEGORIES SEARCH CURRENT ARCHIVES  
ANNOUNCEMENTS CFP



#25863 Review

SUMMARY REVIEW EDITING

### Submission

Authors Florentinus Budi Setiawan, Yosia Yovie Christian Wibowo, Leonardus Heru Pratomo, Slamet Riyadi  
Title Perancangan Automated Guided Vehicle Menggunakan Penggerak Motor DC dan Motor Servo Berbasis Raspberry Pi 4  
Section Electrical  
Editor Fitri Arnia

### Peer Review

Round 1  
Review Version 25863-25863-1-REV.PDF 2022-05-09  
Initiated 2022-06-20  
Last modified 2022-06-24  
Uploaded file None

### Editor Decision

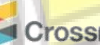
Decision Accept Submission 2022-07-05  
Notify Editor Editor/Author Email Record 2022-07-06  
Editor Version 25863-25863-1-ED.PDF 2022-07-05  
Author Version 25863-25863-1-REV.PDF 2022-06-30 DELETE  
25863-25863-2-ED.PDF 2022-06-30 DELETE  
25863-25863-2-REV.PDF 2022-07-06 DELETE  
Upload Author Version  Tidak ada file yang diunggah

293175  
[View My Stats](#)



Jurnal Rekayasa Elektrika (JRE) is published under license of [Creative Commons Attribution-ShareAlike 4.0 International License](#).

INDEXED BY



LANGUAGE

PAPER NAME  
**18.F1.0018.docx**

WORD COUNT  
**4595 Words**

CHARACTER COUNT  
**26765 Characters**

PAGE COUNT  
**41 Pages**

FILE SIZE  
**74.9KB**

SUBMISSION DATE  
**Jul 7, 2022 12:04 PM GMT+7**

REPORT DATE  
**Jul 7, 2022 12:05 PM GMT+7**

● **11% Overall Similarity**

The combined total of all matches, including overlapping sources, for each database.

- 11% Internet database
- 1% Publications database
- Crossref database
- Crossref Posted Content database
- 6% Submitted Works database



Summary