

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("")

```

1. Tahapan cara Install Python versi terbaru di Raspberry

Hal pertama yang dapat dilakukan adalah mengecek apakah Python default Anda ada pada Python 3.x atau pada Python 2.x, untuk memastikan hal ini gunakan command sebagai beriku pada bagian terminal :

```
$ python --version
```

Beberapa hal perlu dilakukan untuk mengubah Python 2.7 menjadi Python 3.6. Hal pertama yang perlu diperhatikan adalah memastikan bahwa beberapa package dependencies sudah di-install pada Raspberry Pi.

Pastikan dengan menggunakan command sebagai berikut:

```
# apt-get update
```

```
# apt-get install build-essential tk-dev libncurses5-dev libncursesw5-dev libreadline6-dev libdb5.3-dev
libgdbm-dev libsqlite3-dev libssl-dev libbz2-dev libexpat1-dev liblzma-dev zlib1g-dev
```

Setelah melaksanakan beberapa program tersebut, Anda dapat langsung mengunduh source code Python 3.6 dari website resmi Python. Source code Python 3.6 yang tersedia berupa tar.xz sehingga anda perlu

melakukan proses unpack. Sebelum mengunduh dengan wget, pastikan Anda sudah di direktori /tmp/ agar source code yang Anda unduh akan terhapus secara otomatis setelah reboot.

```
$ cd /tmp/
```

```
$ wget https://www.python.org/ftp/python/3.6.4/Python-3.6.4.tar.xz
```

```
$ tar xf Python-3.6.4.tar.xz
```

Setelah Anda berhasil mengekstrak seperti pada Gambar 2, Anda dapat melihat bahwa source code Python 3.6.4 sudah berada pada /tmp/Python-3.6.4. Untuk melanjutkan proses instalasi Python 3.6.4, akses direktori Python-3.6.4

```
$ cd Python-3.6.4
```

Setelah membuka direktori, ikuti proses ini untuk instalasi Python 3.6.4:

```
$ ./configure
```

```
$ make
```

```
# make altinstall
```

Setelah instalasi Python 3.6, Anda akan melihat bahwa Python default masih seperti Python sebelumnya, untuk saya Python masih 2.7.13. Untuk mengubah default Python yang digunakan perlu digunakan update-alternatives. Pertama, pastikan Anda tahu posisi instalasi Python 3.6 dan bahwa Python 3.6 yang sudah diinstal dapat diakses. Lakukan kedua command ini untuk mengetahui posisi Python 3.6

```
$ ls /usr/bin/python*
```

```
$ ls /usr/local/bin/python*
```

Setelah mengetahui posisi executable Python 3.6, lakukan update-alternatives untuk menambahkan Python 2.7.13 dan Python 3.6.4 sebagai alternatif Python. Gambar 3 menunjukkan hasil keluaran dari command untuk mengetahui posisi executable dari Python2.7, Python3.5 dan Python3.6. Tambahkan semua executable Python ini sebagai alternatif dari Python. Setelah mengetahui posisi yakni pada /usr/bin/ untuk Python2.7 dan Python3.5 dan pada /usr/local/bin/ pada Python3.6 yang baru diinstal. Lakukan proses penambahan alternatif /usr/bin/python dengan menggunakan command sebagai berikut:

```
# update-alternatives --install /usr/bin/python python /usr/bin/python2.7 1
# update-alternatives --install /usr/bin/python python /usr/bin/python3.5 2
# update-alternatives --install /usr/bin/python python /usr/local/bin/python3.6 3
```

Setelah menambahkan sesuai urutan di atas, Python3.6 menjadi mode otomatis dari python.

Berikut ini adalah proses instalasi OpenCV yang akan digunakan untuk implementasi robot ini

Masukan perintah dibawah ini satu persatu hingga selesai.

```
sudo apt install cmake build-essential pkg-config git
sudo apt install libjpeg-dev libtiff-dev libjasper-dev libpng-dev libwebp-dev libopenexr-dev
sudo apt install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev libxvidcore-dev libx264-dev
libdc1394-22-dev libgstreamer-plugins-base1.0-dev libgstreamer1.0-dev
sudo apt install libgtk-3-dev libqtgui4 libqtwebkit4 libqt4-test python3-pyqt5
sudo apt install libatlas-base-dev liblapacke-dev gfortran
sudo apt install libhdf5-dev libhdf5-103
sudo apt install python3-dev python3-pip python3-numpy
```

Untuk langkah selanjutnya kita harus memperbesar 'swap file' di memory card Raspi untuk memudahkan proses instalasi dari 100 menjadi 2048

```
sudo nano /etc/dphys-swapfile
```

```
CONF_SWAPSIZE=2048
```

untuk menyimpan settingan di atas tekan CTRL-X, pilih Y (yes) kemudian enter. Jangan lupa restart untuk memastikan swap file sudah berubah.

```
sudo systemctl restart dphys-swapfile
```

Setelah itu masukkan perintah berikut satu persatu.

```
git clone https://github.com/opencv/opencv.git
git clone https://github.com/opencv/opencv_contrib.git
mkdir ~/opencv/build
cd ~/opencv/build
cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
```

```
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \  
  
-D ENABLE_NEON=ON \  
  
-D ENABLE_VFPV3=ON \  
  
-D BUILD_TESTS=OFF \  
  
-D INSTALL_PYTHON_EXAMPLES=OFF \  
  
-D OPENCV_ENABLE_NONFREE=ON \  
  
-D CMAKE_SHARED_LINKER_FLAGS=-latomic \  
  
-D BUILD_EXAMPLES=OFF .  
  
make -j$(nproc)  
sudo make install  
sudo ldconfig
```

Proses instalasi paket OpenCV sudah selesai. Anda dapat mengecek apakah instalasi berhasil dengan perintah sbb:

```
python  
import cv2  
cv2.__version__
```

biasanya pada project computer vision dengan OpenCV membutuhkan library tambahan yakni 'opencv-contrib-python'.

```
pip install opencv-contrib-python  
sudo nano /etc/dphys-swapfile  
CONF_SWAPSIZE=100  
sudo systemctl restart dphys-swapfile
```

Berikut adalah tata cara install Library Tesseract untuk pengenalan teks dengan cara memasukan perintah pada Terminal window untuk konfigurasi paket debian

```
sudo dpkg --configure -a
```

Setelah melakukan perintah diatas kita bisa melanjutkan dengan perintah berikutnya untuk menginstal tesseract dengan memasukkan perintah berikut ini.

```
sudo apt-get install tesseract-ocr
```

Sekarang setelah Tesseract OCR terinstal, kita harus menginstal paket PyTesseract menggunakan paket pip install. Pytesseract adalah pembungkus python di sekitar mesin OCR tesseract, yang membantu kita menggunakan tesseract dengan python. Ikuti perintah di bawah ini untuk menginstal pytesseract di python.

Pip install pytesseract

Setelah semua proses ini semua selesai OCR dengan library Tesseract sudah siap digunakan.



PAPER NAME

TA-18.F1.0019.docx

WORD COUNT

4411 Words

CHARACTER COUNT

26765 Characters

PAGE COUNT

28 Pages

FILE SIZE

64.1KB

SUBMISSION DATE

Oct 12, 2022 11:28 AM GMT+7

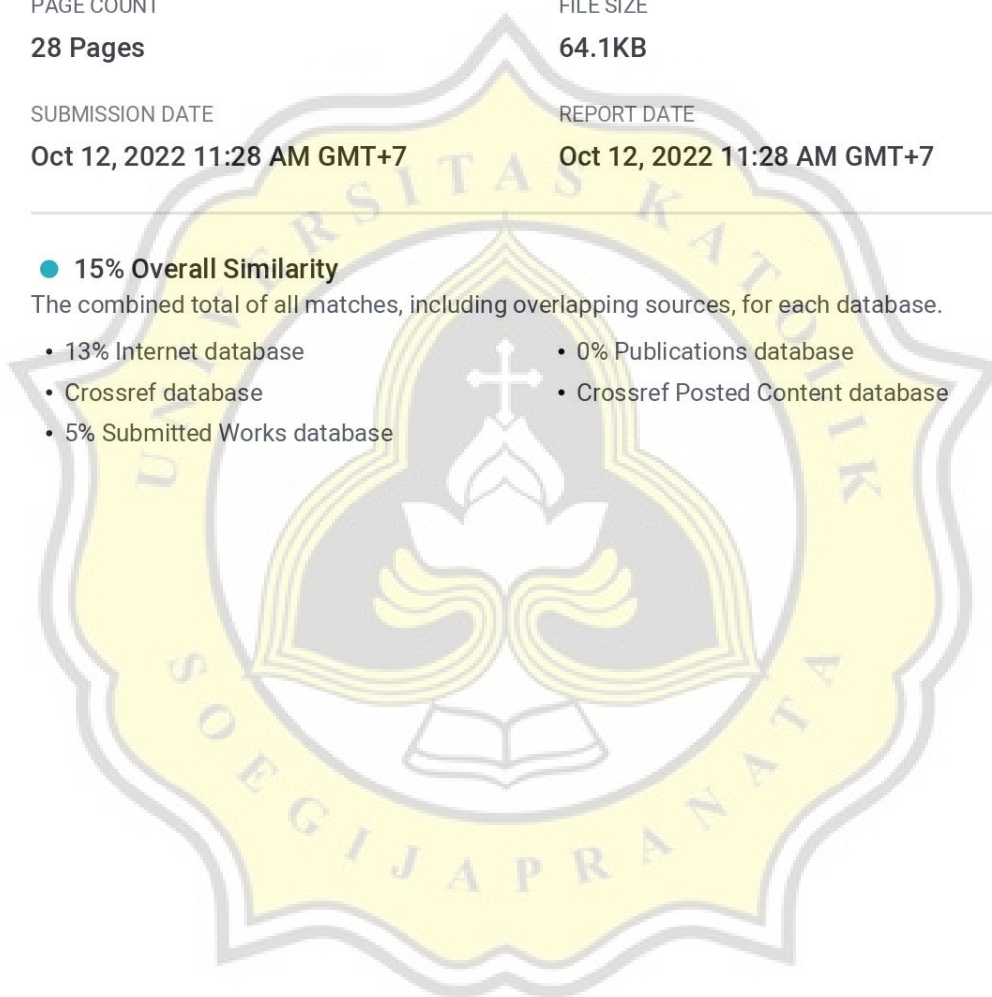
REPORT DATE

Oct 12, 2022 11:28 AM GMT+7

● **15% Overall Similarity**

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

- 13% Internet database
- 0% Publications database
- Crossref database
- Crossref Posted Content database
- 5% Submitted Works database



Summary