

1. *source Code* 01_face_dataset.py

```
import cv2
import os

cam = cv2.VideoCapture(0)
cam.set(3, 640) # Setting Lebar pixel Display
cam.set(4, 480) # Setting Tinggi pixel Display

face_detector =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# For each person, enter one numeric face id
face_id = input("\n Input nomor ID Face anda <Enter> ==> ')

print("\n [INFO] Inialisasi Face Anda. Lihat Kamera dan tunggu
beberapa saat...")
# inialisasi penghitung face sampling
count = 0

while(True):

    ret, img = cam.read()
    img = cv2.flip(img, 1) #flip tampilan
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_detector.detectMultiScale(gray, 1.3, 5)

    for (x,y,w,h) in faces:

        cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
        count += 1
```

```

# simpan hasil capture gambar ke path
cv2.imwrite("dataset1/User." + str(face_id) + '.' + str(count) + ".jpg",
gray[y:y+h,x:x+w])

cv2.imshow('image', img)

k = cv2.waitKey(100) & 0xff # Tekan 'ESC' untuk keluar
if k == 27:
    break
elif count >= 100: # ambil 30 sampling dan berhenti
    break

print("\n Ok Selesai, Terima kasih")
cam.release()
cv2.destroyAllWindows()

```

2. *Source Code* 02_face_training.py

```

import cv2
import numpy as np
from PIL import Image
import os

# Path untuk image database
path = 'dataset1'

recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");

# function to get the images and label data
def getImagesAndLabels(path):

```

```

imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
faceSamples=[]
ids = []

for imagePath in imagePaths:

    PIL_img = Image.open(imagePath).convert('L') # convert it to
    grayscale
    img_numpy = np.array(PIL_img,'uint8')

    id = int(os.path.split(imagePath)[-1].split(".")[1])
    faces = detector.detectMultiScale(img_numpy)

    for (x,y,w,h) in faces:
        faceSamples.append(img_numpy[y:y+h,x:x+w])
        ids.append(id)

return faceSamples,ids

print ("\n [INFO] Training faces. Harap Sabar Menunggu ^_^...")
faces,ids = getImagesAndLabels(path)
recognizer.train(faces, np.array(ids))

# Save the model into trainer/trainer.yml
recognizer.save('trainer/trainer1.yml') # recognizer.save() worked on Mac,
but not on Pi

# Print the numer of faces trained and end program
print("\n [INFO] {0} faces trained. Exiting
Program".format(len(np.unique(ids))))

```

3. *Source Code* 03_face_recognition.py

```
import RPi.GPIO as GPIO
import cv2
import numpy as np
import os
import serial

arduino = serial.Serial('/dev/ttyUSB0',9600,timeout = .1)
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer1.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);

font = cv2.FONT_HERSHEY_SIMPLEX

#inisialisasi id
id = 0
names = ['None', 'Najib', 'Harits', 'Dhadang', 'Z', 'W']

# Mulai realtime video capture
cam = cv2.VideoCapture(0)
cam.set(3, 640) # Setting Lebar pixel Display
cam.set(4, 480) # Setting Tinggi pixel Display

minW = 0.1*cam.get(3)
minH = 0.1*cam.get(4)

while True:

    ret, img =cam.read()
    img = cv2.flip(img, 1) # Flip vertically
```

```

gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
faces = faceCascade.detectMultiScale(
    gray,
    scaleFactor = 1.2,
    minNeighbors = 5,
    minSize = (int(minW), int(minH)),
)
for(x,y,w,h) in faces:
    cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
#    cv2.circle(img,(int(x+w/2),int(y+h/2)),20,(255,0,255),-1)
    id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
    if (w>= 200):
        if (100 - confidence > 40):
            id = names[id]
            confidence = " {0}%".format(round(100 - confidence))
        else:
            id = "unknown"
            confidence = " {0}%".format(round(100 - confidence))
        cv2.imwrite("capture/dicurigai.jpg", img)
#        os.system("curl -i -X POST -H \"Content-Type: multipart/form-
data\" -F \"filename=@capture/dicurigai.jpg\"
http://latekkom2019.com/Raspi/uploadFiles.php")
        cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
        if (id == "Najib"):
            arduino.write("Najib".encode())
            arduino.write('x'.encode())
        cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0),
1)
elif (w<=180):
    arduino.write(str(int(x+w/2)).encode())
    arduino.write('x'.encode())

```

```
cv2.imshow('camera',img)
k = cv2.waitKey(10) & 0xff # Press 'ESC' for exiting video
if k == 27:
    break
```

```
# Do a bit of cleanup
print("\n [INFO] Exiting Program and cleanup stuff")
cam.release()
cv2.destroyAllWindows()
```

4. *Source Code* rasbery_arduino.ino

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
Servo myservo;

String inputString = "";
bool stringComplete = false;
int x, arah;

void setup() {
    pinMode(8, OUTPUT);
    myservo.attach(9);
    myservo.write(110);
    lcd.begin();
    lcd.backlight();
    Serial.begin(9600);
    inputString.reserve(200);
```

```
lcd.print("Face Track and");  
lcd.setCursor(2, 1);  
lcd.print("recognition");  
digitalWrite(8, LOW);  
}
```

```
void loop() {  
  if (stringComplete) {  
    if (inputString == "Najib") {  
      lcd.clear();  
      digitalWrite(8, HIGH);  
      lcd.print("selamat datang");  
      lcd.setCursor(0, 1);  
      lcd.print(" Najib");  
      delay(6000);  
    }  
    else {  
      digitalWrite(8, LOW);  
      Serial.println(inputString);  
      x = inputString.toInt();  
      lcd.home();  
      lcd.print("wajah terdeteksi");  
      lcd.setCursor(0, 1);  
      lcd.print("x ");  
      lcd.print(x);  
      lcd.print(" ");  
      lcd.print("sudut ");  
      lcd.print(myservo.read());  
      lcd.print(" ");  
  
      if (x >= 500 && myservo.read() == 110) { // tengah ke kanan
```

```
for (int i = 110; i <= 120; i++) {  
    myservo.write(i);  
    delay(15);  
}  
inputString = "";  
stringComplete = false;  
}
```

```
else if (x <= 160 && myservo.read() == 110) { // tengah ke kiri  
    for (int i = 110; i >= 100; i--) {  
        myservo.write(i);  
        delay(15);  
    }  
    inputString = "";  
    stringComplete = false;  
}
```

```
else if (x >= 500 && myservo.read() == 100) {  
    for (int i = 100; i <= 110; i++) {  
        myservo.write(i);  
        delay(15);  
    }  
    inputString = "";  
    stringComplete = false;  
}
```

```
else if (x <= 160 && myservo.read() == 100) {  
    for (int i = 100; i >= 90; i--) {  
        myservo.write(i);  
        delay(15);  
    }  
    inputString = "";
```



```
    stringComplete = false;
}
else if (x >= 500 && myservo.read() == 90) {
    for (int i = 90; i <= 100; i++) {
        myservo.write(i);
        delay(15);
    }
    inputString = "";
    stringComplete = false;

}

else if (x >= 160 && myservo.read() == 120) {
    for (int i = 120; i <= 130; i++) {
        myservo.write(i);
        delay(15);
    }
    inputString = "";
    stringComplete = false;

}

else if (x <= 160 && myservo.read() == 120) {
    for (int i = 120; i >= 110; i--) {
        myservo.write(i);
        delay(15);
    }
    inputString = "";
    stringComplete = false;
}
```

```
else if (x <= 160 && myservo.read() == 130) {
  for (int i = 130; i >= 120; i--) {
    myservo.write(i);
    delay(15);
  }
  inputString = "";
  stringComplete = false;

}
inputString = "";
stringComplete = false;
}
}
```

```
void serialEvent() {
  while (Serial.available()) {
    char inChar = (char)Serial.read();
    if (inChar == 'x') {
      stringComplete = true;
    }
    else {
      inputString += inChar;
    }
  }
}}
```