

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] Inisialisasi Face Anda. LihatKameradantunggubeberapasaat...")
# inisialisasipenghitung 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
        # simpanhasil capture gambarke path
        cv2.imwrite("dataset/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' untukkeluar
    if k == 27:
        break
    elif count >= 100: # ambil100 sampling danberhenti
        break
print("\n Ok Selesai, Terimakasih")
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 = 'dataset'
recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");
# function to get the images and label data
def getImagesAndLabels(path):
    imagePath = [os.path.join(path, f) for f in os.listdir(path)]
    faceSamples = []
    ids = []
    for imagePath in imagePath:
        PIL_img = Image.open(imagePath).convert('L') # convert it to grayscale
        img_numpy = np.array(PIL_img, 'uint8')
        id = int(os.path.splitext(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.write('trainer/trainer.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 Gmail.py

```
import cv2
import numpy as np
import os
import Rpi.GPIO as gpio
gpio.setmode(gpio.BCM)
gpio.setup(4, gpio.OUT)
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);
font = cv2.FONT_HERSHEY_SIMPLEX
#inisialisasi id
id = 0
names = ['None', 'Harits', 'Bayu', 'Hamim', 'Pak Slamet', 'Pak Amin']
# Mulai realtime video capture
cam = cv2.VideoCapture(0)
cam.set(3, 640) # Setting Lebar pixel Display
cam.set(4, 480) # Setting Tinggi pixel Display
count = 0
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)
    id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
    # Check if confidence is less than 100 ==> "0" is perfect match
    if (confidence < 50):
        id = names[id]
        confidence = " {0}%".format(round(100 - confidence))
        if (count == 5):
            print("Dikenali", str(id))
print("Kemiripan", str(confidence))
print("Solenoid Terbuka")
    gpio.output(4, gpio.HIGH)
    time.sleep(2)
    gpio.output(4, gpio.LOW)
    time.sleep(2)
else:
    id = "unknown"
    confidence = " {0}%".format(round(100 - confidence))
    print("Tidak Dikenali")
    print("Solenoid Terkunci")
    gpio.output(4, gpio.LOW)
    time.sleep(2)
cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)
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()
```

