

Lampiran

```

@raspberrypi:~/facebookLoc $ nano face-encoding.py
@raspberrypi:~/facebookLoc $ python3 face-encoding.py --dataset dataset --encodings encodings.pickle --detection-method hog
[INFO] mendapatkan model wajah...
[INFO] Memproses gambar 1/47
Estimated : 1.267194229954727 Detik
[INFO] Memproses gambar 2/47
[INFO] Memproses gambar 3/47
Estimated : 1.1800174713134766 Detik
[INFO] Memproses gambar 4/47
Estimated : 1.1883246938916172 Detik
[INFO] Memproses gambar 5/47
Estimated : 3.764216423034668 Detik
[INFO] Memproses gambar 6/47
Estimated : 4.586746047266295 Detik
[INFO] Memproses gambar 7/47
Estimated : 7.695447292327861 Detik
[INFO] Memproses gambar 8/47
[INFO] Memproses gambar 9/47
Estimated : 7.598616871643066 Detik
[INFO] Memproses gambar 10/47
Estimated : 7.602221486952637 Detik
[INFO] Memproses gambar 11/47
Estimated : 11.127368323294687 Detik
[INFO] Memproses gambar 12/47
Estimated : 7.57212662698384 Detik
[INFO] Memproses gambar 13/47
[INFO] Memproses gambar 14/47
Estimated : 7.57659753646851 Detik
[INFO] Memproses gambar 15/47
Estimated : 7.57684747019922 Detik
[INFO] Memproses gambar 16/47
Estimated : 10.961581503949585 Detik
[INFO] Memproses gambar 17/47
Estimated : 10.865856466293335 Detik
[INFO] Memproses gambar 18/47
Estimated : 10.8697479342413 Detik
[INFO] Memproses gambar 19/47
Estimated : 10.70970942824768 Detik

```

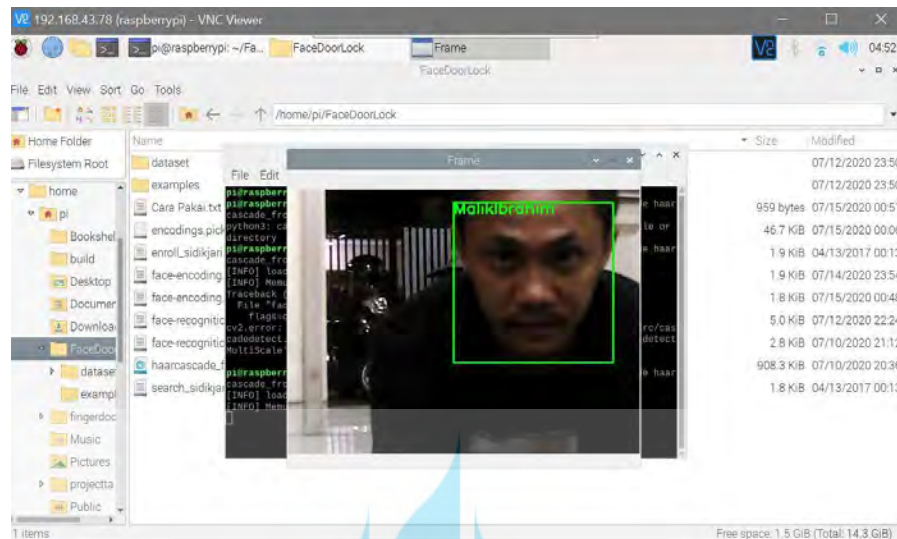
```

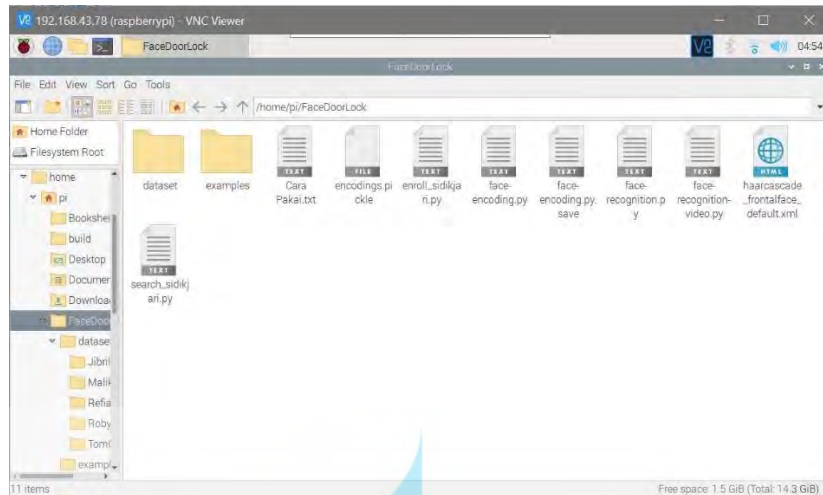
Estimated : 10.861581563949585 Detik
[INFO] Memproses gambar 17/47
Estimated : 10.805856466293335 Detik
[INFO] Memproses gambar 18/47
Estimated : 10.80974793434143 Detik
[INFO] Memproses gambar 19/47
Estimated : 10.78978943824768 Detik
[INFO] Memproses gambar 20/47
Estimated : 10.823493242263794 Detik
[INFO] Memproses gambar 21/47
Estimated : 10.815319061279297 Detik
[INFO] Memproses gambar 22/47
Estimated : 10.807718515396118 Detik
[INFO] Memproses gambar 23/47
Estimated : 10.731463432312612 Detik
[INFO] Memproses gambar 24/47
Estimated : 10.726816654205322 Detik
[INFO] Memproses gambar 25/47
Estimated : 10.911778450612207 Detik
[INFO] Memproses gambar 26/47
Estimated : 10.753458738327026 Detik
[INFO] Memproses gambar 27/47
Estimated : 10.75253582007324 Detik
[INFO] Memproses gambar 28/47
[INFO] Memproses gambar 29/47
Estimated : 150.15884804725647 Detik
[INFO] Memproses gambar 30/47
Estimated : 28.48354172766604 Detik
[INFO] Memproses gambar 31/47
Estimated : 55.03351092338562 Detik
Estimated : 55.033918380737305 Detik
[INFO] Memproses gambar 32/47
[INFO] Memproses gambar 33/47
Estimated : 13.534176349639893 Detik
[INFO] Memproses gambar 34/47
Estimated : 11.143922567367554 Detik
[INFO] Memproses gambar 35/47
Estimated : 2.0132901668548584 Detik
[INFO] Memproses gambar 36/47

```

```
[INFO] Memproses gambar 30/47
Estimated : 28.48354172706604 Detik
[INFO] Memproses gambar 31/47
Estimated : 55.03351092338562 Detik
Estimated : 55.033918380737305 Detik
[INFO] Memproses gambar 32/47
[INFO] Memproses gambar 33/47
Estimated : 13.534176349639893 Detik
[INFO] Memproses gambar 34/47
Estimated : 11.143922567367554 Detik
[INFO] Memproses gambar 35/47
Estimated : 2.0132901668548584 Detik
[INFO] Memproses gambar 36/47
[INFO] Memproses gambar 37/47
Estimated : 11.051708936691284 Detik
[INFO] Memproses gambar 38/47
Estimated : 7.909607172012329 Detik
[INFO] Memproses gambar 39/47
Estimated : 4.4877888614471436 Detik
Estimated : 4.497996807098389 Detik
[INFO] Memproses gambar 40/47
Estimated : 4.174440860748291 Detik
[INFO] Memproses gambar 41/47
Estimated : 3.9828262329101562 Detik
[INFO] Memproses gambar 42/47
Estimated : 2.445004940032959 Detik
[INFO] Memproses gambar 43/47
Estimated : 42.1389535489055 Detik
[INFO] Memproses gambar 44/47
Estimated : 42.098613262176514 Detik
[INFO] Memproses gambar 45/47
Estimated : 42.064011096954346 Detik
[INFO] Memproses gambar 46/47
Estimated : 42.30656361579895 Detik
[INFO] Memproses gambar 47/47
Estimated : 42.05816578865051 Detik
[INFO] Memproses serialize encoding...
pi@raspberrypi:~/FaceDoorLock $
```







```

from imutils.video import VideoStream
from imutils.video import FPS
import face_recognition
import argparse
import imutils
import pickle
import time
import cv2
import hashlib
from pyfingerprint.pyfingerprint import PyFingerprint
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

Relay = 17
GPIO.setup(Relay,GPIO.OUT)
GPIO.output(Relay,1)

sidikjari = 0

# Parsing Argumen
ap = argparse.ArgumentParser()
ap.add_argument("-c", "--cascade", required=True,
                help = "path to where the face cascade resides")
ap.add_argument("-e", "--encodings", required=True,
                help="path to serialized db of facial encodings")
args = vars(ap.parse_args())

```

```

# load pendeteksi wajah dari file cascade OpenCV
print("[INFO] loading encodings + face detector...")
data = pickle.loads(open(args["encodings"], "rb").read())
detector = cv2.CascadeClassifier(args["cascade"])
wajahdikenali = 0

# Nyalakan Kamera
print("[INFO] Memulai Stream dari Webcam...")
#cv2.namedWindow('frame')
vs = VideoStream(src=0).start()
time.sleep(2.0)

# Penghitung FPS (Frame per Second)
#fps = FPS().start()

# loop dari semua frame yang di dapat
while (wajahdikenali == 0):
    # dapatkan frame, dan resize ke 500pixel agar lebih cepat
    frame = vs.read()
    frame = imutils.resize(frame, width=500)

    # Konversi ke grayscale dan konversi ke RGB
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    # deteksi wajah dari frame grayscale
    rects = detector.detectMultiScale(gray, scaleFactor=1.1,
        minNeighbors=5, minSize=(30, 30),
        flags=cv2.CASCADE_SCALE_IMAGE)

# Tampilkan kotak di wajah yang dideteksi
boxes = [(y, x + w, y + h, x) for (x, y, w, h) in rects]

encodings = face_recognition.face_encodings(rgb, boxes)
names = []

# loop di semua wajah yang terdeteksi
for encoding in encodings:
    matches =
face_recognition.compare_faces(data["encodings"],
    encoding)
    name = "Unknown"
    wajahdikenali = 0

    # check apakah ada wajah yang di kenali
    if True in matches:
        matchedIdxs = [i for (i, b) in
enumerate(matches) if b]
        counts = {}
        for i in matchedIdxs:
            name = data["names"][i]
            counts[name] = counts.get(name, 0) + 1

```

```

        name = max(counts, key=counts.get)
        wajahdikenali = 1
        names.append(name)

# loop di semua wajah yang sudah di kenali
for ((top, right, bottom, left), name) in zip(boxes, names):
    # tampilkan nama di wajah yang di kenali
    cv2.rectangle(frame, (left, top), (right, bottom),
                  (0, 255, 0), 2)
    y = top - 15 if top - 15 > 15 else top + 15
    cv2.putText(frame, name, (left, y),
cv2.FONT_HERSHEY_SIMPLEX,
                0.75, (0, 255, 0), 2)

# Tampilkan gambar di layar
cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF

if wajahdikenali == 1:
    print ('Wajah Dikenali : ' + str(wajahdikenali))
    vs.stop()
    cv2.destroyAllWindows()
    break

## Tries to initialize the sensor
try:
    f = PyFingerprint('/dev/ttyUSB0', 57600, 0xFFFFFFFF, 0x00000000)

    if ( f.verifyPassword() == False ):
        raise ValueError('The given fingerprint sensor password is
wrong!')
except Exception as e:
    print('The fingerprint sensor could not be initialized!')
    print('Exception message: ' + str(e))
    exit(1)

## Tries to search the finger and calculate hash
try:
    print('Tempelkan Jari anda...')

    ## Wait that finger is read
    while ( f.readImage() == False ):
        pass

    ## Converts read image to characteristics and stores it in
charbuffer 1
    f.convertImage(0x01)

    ## Searchs template
    result = f.searchTemplate()

```

```

positionNumber = result[0]
accuracyScore = result[1]

if ( positionNumber == -1 ):
    print('No match found!')
    exit(0)
else:
    sidikjari = 1
    print('Found template at position #' + str(positionNumber))
    print('Tingkat Akurasi: ' + str(accuracyScore))

## OPTIONAL stuff
##

## Loads the found template to charbuffer 1
f.loadTemplate(positionNumber, 0x01)

## Downloads the characteristics of template loaded in
charbuffer 1
characterics = str(f.downloadCharacteristics(0x01)).encode('utf-
8')

## Hashes characteristics of template
print('Sidikjari ditemukan = ' + str(sidikjari))

except Exception as e:
    print('Operation failed!')
    print('Exception message: ' + str(e))
    exit(1)

if ((wajahdikenali == 1)and(sidikjari == 1)):
    print('Pintu Terbuka')
    GPIO.output(Relay,0)
    exit(1)
else:
    GPIO.output(Relay,1)

```