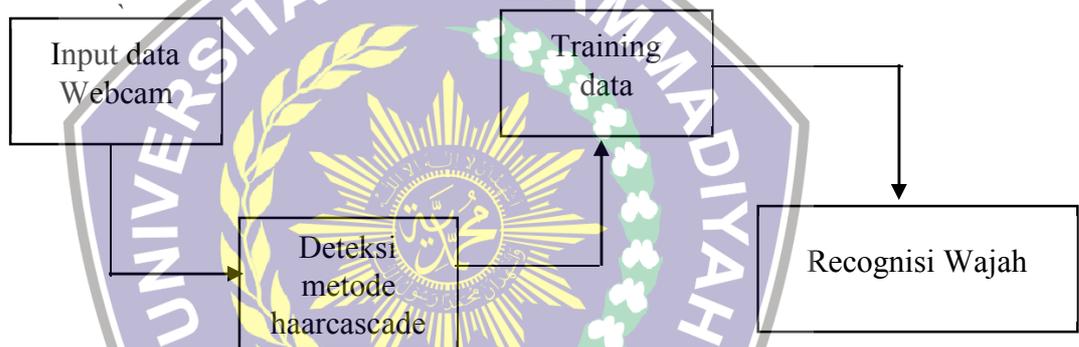


## BAB 4 ANALISA HASIL DAN PEMBAHASAN

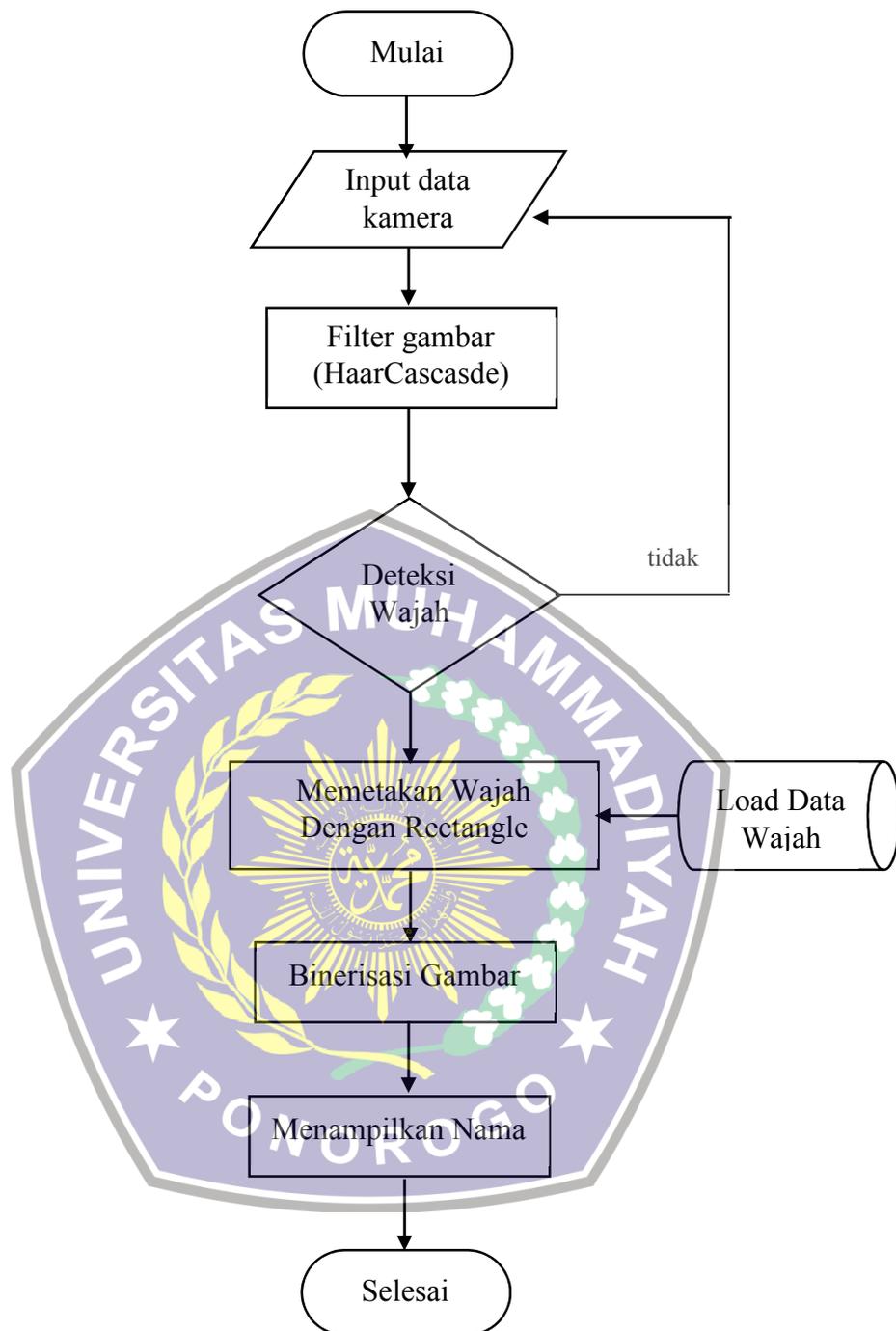
### 4.1 PERCOBAAN YANG DILAKUKAN

Percobaan yang dilakukan peneliti pada tugas akhir ini meliputi beberapa tahapan yaitu pengambilan data wajah, kemudian dari data wajah yang sudah diambil kemudian dilakukan proses pelatihan dalam mesin, yaitu percobaan bagaimana mesin dapat mengenali wajah yang sudah diambil data sebelumnya, atau nanti dikenal dengan istilah *Training*. Setelah dilakukan *training* akan dilakukan percobaan akan diteruskan ke tahap berikutnya yaitu pendeteksian wajah atau nanti dikenal dengan istilah *Recognisi*. Dibawah ini skema yang dilakukan pada penelitian ini.



Gambar 4.1 Alur Percobaan

Dari gambar diatas, secara garis besarnya deteksi wajah yang akan dilakukan adalah capture data wajah melalui webcam, kemudian dideteksi dengan menggunakan metode algoritma haarcascade, setelah itu dilakukan training data dari data wajah yang terekam, kemudian dilakukan pengenalan wajah sesuai dengan label nama yang disimpan dalam database.



Gambar 4.1 Flowcart Sistem Deteksi Wajah

#### 4.2 PENGAMBILAN DATA WAJAH

Pengambilan data wajah dilakukan dengan menggunakan webcam. Webcam bisa menggunakan webcam internal atau webcam eksternal. Peneliti sudah melakukan beberapa pengambilan wajah dengan menggunakan

beberapa webcam, baik itu internal maupun eksternal. Dari hasil percobaan yang sudah dilakukan didapatkan hasil bahwa wajah dapat dikenali dengan baik menggunakan kedua jenis webcam tersebut.

Pada penelitian ini peneliti mengambil beberapa gambar wajah dari beberapa orang. Pada proses pengambilan gambar wajah, webcam mengambil gambar wajah untuk dideteksi. Proses pendeteksian wajah menggunakan metode *algoritma haar cascade*. Pemilihan algoritma ini dikarenakan algoritma *haarcascade* sudah sangat terkenal untuk mendeteksi wajah manusia. File algoritma ini diambil dari website [https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade\\_frontalface\\_default.xml](https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml).

Berikut adalah beberapa capture pendeteksian data wajah yang dilakukan oleh sistem.





Gambar 4.2 Pengambilan gambar secara *real time*

Pada gambar di atas, terlihat sistem dapat mendeteksi wajah dengan ditandai frame berwarna merah. Deteksi wajah dapat dilakukan jika semua orang menghadapkan wajah mereka ke webcam. Sistem pun dapat mendeteksi beberapa wajah sekaligus terhadap beberapa orang yang menghadapkan wajahnya ke kamera, seperti yang ditunjukkan pada gambar d, e, dan f.

Dari kelima orang yang diambil gambarnya untuk dideteksi, sistem mampu menunjukkan tingkat akurasi 100% dengan posisi semua orang menghadapkan wajahnya ke kamera.

Adapun rumus yang digunakan adalah:

$$\% = \frac{\text{jumlah data yang terdeteksi}}{\text{jumlah yang diambil}} \times 100\%$$

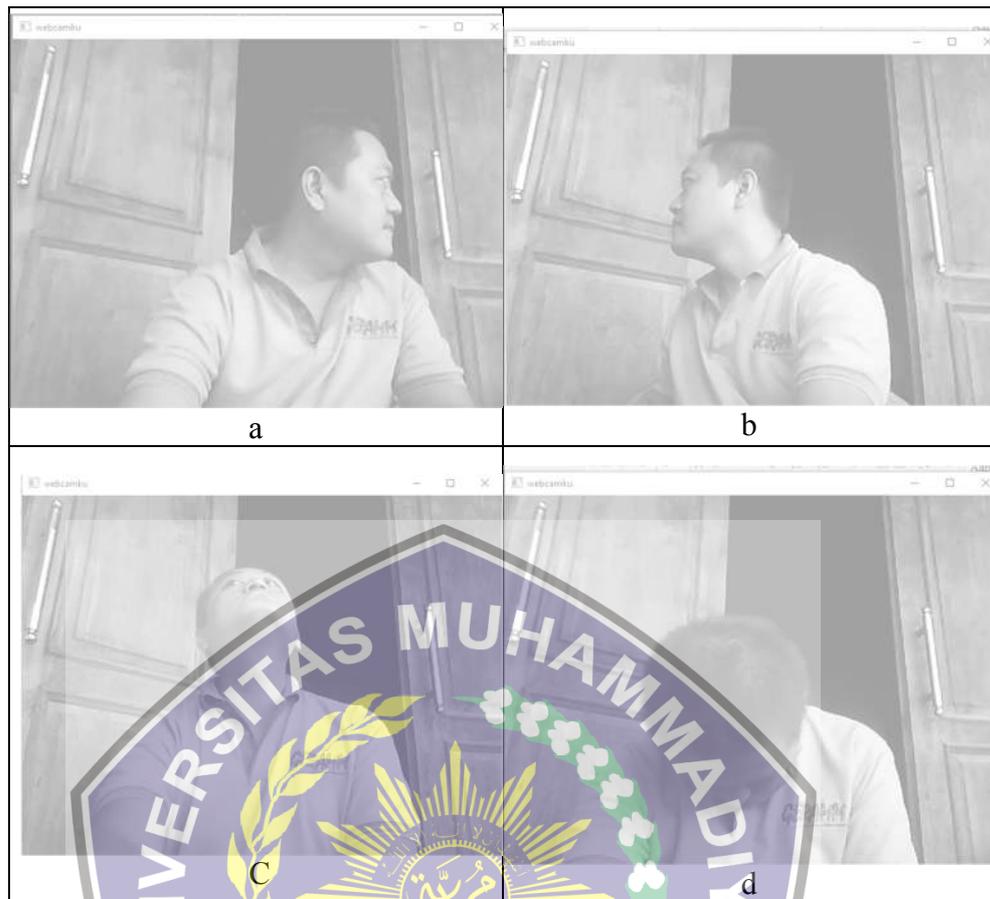
jumlah yang terdeteksi = 5

jumlah data yang diambil = 5

jadi akurasi yang dihasilkan adalah:

$$= 5/5 \times 100\% = 100\%$$

Berikut adalah beberapa capture wajah dengan beberapa posisi wajah dengan dipalingkan dari kamera.



Gambar 4.3 Pengambilan Gambar dengan posisi tidak sesuai dengan kamera

Dari keempat capture wajah yang diambil gambarnya untuk dideteksi, sistem tidak dapat mendeteksi wajah dengan akurasi 0% dengan posisi semua wajah dipalingkan dari kamera dan atau tidak menghadap ke kamera.

Adapun rumus yang digunakan adalah:

$$\% = \frac{\text{jumlah data yang terdeteksi}}{\text{jumlah yang diambil}} \times 100\%$$

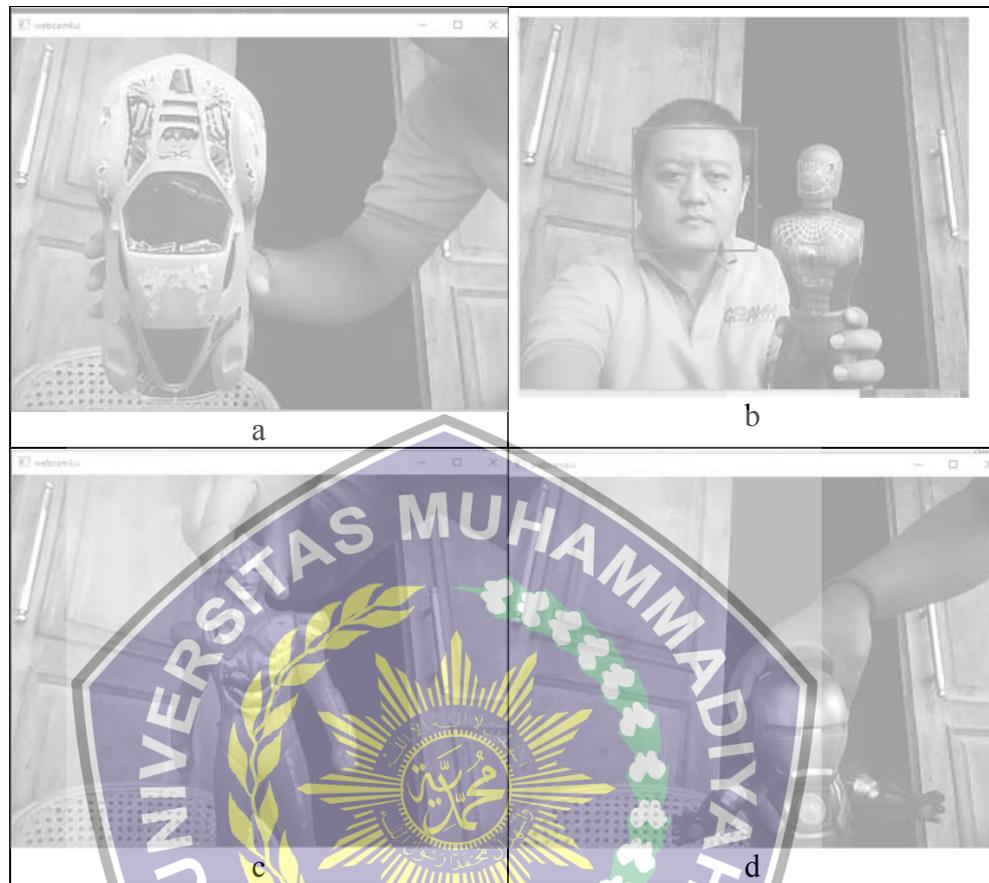
jumlah yang terdeteksi = 0

jumlah data yang diambil = 5

jadi akurasi yang dihasilkan adalah:

$$= 0/5 \times 100\% = 0\%$$

Berikut adalah capture dari objek bukan wajah.



Gambar 4.4 Pengambilan Gambar Selain Wajah

Dari gambar di atas, terlihat sistem tidak dapat mendeteksi objek yang bukan wajah manusia dengan akurasi 0%.

Adapun rumus yang digunakan adalah:

$$\% = \frac{\text{jumlah data yang terdeteksi}}{\text{jumlah yang diambil}} \times 100\%$$

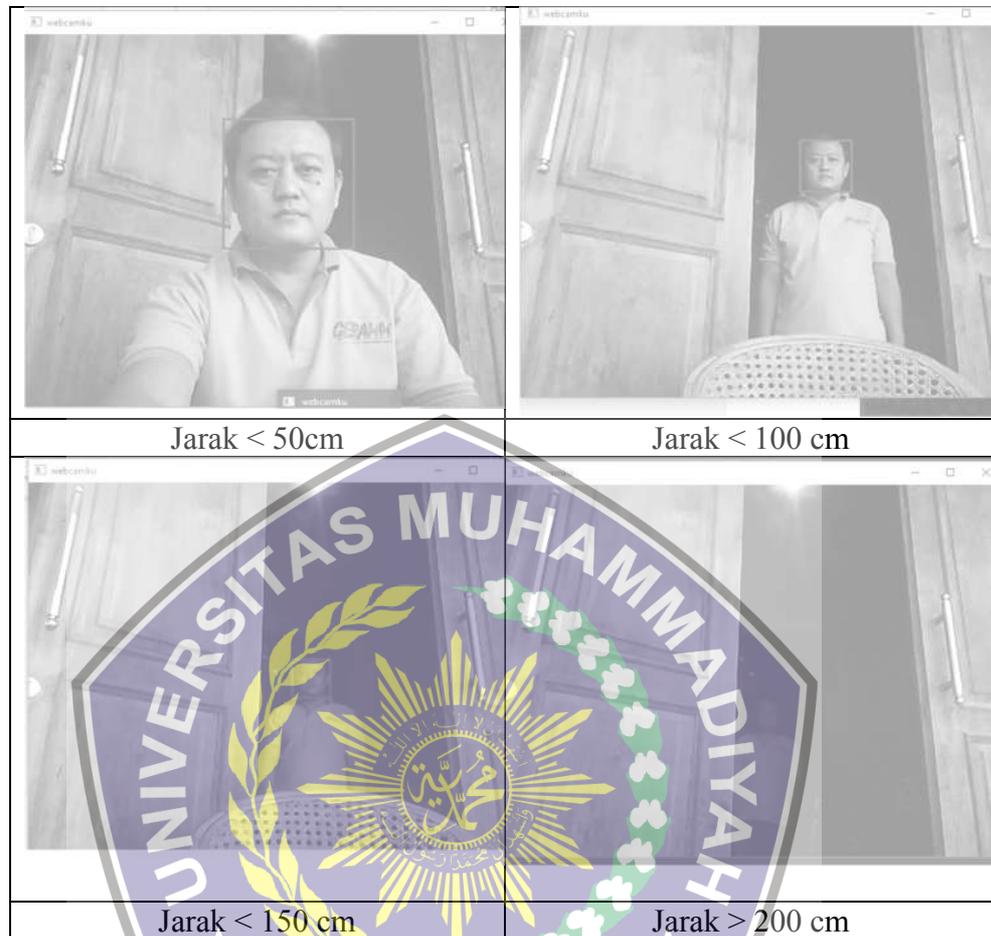
jumlah data yang terdeteksi = 0

jumlah yang diambil = 5

jadi akurasi yang dihasilkan adalah:

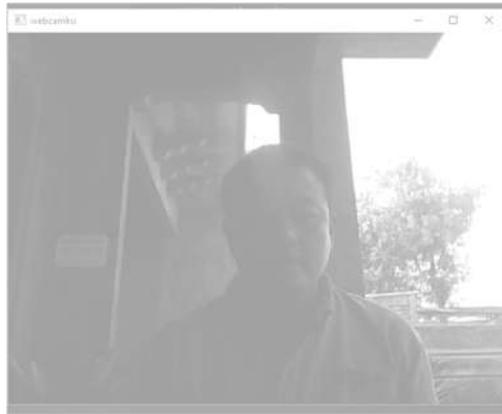
$$= 0/5 \times 100\% = 0\%$$

Berikut adalah capture wajah berdasarkan jarak dari webcam



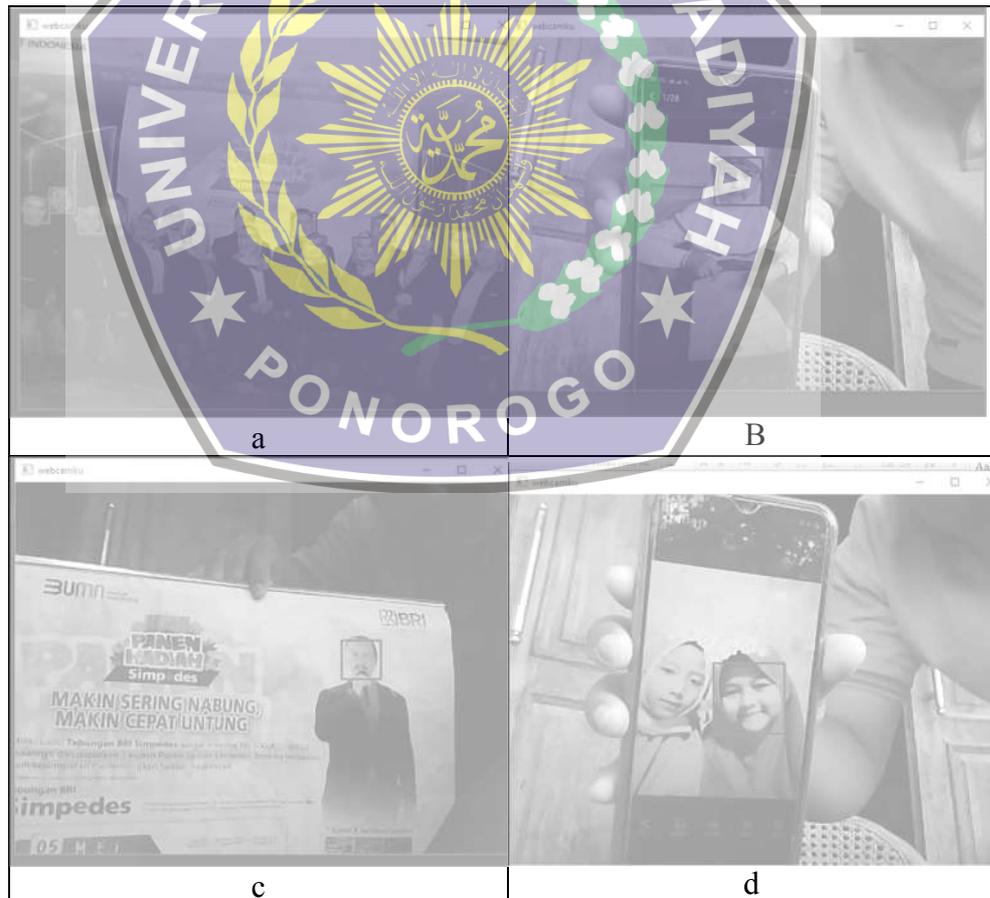
Gambar 4.5 Pengambilan Gambar dengan Perbedaan Jarak terhadap kamera

Dari gambar di atas, terlihat sistem masih dapat mendeteksi wajah dari webcam dengan jarak kurang dari 150 cm. Jika jarak kamera > 200 cm, sistem tidak dapat mendeteksi objek.



Gambar 4.6 Pengambilan Gambar Dengan Intensitas Cahaya Yang Kurang  
Gambar di atas, menunjukkan sistem tidak dapat mendeteksi objek dengan pencahayaan yang kurang.

Berikut adalah beberapa capture wajah yang diambil secara tidak langsung dari webcam.



Gambar 4.7 Pengambilan Gambar dari Foto Wajah

Dari gambar di atas, terlihat bahwa sistem dapat mendeteksi objek yang diambil secara tidak langsung dari webcam. Objek diambil dari kalender yang terdapat data wajah, dan image yang terdapat pada handphone. Tingkat akurasi sistem untuk mendeteksi objek mencapai 70% dari 20 data.

Adapun rumus yang digunakan adalah:

$$\% = \frac{\text{jumlah data yang terdeteksi}}{\text{jumlah yang diambil}} \times 100\%$$

jumlah yang terdeteksi = 14

jumlah data yang diambil = 20

jadi akurasi yang dihasilkan adalah:

$$= 14/20 \times 100\% = 70\%$$

Berikut adalah capture wajah dengan berbagai kondisi





Gambar 4.8 Pengambilan Gambar dengan Berbagai Kondisi

Keterangan gambar 4.8

- Pada gambar (a) terlihat hanya 1 saja orang yang terdeteksi dan tiga orang tidak terdeteksi. Hal ini dikarenakan ada dua orang yang memalingkan wajahnya dari kamera, dan ada orang yang menutupi wajahnya dengan menggunakan masker.
- Pada gambar (b), terlihat 2 orang yang terdeteksi, sedangkan dua orang tidak terdeteksi. Satu orang menutupi wajahnya dengan tangan, dan satu orang lagi tidak menghadapkan wajahnya ke webcam.
- Pada gambar (c), wajah bisa terdeteksi meskipun pengguna menggunakan kacamata.
- Pada gambar (d), wajah bisa terdeteksi meskipun pengguna menggunakan kacamata dan tutup kepala.

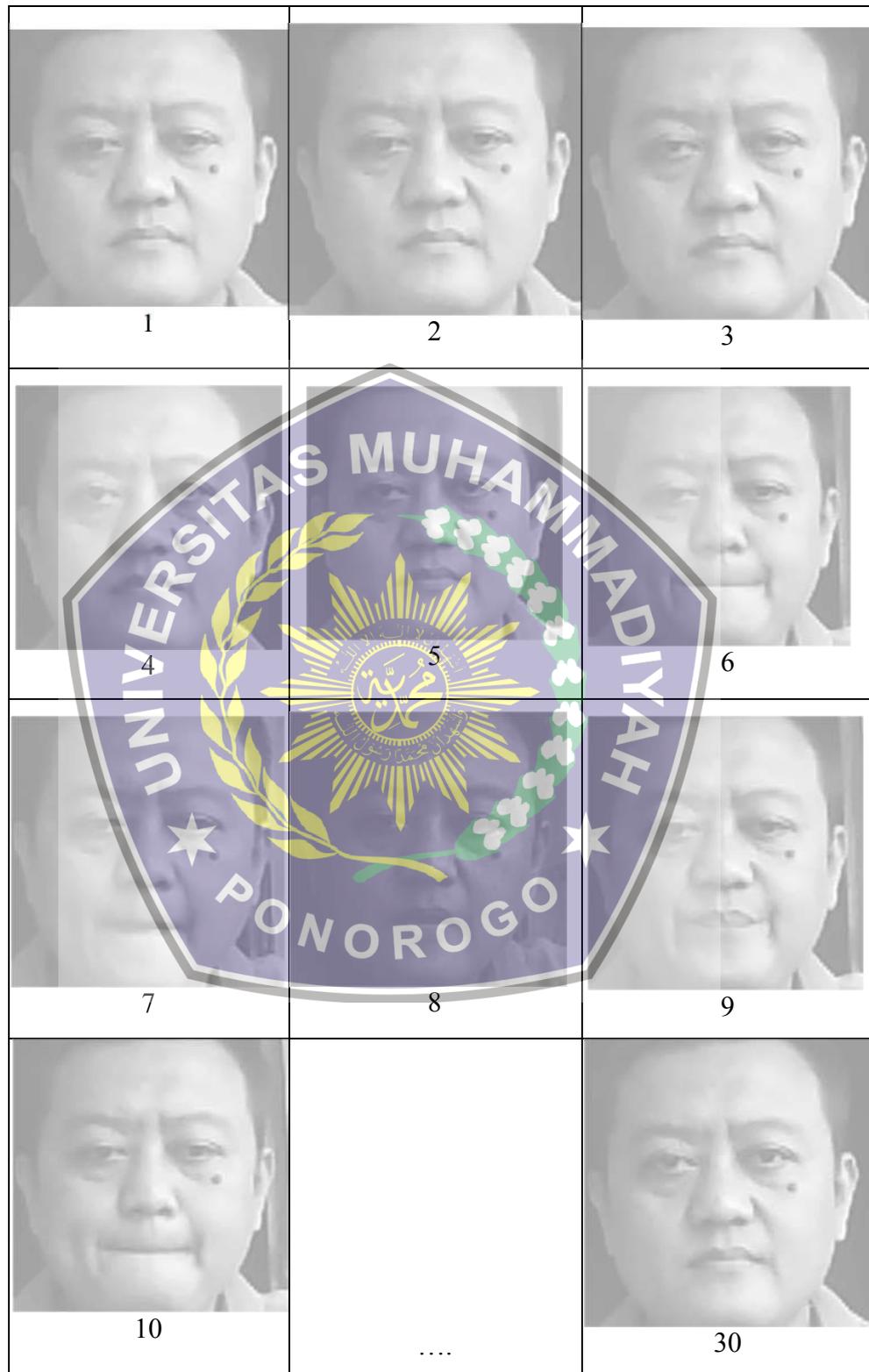
### 4.3 PROSES GRAYSCALLING

Setelah dilakukan proses pengambilan wajah, tahap berikutnya adalah proses grayscale. Awal mula tahapan yang dilakukan pada metode *haar cascade classifier* adalah dengan mengubah citra berwarna menjadi citra *grayscale*. Setelah itu, proses selanjutnya melakukan pemindaian pada citra *grayscale* untuk mendapatkan nilai fitur dari citra tersebut.

0	1	2	...	128	...	...	...	...	255

Gambar 4.9 Proses *Grayscale*

Berikut ini adalah hasil capture wajah dari webcam.



Gambar 4.11 Pengubahan Gambar Citra Menjadi Grayscale

#### 4.4 TRAINING DATA WAJAH

Setelah rekam data wajah dilakukan, tahapan berikutnya adalah mentraining data. Maksud dari mentraining data adalah membaca dari seluruh image yang sudah kita simpan pada direktori. Dari data wajah pada direktori, kemudian data akan ditraining berdasarkan id. Id ini harus disesuaikan dengan nama id.

#### 4.5 PENGENALAN WAJAH (*FACE RECOGNITION*)

Setelah training data wajah dilakukan, proses berikutnya adalah proses recognisi wajah, yaitu pengenalan dan identifikasi wajah. Pengenalan wajah menggunakan algoritma *Local Binary Patterns Histogram (LPBH)*. Konsepnya adalah dengan membaca directory dari seluruh image yang sudah ditraining dan komputer akan menebak sampel wajah yang ada di depan webcam.

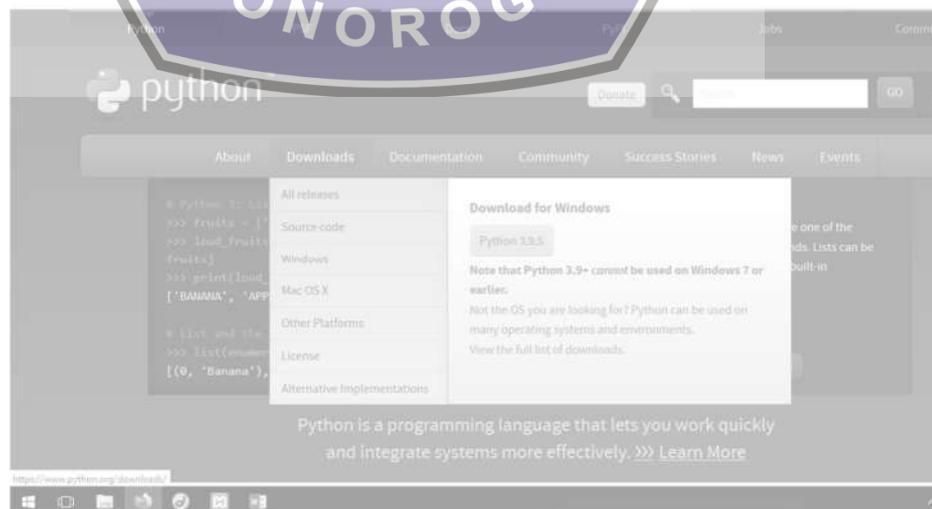
#### 4.6 PENGEMBANGAN

Pada tahapan ini akan dijelaskan bagaimana cara menginstalasi program python berikut dengan atributnya.

##### a. Instalasi Python

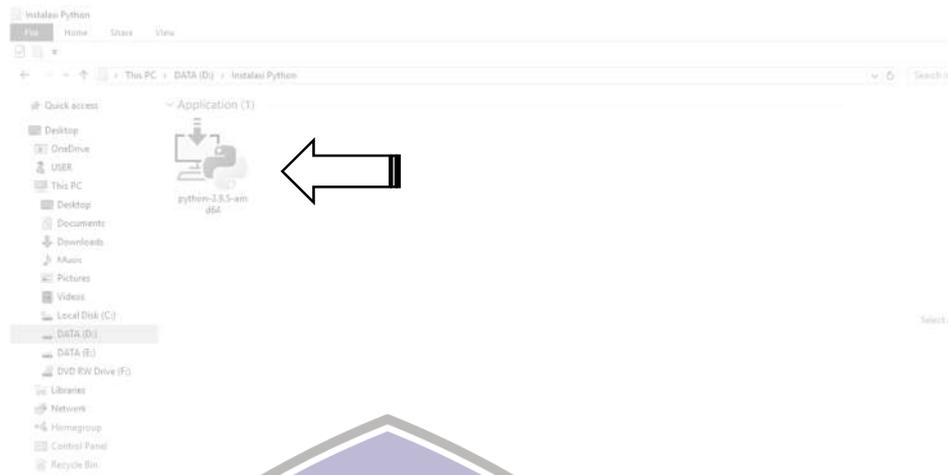
Berikut cara instalasi Python pada OS Windows.

1. Mendownload file instalasi di situs resmi python ([python.org](http://python.org))



Gambar 4.11 Tampilan Situs Resmi Pyhton.org

2. Membuka file python python-3.9.5-amd yang sudah didownload, kemudian menekan dua kali pada ikon software python 3.9.5



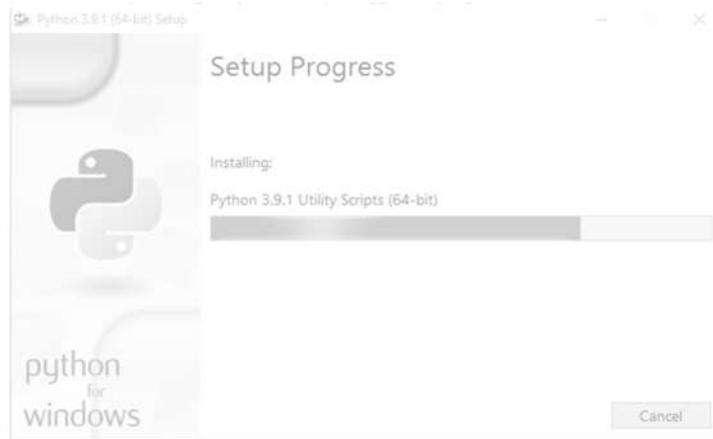
Gambar 4.12 Tampilan Ikon Software Instalasi Python-3.9.5

3. Memberi tanda centang (✓) *install launcher for all user (recommended)* dan *Add Python 3.9 to PATH* untuk penambahan *path command Python*, lalu jalankan proses **Klik Install Now** dan klik **Yes** bila ada notifikasi.



Gambar 4.13 Tampilan *Customize Setup Python*

4. Proses setup sedang berjalan



Gambar 4.14 Tampilan Setup Program Instalasi Python

5. Proses Setup telah selesai, Klik Close



Gambar 4.15 Tampilan Instalasi Python berhasil diinstal

6. Menguji coba Python

```

Python 3.9 (64-bit)
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print "Hello World"
File "<stdin>", line 1
  print "Hello World"
    ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print("Hello World")?
>>> print ("Hello World")
Hello World
>>> nama = "Irfan Fauzi"
>>> print ("Perkenalkan saya "+nama)
Perkenalkan saya Irfan Fauzi
>>>

```

Gambar 4.16 Tampilan Python pada IDLE

7. Setelah Python sukses diinstal di program Windows, perlu ditambahkan instalasi openCV untuk mendukung penelitian ini.

Instalasi untuk openCV dapat dilakukan dengan menggunakan perintah *pip install opencv-python* melalui *cmd* atau *IDLE Python*.

b. Perekaman Data Wajah

- Pembuatan library

```

import cv2,
os

```

cv2; merupakan library openCV yang terdapat pada python  
os; yaitu modul yang ada di python untuk untuk berinteraksi langsung dengan sistem operasi

- Proses Pendeteksian Wajah

```

import cv2
cam = cv2.VideoCapture(1)
cam.set(3, 640)#ubah lebar cam
cam.set(4, 480)#ubah tinggi cam
faceDetector = cv2.CascadeClassifier('C:\\Users\\USER\\pe
ngenalanjajah\\haarcascade_frontalface_default.xml')

```

Algoritma haarcascade dapat digunakan untuk melakukan identifikasi wajah manusia. Identifikasi dilakukan berdasarkan mata, hidung, dan mulut.

- Membalikkan kamera dan perubahan warna menjadi *grayscale*

```
- retV, frame = cam.read()
-     frame = cv2.flip(frame, 1)
-     Gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

Kode ini digunakan untuk mengubah image citra warna menjadi grayscale.

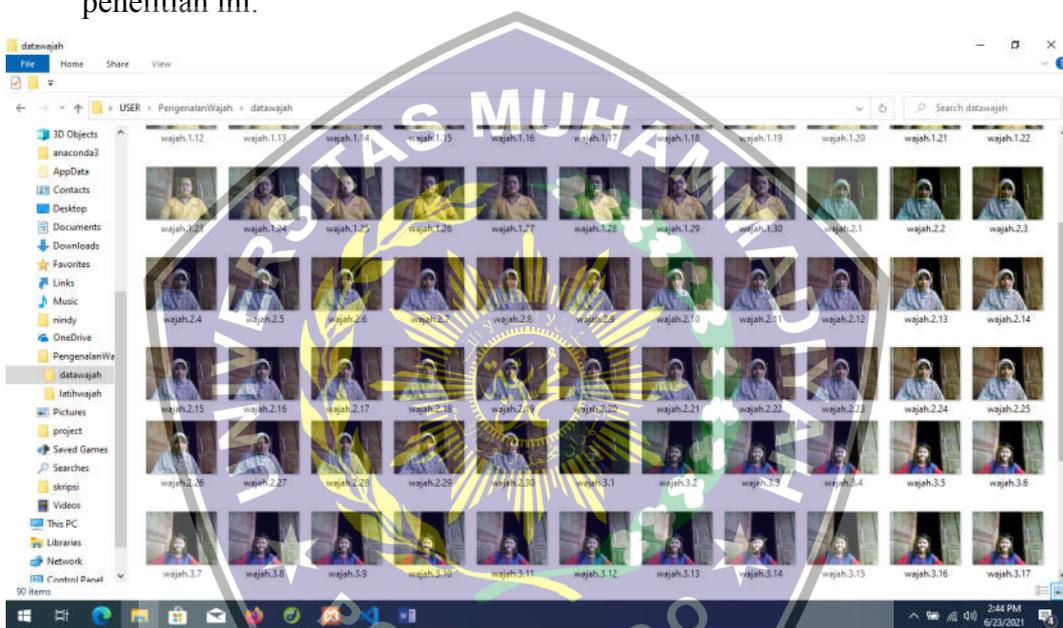
Berikut adalah *source code* yang digunakan dalam proses pengambilan data wajah.

```
import cv2, os
wajahDir = 'datawajah'
cam = cv2.VideoCapture(1)
cam.set(3, 640)#ubah lebar cam
cam.set(4, 480)#ubah tinggi cam
faceDetector = cv2.CascadeClassifier('C:\\Users\\USER\\pengenalanwajah\\haarcascade_frontalface_default.xml')
eyeDetector = cv2.CascadeClassifier('C:\\Users\\USER\\pengenalanwajah\\haarcascade_eye.xml')
faceID = input("Masukkan Face ID yang akan direkam datanya [kemudian tekan ENTER]:")
print ("Tatap Wajah Anda ke depan dalam Webcam. Tunggu proses pengambilan data wajah selesai.")
ambilData = 1
while True:
    retV, frame = cam.read()
    Gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = faceDetector.detectMultiScale(Gray, 1.3,5)
    for (x, y, w, h) in faces:
        frame = cv2.rectangle(frame, (x,y), (x+w, y+h), (0,255,255), 2)

    namaFile = 'wajah.' +str(faceID)+'.'+str(ambilData)+' .jpg'
    cv2.imwrite(wajahDir+'/' +namaFile,frame)
    ambilData += 1
    roiGray = Gray[y:y+h, x:x+w]
    roiWarna = frame[y:y+h,x:x+w]
    eyes = eyeDetector.detectMultiScale(roiGray)
    for (xe, ye, we, he) in eyes:
        cv2.rectangle (roiWarna, (xe,ye), (xe+we,ye+he), (0,0,255),1)
2)
```

```
cv2.imshow('webcamku', frame)
#cv2.imshow('webcamku2', Gray)
k = cv2.waitKey(1) & 0xFF
if k == 27 or k == ord('q'):
    break
elif ambilData>30:
    break
print("Pengambilan Data Selesai")
cam.release()
cv2.destroyAllWindows
```

Berikut adalah hasil pengambilan gambar wajah yang digunakan untuk penelitian ini.



Gambar 4.17 Hasil Pengambilan Gambar Melalui Webcam

c. Proses Training

Training disini dimaksudkan untuk melatih mesin computer untuk dapat mengenali dari data gambar yang sudah diambil sebelumnya. Dibawah ini adalah *source code* dalam men-*train* data wajah dan penjelasannya.

- Pembuatan Library:

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

Cv2 adalah *library openCV*

Numpy digunakan untuk operasi vector dan matriks.

PIL; library digunakan untuk pemrosesan gambar. PIL dapat dikatakan sebagai serangkaian tahapan untuk mengelola dan menganalisis gambar dengan melibatkan persepsi visual secara digital.

os; yaitu modul yang ada di python untuk untuk berinteraksi langsung dengan sistem operasi

Hasil dari proses training didapatkan file berekstensi .xml sebagaimana ditunjukkan pada gambar di bawah ini.



Gambar 4.18 Hasil Training

- Proses Recognisi Wajah

```
- faceRecognizer = cv2.face.LBPHFaceRecognizer_create()  
- faceDetector = cv2.CascadeClassifier('C:\\Users\\USER\\pe  
ngenalanwajah\\haarcascade_frontalface_default.xml')
```

Untuk pengenalan wajah, digunakan algoritma *Local Binary Pattern Histogram (LBPH)*, yaitu dengan membuat klasifikasi pengenalan wajah berdasarkan nomor id.

- Proses perubahan gambar grayscale menjadi data integer

```
- imgNum = np.array(PILimg, 'uint8')
```

Pada tahapan ini gambar diekstraksi ke dalam bentuk integer untuk mencari nilai fitur.

- Proses pembuatan integer

```
- faces,IDs = getImageLabel(wajahDir)
- faceRecognizer.train(faces,np.array(IDs))
```

Pada tahapan pembuatan integer dilakukan yaitu dengan cara membagi gambar menjadi 8x8 dan setiap kotak akan dibagi 3x3 kemudian dilakukan proses pelatihan (*training*) untuk mengambil nilai rata-rata *grayscale* pada gambar.

Dibawah ini adalah source code yang digunakan dalam tahapan ini.

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

wajahDir = 'datawajah'
latihDir = 'latihwajah'
def getImageLabel(path):
    imagePath = [os.path.join(path,f) for f in os.listdir(path)]
    faceSamples = []
    faceIDs = []
    for imagePath in imagePath:
        PILimg = Image.open(imagePath).convert('L')
        imgNum = np.array(PILimg,'uint8')
        faceID = int(os.path.split(imagePath)[-1].split(".")[1])
        faces = faceDetector.detectMultiScale(imgNum)
        for (x, y, w, h) in faces:
            faceSamples.append(imgNum[y:y+h,x:x+w])
            faceIDs.append(faceID)
    return faceSamples,faceIDs

faceRecognizer = cv2.face.LBPHFaceRecognizer_create()
faceDetector = cv2.CascadeClassifier('C:\\Users\\USER\\pengenalanwajah\\haarcascade_frontalface_default.xml')

print ("Mesin sedang melakukan training data wajah. Tunggu dalam beberapa detik.")
faces,IDs = getImageLabel(wajahDir)
faceRecognizer.train(faces,np.array(IDs))
#simpan
faceRecognizer.write(latihDir+'/training.xml')
print ("Sebanyak {0} data wajah telah ditrainingkan ke mesin." ,format(len(np.unique(IDs))))
```

#### d. Pengenalan Wajah/Recognisi

- Pembuatan Library

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

- import Cv2 adalah *library openCV*
- Numpy digunakan untuk operasi vector dan matriks.
- PIL; library digunakan untuk pemrosesan gambar. PIL dapat dikatakan sebagai serangkaian tahapan untuk mengelola dan menganalisis gambar dengan melibatkan persepsi visual secara digital.
- os; yaitu modul yang ada di python untuk berinteraksi langsung dengan sistem operasi
- Proses Pengenalan Wajah

```
faceDetector = cv2.CascadeClassifier('C:\\Users\\USER\\pe
ngenalanwajah\\haarcascade_frontalface_default.xml')
faceRecognizer = cv2.face.LBPHFaceRecognizer_create()
faceRecognizer.read(latihDir+'training.xml')
```

Pada tahapan ini dilakukan gambar yang diperoleh akan dibaca oleh mesin, sehingga gambar masukan dari kamera dapat dicocokkan dengan gambar yang terdapat pada database

- Proses pemberian nama berdasarkan nomor id

```
id= 0
names = ['Tidak diketahui', 'Irfan Fauzi', 'Ata']
```

Pada proses ini diberikan nama berdasarkan nomor id

- Proses pencocokan gambar

```

- if confidence<=50 :
-     nameID = names[id]
-     confidenceTxt= " {0}%".format(round(100-
- confidence))
- else :
-     nameID = names[0]
-     confidenceTxt = " {0}%".format(round(100 - co
- nfidence))

```

Pada proses ini gambar masukan dari kamera akan dicocokkan dengan gambar yang terdapat pada *dataset* dan apabila ditemukan kecocokan berdasarkan nilai *confidence* maka diberikan label sesuai nama yang telah terdaftar berdasarkan nomor id.

Berikut adalah kode untuk pengenalan wajah

```

import cv2, os, numpy as np

wajahDir = 'datawajah'
latihDir = 'latihwajah'
cam = cv2.VideoCapture(1)
cam.set(3, 640)#ubah lebar cam
cam.set(4, 480)#ubah tinggi cam
faceDetector = cv2.CascadeClassifier('C:\\Users\\USER\\pengenalanwaj
ah\\haarcascade_frontalface_default.xml')
faceRecognizer = cv2.face.LBPHFaceRecognizer_create()

faceRecognizer.read(latihDir+'/training.xml')
font = cv2.FONT_HERSHEY_SIMPLEX

id= 0
names = ['Tidak diketahui', 'Irfan Fauzi', 'Ata']

minWidth = 0.1*cam.get(3)
minHeight = 0.1*cam.get(4)

while True:
    retV, frame = cam.read()
    frame = cv2.flip(frame, 1)
    Gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = faceDetector.detectMultiScale(Gray, 1.2, 5, minSize=(rou
nd(minWidth),round(minHeight)))
    for (x, y, w, h) in faces:

```

```

frame = cv2.rectangle(frame, (x,y), (x+w, y+h), (0,255,0),2)
id, confidence = faceRecognizer.predict(Gray[y:y+h,x:x+w])
if confidence<=50 :
    nameID = names[id]
    confidenceTxt= " {0}%".format(round(100-confidence))
else :
    nameID = names[0]
    confidenceTxt = " {0}%".format(round(100 - confidence))
cv2.putText(frame,str(nameID), (x+5,y-
5),font,1,(255,255,255),2)
cv2.putText(frame,str(confidenceTxt), (x + 5, y+h-
5), font, 1,(255, 255, 0), 1)

cv2.imshow('Recognisi Wajah', frame)
#cv2.imshow('webcamku2', Gray)
k = cv2.waitKey(1) & 0xFF
if k == 27 or k == ord('q'):
    break
print("EXIT")
cam.release()
cv2.destroyAllWindows

```

Berikut adalah hasil pengenalan wajah pada sistem serta persentase kemiripan dengan data *training* pada gambar.



Gambar 4.19 Wajah dikenali oleh mesin

Nampak pada gambar 4.9 wajah dapat dikenali dengan nama id yang sudah direkam sebelumnya.



Gambar 4.20 Wajah tidak dikenali mesin

Sedangkan pada gambar diatas, karena user menambahkan jari-jari tangannya.

#### 4.7 PENGUJIAN

Peneliti melakukan uji coba terhadap beberapa orang dan didapatkan hasil sebagai berikut:

Topik pengujian	Rekam data		
Tanggal pengujian	8 Mei 2021		
Lama pengujian	1 menit		
Penguji	Hesty Fatmawati		
No	Fungsi Pokok	Sesuai	
		Ya	Tidak
1.	Memasukkan nomor id	1	
2.	Menangkap gambar	1	
3.	Menyimpan data	1	
	Jumlah	3	0

Pada tabel di atas, merupakan hasil dari pengujian kepada penguji terhadap rekam data oleh Hesty Fatmawati sebagai penguji. Hasilnya tiga kasus pengujian sesuai dengan harapan peneliti.

Topik pengujian	<i>Training Data</i>
Tanggal pengujian	8 Mei 2021
Lama pengujian	1 menit
Penguji	Hesty Fatmawati

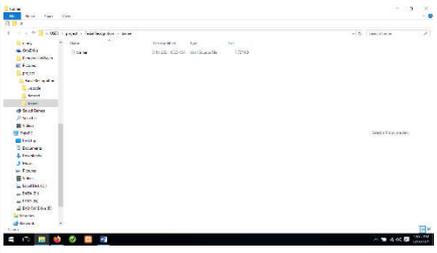
No	Fungsi Pokok	Sesuai	
		Ya	Tidak
1.	Melakukan <i>training data</i>		1
Jumlah			1

Pada tabel di atas, merupakan hasil dari pengujian kepada pengguna terhadap *training data* oleh Hesty Fatmawati sebagai pengguna. Hasilnya satu kasus pengujian sesuai dengan harapan peneliti.

Topik pengujian	Pengenalan Wajah		
Tanggal pengujian	8 Mei 2021		
Lama pengujian	1 menit		
Penguji	Hesty Fatmawati		
No	Fungsi Pokok	Sesuai	
		Ya	Tidak
1.	Mengenali wajah		1
Jumlah			1

Pada tabel di atas, merupakan hasil dari pengujian kepada penguji terhadap pengenalan wajah oleh Hesty Fatmawati sebagai penguji. Hasilnya tiga kasus pengujian sesuai dengan harapan peneliti.

No	Sistem	Pengujian	Validasi	Hasil
1.	Rekam data	Pengambilan 30 gambar wajah yang akan disimpan menjadi sebuah folder dan pemberian nomor id		Berhasil

2.	Training data	Melatih mesin untuk membaca gambar dengan menggunakan id		Belum Berhasil
3.	Pengenalan Wajah	Mampu mengenal wajah orang yang berbeda-beda dari webcam		Belum berhasil

