

# PRESENTASI PROJEK

# COMPUTER VISION



# Anggota Kelompok



2440016905

*Mario*



2401962402

*Lawysen*



2401954205

*Nelsen Anggara*



# Project yang dibuat



Project yang kami buat adalah suatu program untuk absensi dengan menggunakan wajah sebagai tanda untuk kehadiran



# DEMONSTRASI FACE RECOGNITION



# Import Libraries dan Picture

```
import cv2
import numpy as np
import face_recognition

# ambil gambar dari file yang uda di sediain
imgTrain = face_recognition.load_image_file('./jennyTraining.jpg')
# ubah warna gambar tersebut menjadi rgb
imgTrain = cv2.cvtColor(imgTrain, cv2.COLOR_BGR2RGB)

imgTest = face_recognition.load_image_file('./ImageAttendance/jenny.jpg')
imgTest = cv2.cvtColor(imgTest, cv2.COLOR_BGR2RGB)
```

## Step Awal :

- Import Libraries yang akan digunakan
- Mengimport semua gambar yang akan kita gunakan
- Mengubah color space gambar dari BGR ke RGB



# Gambar Yang Digunakan



imgTrain



imgTest

# Mendeteksi Wajah

Pada step ini kita akan mendeteksi lokasi wajah serta meng-encode gambar yang telah dimasukkan pada step sebelumnya serta menambahkan sebuah kotak

## Deteksi Wajah imgTrain

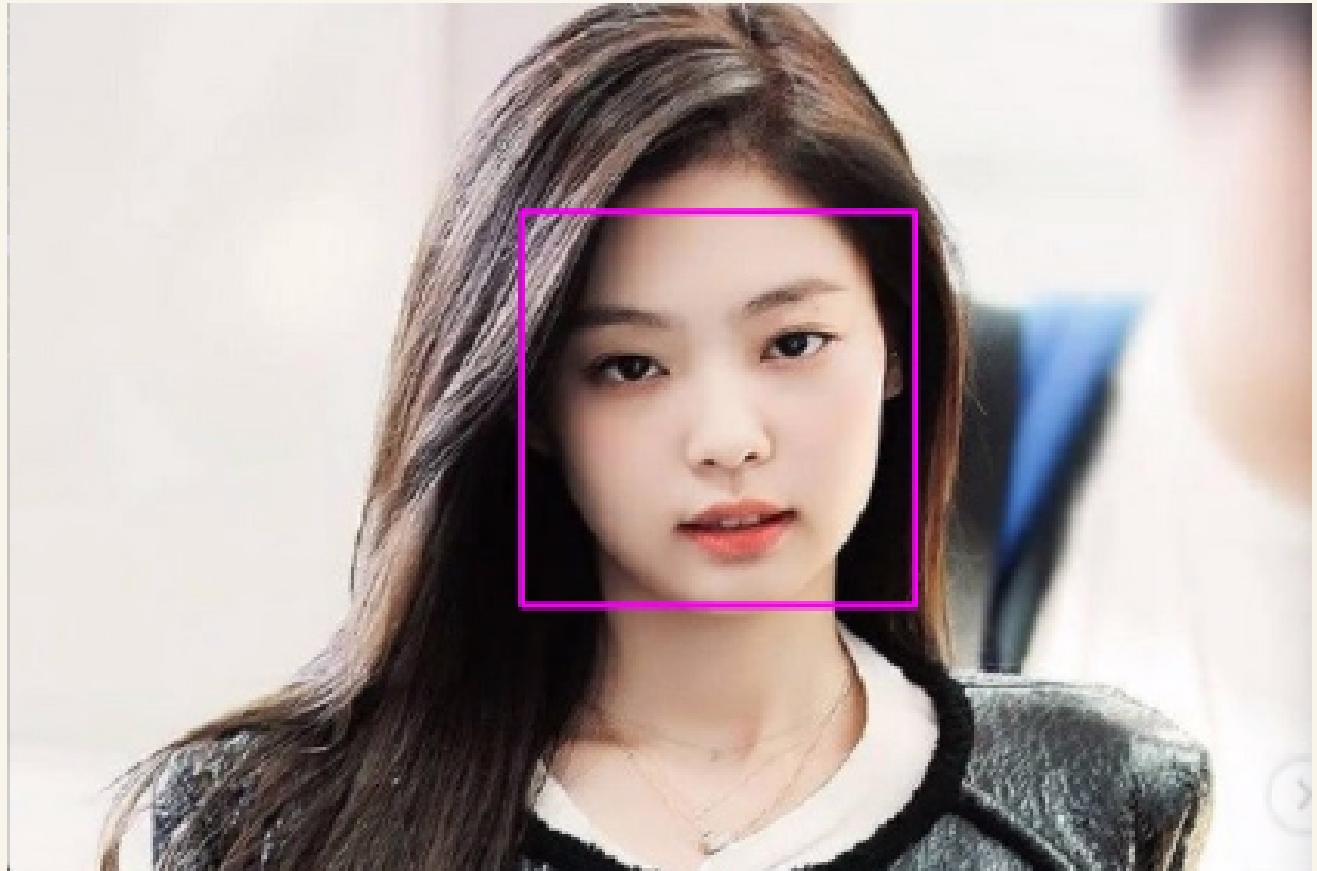
```
# mencari lokasi wajah
faceLoc = face_recognition.face_locations(imgTrain)[0]
encodeTrain = face_recognition.face_encodings(imgTrain)[0]
# memberikan kotak kepada lokasi wajah yang telah di temukan
cv2.rectangle(imgTrain, (faceLoc[3], faceLoc[0]),
              (faceLoc[1], faceLoc[2]), (255, 0, 255), 2)
```

## Deteksi Wajah imgTest

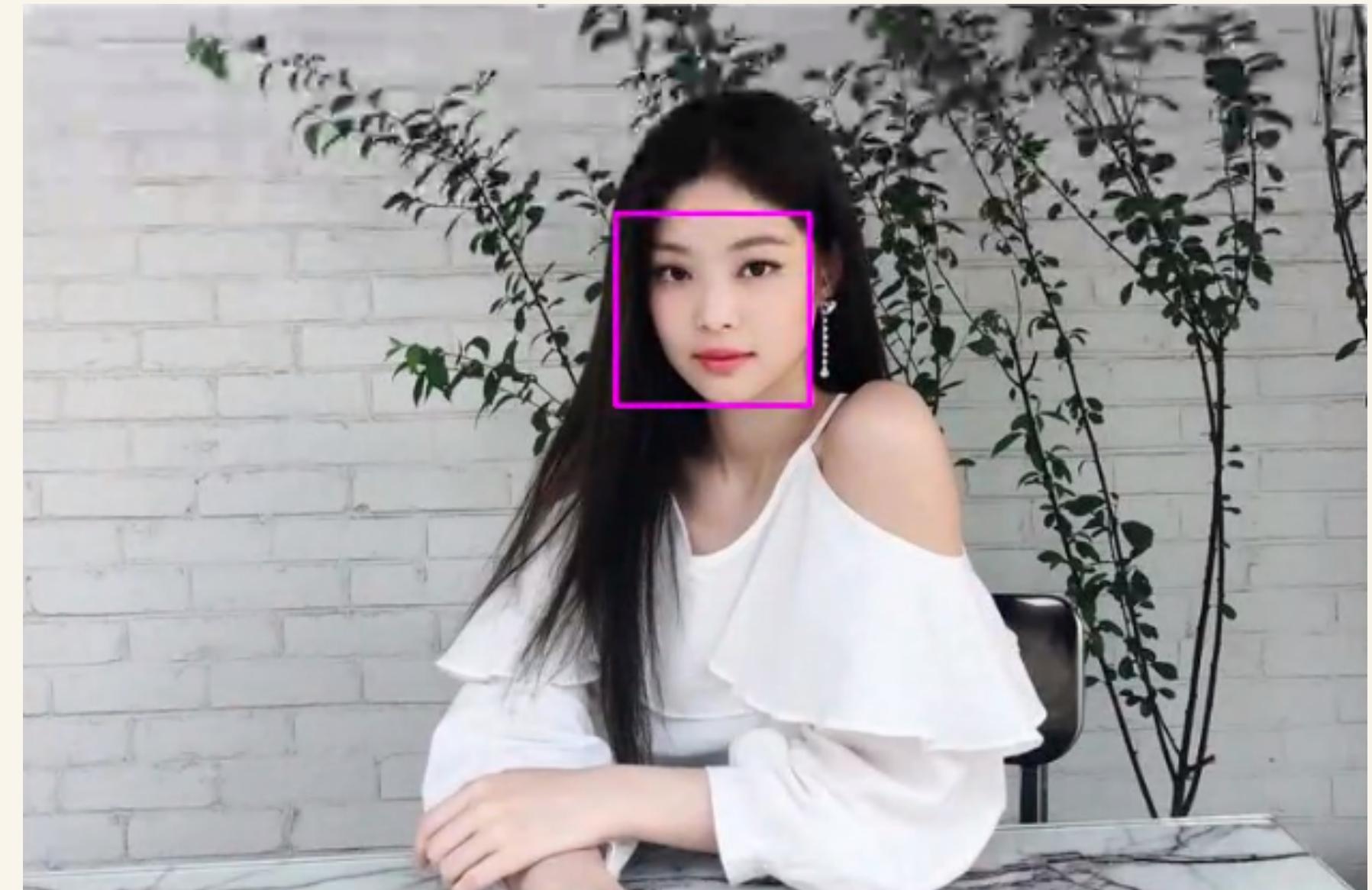
```
faceLocTest = face_recognition.face_locations(imgTest)[0]
encodeTest = face_recognition.face_encodings(imgTest)[0]
cv2.rectangle(imgTest, (faceLocTest[3], faceLocTest[0]),
              (faceLocTest[1], faceLocTest[2]), (255, 0, 255), 2)
```

**cv2.rectangle** → Memvisualisasikan persegi panjang menggunakan koordinat sudut persegi panjang yang diperoleh. Dalam contoh ini, persegi panjang akan memiliki border berwarna ungu, setebal 2 piksel.

# Hasil Deteksi Wajah



imgTrain



imgTest

# Membandingkan Gambar

```
#Ini untuk mengecek apakah wajahnya itu sama atau tidak  
results = face_recognition.compare_faces([encodeTrain], encodeTest)  
#Semakin pendek jaraknya maka semakin mirip mukanya  
faceDis = face_recognition.face_distance([encodeTrain], encodeTest)
```

✓ 0s

▶

```
print(results)  
print(faceDis)
```

[True]  
[0.31549845]

✓ 0s

[31] print(results)  
print(faceDis)

[False]  
[0.78494976]

Pada step ini kita akan membandingkan wajah dari gambar referensi dan gambar test

# Menambah Teks Pada Gambar

```
# memberikan tulisan pada gambar apakah sama(true) atau tidak(false)
# dan jarak kemiripan antara gambar test dengan elon
cv2.putText(imgTest, f'{results} {round(faceDis[0], 2)}',
            (50,50), cv2.FONT_HERSHEY_COMPLEX, 1, (0,0,255), 2)
```

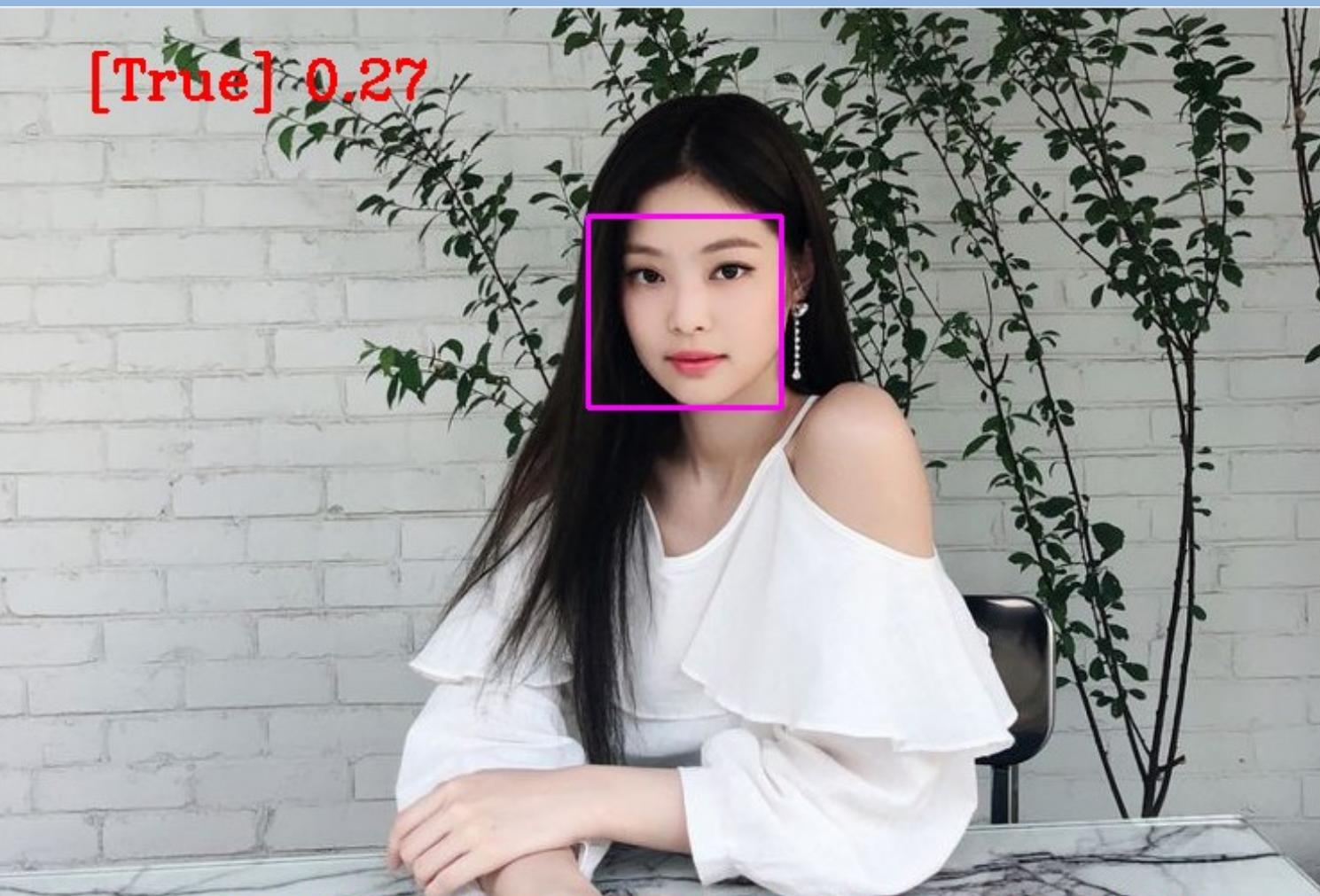
Pada step ini kita akan menambahkan nama dari wajah yang terdetek dan statusnya apakah wajah yang dibandingkan itu sama atau tidak

## Parameter :

- **imgTest** --> gambarnya
- **f'{results} {round(faceDis[0], 2)}** --> hasil dari variable "results" dan "round(faceDis[0], 2)" akan dijadikan isi dari teksnya
- **(50, 50)** --> teks terletak 50 piksel dari tepi kiri dan tepi atas gambar
- **cv2.FONT\_HERSHEY\_COMPLEX** --> font teks = "heresy complex"
- **1** --> ukuran teks
- **(0, 0, 255)** --> warna teks (B,G,R) = merah
- **2** --> ketebalan teks = 2 piksel

# Menampilkan Hasil

```
# menampilkan gambar Train dan test  
cv2.imshow('Train', imgTrain)  
cv2.imshow('Test', imgTest)  
cv2.waitKey(0)
```



Berikut adalah hasil akhir yang akan dihasilkan dari semua step yang telah dilakukan

# PROJECT: ATTENDANCE WITH FACE RECOGNITION



# Mengimport Library dan Membuat Variabel

Pada step awal, terlebih dahulu kita mengimport berbagai library yang dibutuhkan untuk mengerjakan projek ini, serta mendeclare variabel yang akan kita gunakan.

```
import cv2
import numpy as np
import face_recognition
import os
from datetime import datetime
```

Mengimport Library

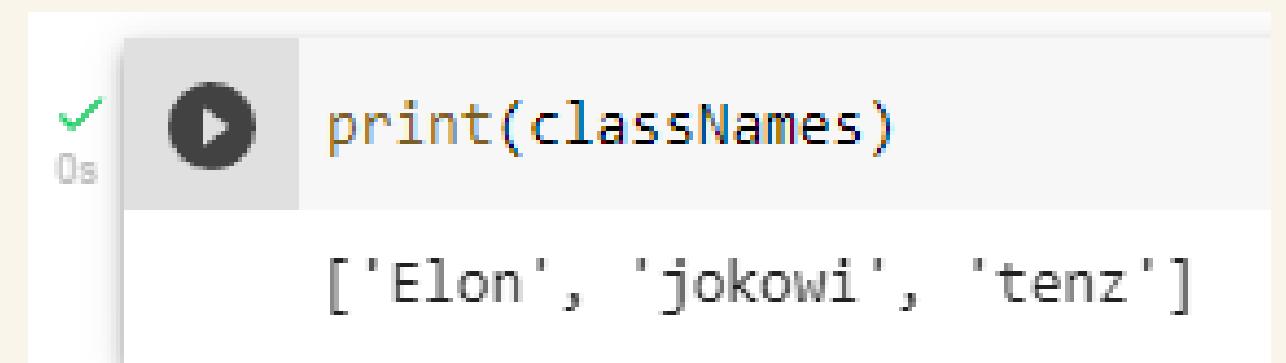


```
#buat nampung gambar
path = 'ImageAttendance'
images = []
classNames = []
myList = os.listdir(path)
```

Mendeclare Variabel

# Memasukkan Semua Gambar ke Dalam Variabel

```
#Masukkan semua gambar dan class ke dalam list
for cls in myList:
    curImg = cv2.imread(f'{path}/{cls}')
    images.append(curImg)
    classNames.append(os.path.splitext(cls[0]))
```



A screenshot of a code editor showing a successful execution of a Python script. The code prints the list of class names: ['Elon', 'jokowi', 'tenz']. The editor interface includes a play button icon and a status bar indicating '0s'.

```
✓ 0s      print(classNames)
      ['Elon', 'jokowi', 'tenz']
```

# Membuat Function yang dibutuhkan

```
#Encoding Function yang menghitung semua encodings dari gambar yang ada
def findEncodings(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList
```

Menyimpan Hasil Encode ke Dalam Suatu Variabel

```
# menyimpan data gambar(nama) kedalam variable encodeListKnown
encodeListKnown = findEncodings(images)
```

Function untuk Encoding



```
# membuat sebuah fungsi yang berfungsi untuk mengambil nama dan waktu untuk ditulis ke dalam suatu file
# dalam kasus ini kami menggunakan file csv atau excel
def markAttendance(name):
    with open('Attendance.csv', 'r+') as f:
        myDataList = f.readlines()
        nameList = []
        for line in myDataList:
            #Untuk ngambil namanya saja
            entry = line.split(',')
            nameList.append(entry[0])

        #Untuk ngambil waktu saat attendance
        if name not in nameList:
            now = datetime.now()
            dtString = now.strftime('%H:%M:%S')
            f.writelines(f'\n{name}, {dtString}')
```

Function untuk memasukkan data (nama dan waktu) ke dalam file

# Mengakses Webcam

Ini adalah cara untuk mengakses dan mengambil gambar secara langsung dari webcam laptop, di mana nantinya muka yang terdeteksi pada webcam akan tercatat kehadirannya

```
#Untuk mengambil image dari webcam  
cap = cv2.VideoCapture(0)
```



# Proses untuk Menampilkan Hasil Akhir dan Memasukkan Data Attendance ke Dalam File

```
# terus melakukan pengecekan selama webcam menyala
while True:
    success, img = cap.read()
    #Mereduce size dari gambar agar proses dapat berjalan lebih cepat
    imgS = cv2.resize(img,(0,0), None, 0.25, 0.25)
    imgS = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    #Mencarri lokasi muka di dalam gambar pada webcam
    faceCurFrame = face_recognition.face_locations(imgS)
    #Mengencode gambar pada webcam
    encodesCurFrame = face_recognition.face_encodings(imgS, faceCurFrame)

# melakukan pengecekan antara gambar dari webcam dengan gambar yang telah kita simpan sebelumnya
for encodeFace, faceLoc in zip(encodesCurFrame, faceCurFrame):
    matches = face_recognition.compare_faces(encodedListKnown, encodeFace)
    faceDis = face_recognition.face_distance(encodedListKnown, encodeFace)

    #Mencari nilai index yang terkecil
    matchIndex = np.argmin(faceDis)

    #Menampilkan nama dari index yang terkecil
    if matches[matchIndex]:
        name = classNames[matchIndex].upper()
        # print(name)
        y1,x2,y2,x1 = faceLoc
        #Untuk menyesuaikan dengan gambar yang telah di resize
        y1,x2,y2,x1 = y1*4,x2*4,y2*4,x1*4
        #Menandai wajah dengan kotak
        cv2.rectangle(img, (x1,y1), (x2,y2), (0,255,0), 2)
        cv2.rectangle(img, (x1,y2-35), (x2,y2), (0,255,0), cv2.FILLED)
        #Menampilkan nama dari wajah yang terdeteksi
        cv2.putText(img, name, (x1+6, y2-6), cv2.FONT_HERSHEY_COMPLEX, 1, (255,255,255), 2)
        #Masukkan data attendace ke csv
        markAttendance(name)

    #Untuk menampilkan webcam
    cv2.imshow('webcam', img)
    cv2.waitKey(1)
```

# Penjelasan Code (1)

```
# terus melakukan pengecekan selama webcam menyala
while True:
    success, img = cap.read()
    #Mereduce size dari gambar agar proses dapat berjalan lebih cepat
    imgS = cv2.resize(img, (0,0), None, 0.25, 0.25)
    imgS = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

Bagian ini berfungsi untuk mendeteksi wajah dan mengencode wajah dari gambar yang diambil oleh webcam

Loop 'While' ini berfungsi untuk terus menjalankan seluruh proses face recognition and attendance selama webcam masih nyala.

cap.read() berfungsi untuk memanggil variabel cap untuk mengakses webcam

cv2.resize digunakan untuk mereduce ukuran gambar yang diambil webcam, lalu cv2.vtColor untuk mengubah format warna dari BGR -> RGB

```
#Mencarri lokasi muka di dalam gambar pada webcam
faceCurFrame = face_recognition.face_locations(imgS)
#Mengencode gambar pada webcam
encodesCurFrame = face_recognition.face_encodings(imgS, faceCurFrame)
```

# Proses untuk Menampilkan Hasil Akhir dan Memasukkan Data Attendance ke Dalam File

```
# terus melakukan pengecekan selama webcam menyala
while True:
    success, img = cap.read()
    #Mereduce size dari gambar agar proses dapat berjalan lebih cepat
    imgS = cv2.resize(img,(0,0), None, 0.25, 0.25)
    imgS = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    #Mencarri lokasi muka di dalam gambar pada webcam
    faceCurFrame = face_recognition.face_locations(imgS)
    #Mengencode gambar pada webcam
    encodesCurFrame = face_recognition.face_encodings(imgS, faceCurFrame)

    # melakukan pengecekan antara gambar dari webcam dengan gambar yang telah kita simpan sebelumnya
    for encodeFace, faceLoc in zip(encodesCurFrame, faceCurFrame):
        matches = face_recognition.compare_faces(encodedListKnown, encodeFace)
        faceDis = face_recognition.face_distance(encodedListKnown, encodeFace)

        #Mencari nilai index yang terkecil
        matchIndex = np.argmin(faceDis)

        #Menampilkan nama dari index yang terkecil
        if matches[matchIndex]:
            name = classNames[matchIndex].upper()
            # print(name)
            y1,x2,y2,x1 = faceLoc
            #Untuk menyesuaikan dengan gambar yang telah di resize
            y1,x2,y2,x1 = y1*4,x2*4,y2*4,x1*4
            #Menandai wajah dengan kotak
            cv2.rectangle(img, (x1,y1), (x2,y2), (0,255,0), 2)
            cv2.rectangle(img, (x1,y2-35), (x2,y2), (0,255,0), cv2.FILLED)
            #Menampilkan nama dari wajah yang terdeteksi
            cv2.putText(img, name, (x1+6, y2-6), cv2.FONT_HERSHEY_COMPLEX, 1, (255,255,255), 2)
            #Masukkan data attendace ke csv
            markAttendance(name)

    #Untuk menampilkan webcam
    cv2.imshow('webcam', img)
    cv2.waitKey(1)
```

# Penjelasan Code (2)

```
# melakukan pengecekan antara gambar dari webcam dengan gambar yang telah kita simpan sebelumnya
for encodeFace, faceLoc in zip(encodesCurFrame, faceCurFrame):
    matches = face_recognition.compare_faces(encodedListKnown, encodeFace)
    faceDis = face_recognition.face_distance(encodedListKnown, encodeFace)

#Mencari nilai index yang terkecil
matchIndex = np.argmin(faceDis)
```

If ini hanya akan menjalankan gambar yang memiliki distance yang paling kecil saja

Memanggil function yang telah dibuat sebelumnya untuk memasukkan data berupa nama dan waktu attendance ke dalam file

Memberi kotak ke wajah yang terdeteksi dan juga menampilkan nama dari wajah yang terdeteksi tersebut (semua hurufnya kapital).

Karena gambar yang diambil oleh webcam dilakukan resize, maka agar kotak dan nama letaknya sesuai, pada tahapan ini semua x dan y dikali dengan 4 (karena gambar di resize 0,25)

Loop 'for' ini berfungsi untuk terus menjalankan seluruh proses sebanyak wajah yang terdeteksi

matchIndex akan menampung nilai faceDis yang paling kecil karena gambar dengan nilai faceDis terkecil adalah gambar yang paling sesuai

```
#Menampilkan nama dari index yang terkecil
if matches[matchIndex]:
    name = classNames[matchIndex].upper()
    # print(name)
    y1,x2,y2,x1 = faceLoc
    #Untuk menyesuaikan dengan gambar yang telah di resize
    y1,x2,y2,x1 = y1*4,x2*4,y2*4,x1*4
    #Menandai wajah dengan kotak
    cv2.rectangle(img, (x1,y1), (x2,y2), (0,255,0), 2)
    cv2.rectangle(img, (x1,y2-35), (x2,y2), (0,255,0), cv2.FILLED)
    #Menampilkan nama dari wajah yang terdeteksi
    cv2.putText(img, name, (x1+6, y2-6), cv2.FONT_HERSHEY_COMPLEX, 1, (255,255,255), 2)
    #Masukkan data attendance ke csv
    markAttendance(name)
```

# Proses untuk Menampilkan Hasil Akhir dan Memasukkan Data Attendance ke Dalam File

```
# terus melakukan pengecekan selama webcam menyala
while True:
    success, img = cap.read()
    #Mereduce size dari gambar agar proses dapat berjalan lebih cepat
    imgS = cv2.resize(img,(0,0), None, 0.25, 0.25)
    imgS = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    #Mencarri lokasi muka di dalam gambar pada webcam
    faceCurFrame = face_recognition.face_locations(imgS)
    #Mengencode gambar pada webcam
    encodesCurFrame = face_recognition.face_encodings(imgS, faceCurFrame)

    # melakukan pengecekan antara gambar dari webcam dengan gambar yang telah kita simpan sebelumnya
    for encodeFace, faceLoc in zip(encodesCurFrame, faceCurFrame):
        matches = face_recognition.compare_faces(encodedListKnown, encodeFace)
        faceDis = face_recognition.face_distance(encodedListKnown, encodeFace)

        #Mencari nilai index yang terkecil
        matchIndex = np.argmin(faceDis)

        #Menampilkan nama dari index yang terkecil
        if matches[matchIndex]:
            name = classNames[matchIndex].upper()
            # print(name)
            y1,x2,y2,x1 = faceLoc
            #Untuk menyesuaikan dengan gambar yang telah di resize
            y1,x2,y2,x1 = y1*4,x2*4,y2*4,x1*4
            #Menandai wajah dengan kotak
            cv2.rectangle(img, (x1,y1), (x2,y2), (0,255,0), 2)
            cv2.rectangle(img, (x1,y2-35), (x2,y2), (0,255,0), cv2.FILLED)
            #Menampilkan nama dari wajah yang terdeteksi
            cv2.putText(img, name, (x1+6, y2-6), cv2.FONT_HERSHEY_COMPLEX, 1, (255,255,255), 2)
            #Masukkan data attendace ke csv
            markAttendance(name)

#Untuk menampilkan webcam
cv2.imshow('webcam', img)
cv2.waitKey(1)
```

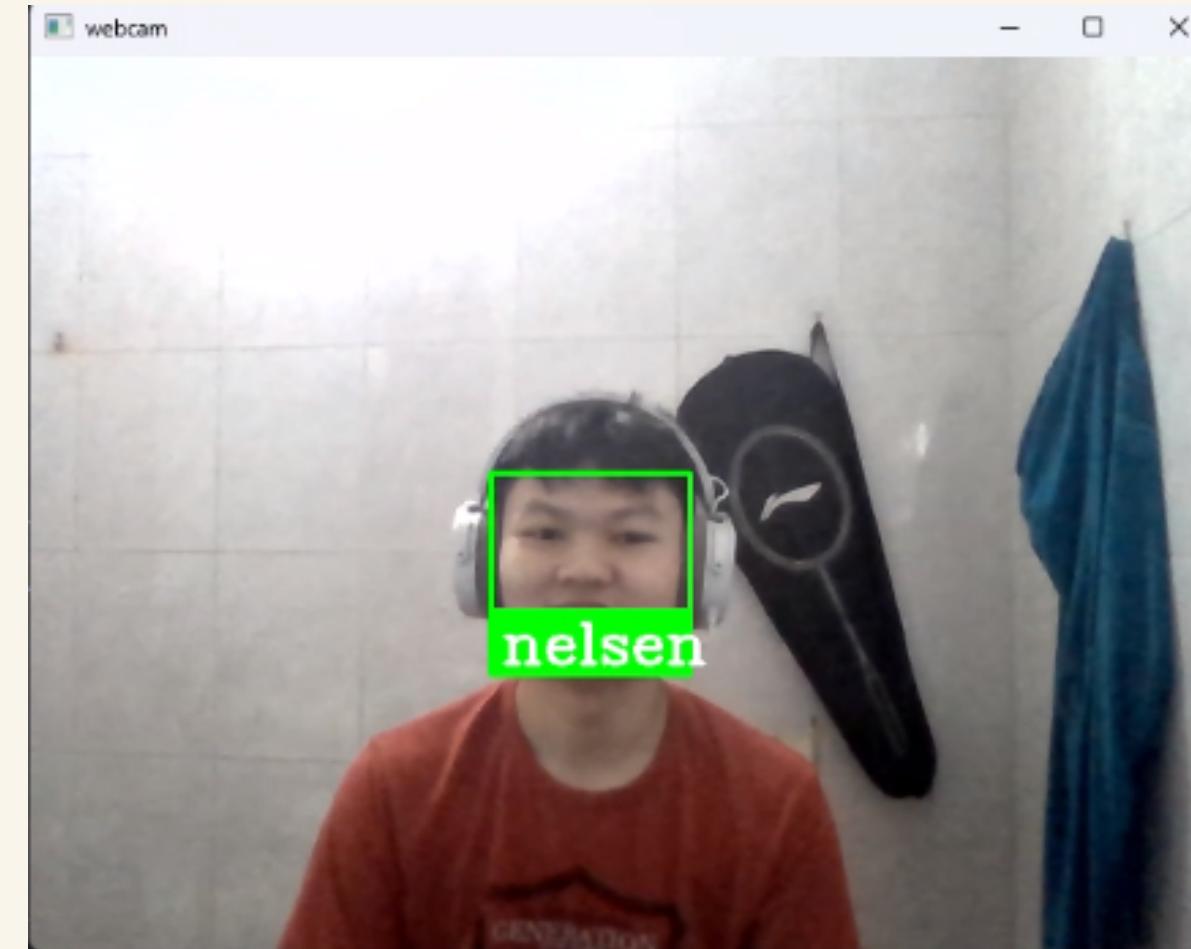
# Penjelasan Code (3)

```
#Untuk menampilkan webcam  
cv2.imshow('webcam', img)  
cv2.waitKey(1)
```

Code ini digunakan untuk menampilkan gambar yang ditangkap dari webcam.



gambar referensi untuk "nelsen"



deteksi muka "nelsen" pada webcam

```
#Masukkan data attendace ke csv  
markAttendance(name)
```

	A	B
1	Name,Time	
2		
3	nelsen, 14:08:21	
4		

pencatatan nama dan waktu deteksi untuk "nelsen"

**TERIMA  
KASIH**