

# **Artificial Intelligence** Pada Pengenalan Wajah

CLASSROOM DEV TEAM

August 19, 2022

# Contents

<b>1</b>	<b>Pengantar</b>	<b>1</b>
1.1	Artificial Intelligence Pada Pengenalan Wajah . . . . .	1
1.2	Face Detection . . . . .	2
1.3	Face Recognition . . . . .	2
<b>2</b>	<b>Model Algoritma Face Recognition Pada OpenCV</b>	<b>4</b>
2.1	Eigenface . . . . .	4
2.2	Fisherfaces . . . . .	5
<b>3</b>	<b>Implementasi Sistem</b>	<b>6</b>
3.1	Proses Instalasi . . . . .	6
3.1.1	Instalasi Python . . . . .	6
3.1.2	Instalasi OpenCV . . . . .	9
3.2	Pembuatan Sistem Face Detection . . . . .	10
3.3	Pembuatan Sistem Face Recognition . . . . .	12

# List of Figures

3.1	Website resmi python . . . . .	6
3.2	Pilihan Versi Python . . . . .	7
3.3	Instalator python . . . . .	7
3.4	Pilihan fitur . . . . .	8
3.5	Pilihan lanjutan dan penyesuaian lokasi . . . . .	8
3.6	Instalasi openCV . . . . .	9
3.7	Cek openCV pada python IDLE . . . . .	9
3.8	Cek openCV pada CMD . . . . .	9
3.9	Memasukan library openCV . . . . .	10
3.10	Memasukan library openCV . . . . .	10
3.11	Memasukan video . . . . .	10
3.12	Membuka video dan ubah warna cira . . . . .	10
3.13	Deteksi wajah . . . . .	11
3.14	Kode bingkai wajah . . . . .	11
3.15	Kode deteksi wajah . . . . .	11
3.16	Hasil deteksi wajah . . . . .	11

# 1

## Pengantar

### 1.1 Artificial Intelligence Pada Pengenalan Wajah

Dilansir dari Stanford Computer science, Artificial Intelligence(AI) atau kecerdasan buatan adalah ilmu dan rekayasa pembuatan mesin cerdas, melibatkan mekanisme untuk menjalankan suatu tugas menggunakan komputer. Sehingga artificial intelligence merupakan sebuah teknologi yang memungkinkan sistem komputer, perangkat lunak, program dan robot untuk “berpikir” secara cerdas layaknya manusia. Kecerdasan buatan suatu mesin dibuat oleh manusia melalui algoritma pemrograman yang kompleks.<sup>1</sup>

Secara garis besar, AI dapat melakukan salah satu dari keempat faktor berikut:

- a. *Acting Humanly* , sistem bertindak layaknya manusia.
- b. *Thinking Humanly* , sistem dapat berpikir seperti manusia.
- c. *Think Rationally* , sistem dapat berpikir secara rasional.
- d. *Act Rationally* , sistem mampu bertindak secara rasional.

Pengenalan dan identifikasi wajah merupakan contoh sistem penerapan konsep Artificial Intelligence menggunakan biometrik wajah yang terus berkembang pada bidang *computer vision*. Kecerdasan buatan ini digunakan secara *real-time* untuk menangkap dan mengenali wajah seseorang pada kamera.

Computer Vision adalah bagaimana komputer/mesin dapat melihat, teknik computer vision mampu memvisualisasikan data menganalisis berupa gambar image atau dalam bentuk video. Tujuan utama dari Computer Vision adalah

---

<sup>1</sup>Mustofa, Zaenal. *Artificial Intelligence (AI): Pengertian, Perkembangan, Cara Kerja, Dan Dampaknya*. Universitas STEKOM

agar komputer atau mesin dapat meniru kemampuan perseptual mata manusia dan otak, atau bahkan dapat mengunggulinya untuk tujuan tertentu.<sup>2</sup>

## 1.2 Face Detection

*Face Detection* atau pengenalan wajah merupakan sebuah teknologi untuk menangkap wajah seseorang pada kamera yang menjadi tahap awal dalam sistem pengenalan wajah (*Face Recognition*) yang digunakan dalam identifikasi biometrik. Deteksi wajah juga dapat digunakan untuk pencarian atau pengindeksan data wajah dari citra atau video yang berisi wajah dengan berbagai ukuran, posisi, dan latar belakang.<sup>3</sup>

Pembuatan pendeteksi wajah ini dapat dibuat menggunakan openCV yang merupakan aplikasi perangkat lunak untuk pengolahan citra dinamis secara *real-time*, selain itu openCV juga banyak mendukung bahasa pemrograman diantaranya C++, C, python, dan java. Pada pembahasan kali ini, penjelasan mengenai proses pembuatan deteksi wajah akan menggunakan openCV dengan bahasa pemrograman python. Proses deteksi objek maupun wajah dapat menggunakan metode algoritma Haar Cascade Classifier.

Algoritma Haar Cascade Classifier merupakan salah satu algoritma yang digunakan untuk mendeteksi sebuah wajah dengan cepat dan *real-time* sebuah benda termasuk wajah manusia. Metode ini menggunakan haar-like features dimana perlu dilakukan training terlebih dahulu untuk mendapatkan suatu pohon keputusan dengan nama cascade classifier sebagai penentu apakah ada obyek atau tidak dalam frame yang diproses, dengan mengelompokkan fitur-fitur pada gambaran wajah berdasarkan sisi yang terang dan sisi yang gelap. Adanya fitur Haar ditentukan dengan cara mengurangi rata-rata piksel pada daerah gelap dari rata-rata piksel pada daerah terang<sup>4</sup>

## 1.3 Face Recognition

Face recognition adalah sebuah teknologi yang mampu untuk mengidentifikasi dan mengkonfirmasi identitas seseorang menggunakan wajah mereka. Face recognition menjadi salah satu sistem identifikasi biometrik yang paling baik dalam mengidentifikasi seseorang dengan fitur-fitur khusus pada tubuh maupun DNA yang menjadi pembeda antara satu orang dengan orang lainnya.

---

<sup>2</sup>Wibowo, Ari. *Implementasi Teknik Computer Vision*. Universitas Widyatama

<sup>3</sup>NUGROHO, Setyo, Drs. Agus Hardjoko, MSc., PhD. *Sistem pendeteksi wajah manusia pada citra digital*, Universitas Gajah mada, diakses dari <http://etd.repository.ugm.ac.id/penelitian/detail/23416>

<sup>4</sup>Suhepy Abidin. *Deteksi Wajah Menggunakan Metode Haar Cascade Classifier Berbasis Webcam Pada Matlab*. Jurusan Teknik Elektro, Politeknik Negeri Ujung Pandang

Menurut US Government Accountability Office, ada 4 komponen yang dibutuhkan untuk melakukan face recognition, yaitu: kamera, faceprint, Database dan terakhir Algoritme untuk membandingkan faceprint dari wajah target dengan faceprint dalam database.<sup>5</sup> Setelah terpenuhinya komponen tersebut, dilakukan beberapa tahap untuk melakukan face recognition.

Menurut Haisong Gu, Qiang Ji, dan Zhiwei Zu (2002), pengenalan wajah umumnya melalui 3 tahapan untuk mendapatkan hasil.

1. *Face Detection*, dilakukan untuk mendeteksi adanya atau tidak pada frame yang dibaca sistem. Pada proses ini menggunakan metode *Haar-cascade Classifier*
2. *Facial Expression Information Extraction*, dilakukan pada wajah yang sudah terdeteksi untuk mengekstraksi informasi penting yang akan didapatkan dari wajah untuk membedakan wajah. Fitur atau bagian wajah yang telah diekstraksi nantinya akan digunakan untuk pencocokan wajah.
3. *Expression Classification*, merupakan proses akhir dalam pengenalan wajah, dimana sistem melakukan pencocokan dari masukan wajah dengan data yang ada pada database

Ada beberapa algoritma untuk melakukan pencocokan pada proses pengenalan wajah yang disediakan oleh openCV. Local Binary Pattern Histogram (LBPH) adalah salah satu dari tiga algoritma pengenalan wajah bawaan pada library OpenCV antara lain Eigenface, Fisherfaces, dan LBPH. Dibandingkan dengan kedua algoritma tersebut, LBPH tidak hanya dapat mengenali muka depan, tetapi juga mengenali muka samping yang lebih fleksibel. Metode ini bekerja dengan membandingkan dan mencocokkan histogram yang sudah diekstraksi dengan citra wajah yang sudah ada pada database/dataset.

---

<sup>5</sup>Putri, Monica. *Cara Kerja Face Recognition*. Universitas Binus

## 2

# Model Algoritma Face Recognition Pada OpenCV

Ada beberapa model algoritma yang sudah disediakan openCV untuk melakukan pengenalan wajah, diantaranya adalah Eigenface, Fisherfaces, dan *Local binary Pattern Histogram*(LBPH). Berikut penjelasan untuk perbedaan dari ketiga model algoritma yang disediakan oleh openCV.

## 2.1 Eigenface

Eigenface adalah salah satu algoritma pengenalan wajah yang berdasarkan pada Principle Component Analysis (PCA) yang dikembangkan di MIT. Algoritma EigenFace secara keseluruhan cukup sederhana, Training Image direpresentasikan dalam sebuah vektor flat (gabungan vektor) dan digabung bersama-sama menjadi sebuah matriks tunggal. Eigen Vector kemudian diekstraksi dan disimpan dalam file temporary atau database. Training image kemudian diproyeksikan dalam feature space yang di namai face space yang ditentukan oleh eigen vektor(Mukti, 2008).<sup>1</sup>

Principal Component Analysis (PCA) atau disebut juga transformasi KarhunenLoeve adalah teknik yang digunakan untuk menyederhanakan suatu data, dengan cara mentransormasi linear sehingga terbentuk system koordinat baru dengan variansi maksimum. PCA dapat digunakan untuk mereduksi dimensi suatu data tanpa mengurangi karakteristik data tersebut secara signifikan (Cahyadi, 2007: 93) <sup>2</sup>

---

<sup>1</sup>Alam, R.G guntur, dkk.*MPLEMENTASI ALGORITMA EIGENFACE UNTUK FACE RECOGNITION PADA OBJECK FOTO ID CARD*.Telematik : Vol 7, No 2, April 2015.

<sup>2</sup>Firliana, Lina, dkk.*Implementasi Principal Component Analysis (PCA) Untuk Pengenalan Wajah Manusia*. Universitas Nusantara

## 2. MODEL ALGORITMA FACE RECOGNITION PADA OPENCV 5

Berikut proses langkah-langkah pengenalan wajah menggunakan model algoritma Eigenface.

1. Penyusunan flat vektor matriks
2. Mengambil nilai tengah dari kumpulan matriks
3. Menghitung selisih antara nilai matriks *training image* dengan nilai tengah
4. Menghitung nilai matriks kovarian
5. Menghitung nilai *eigenvalue* dan *eigenvector*
6. Menghitung nilai *eigenface*
7. Proses indentifikasi wajah

### 2.2 Fisherfaces



## 3

# Implementasi Sistem

## 3.1 Proses Instalasi

Pembuatan sistem face detection dan face recognition akan menggunakan bahasa pemrograman Python dan *library* openCV. Berikut adalah cara instalasi Python serta library openCV yang akan digunakan.

### 3.1.1 Instalasi Python

Instalator Python dapat didownload pada website resmi python <https://www.python.org/downloads>

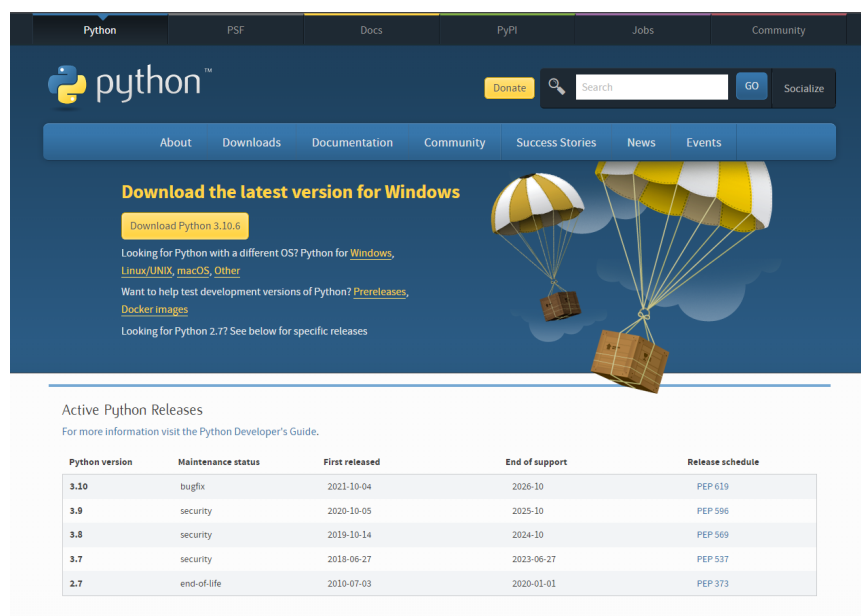


Figure 3.1: Website resmi python

Download instalator versi terbaru dari python atau sesuaikan dengan kebutuhan penggunaan

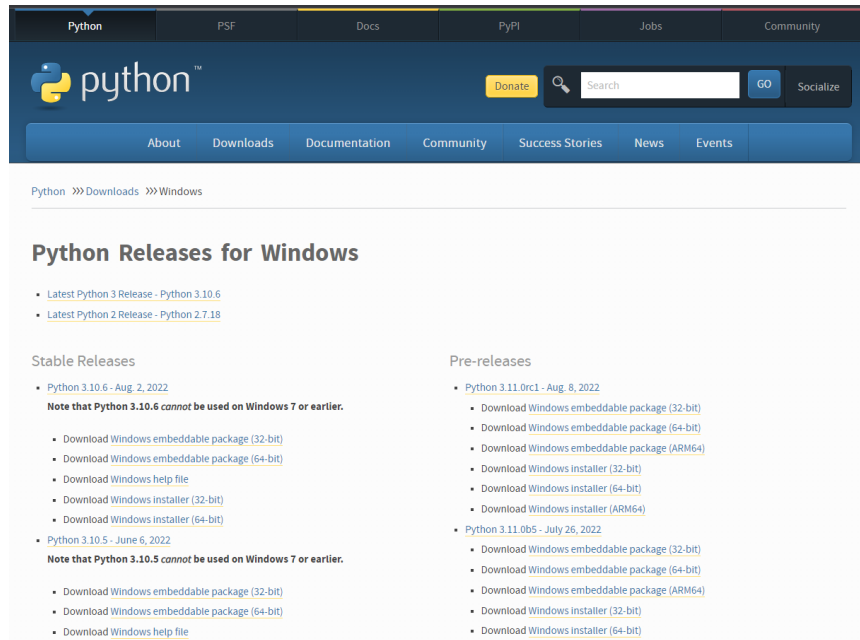


Figure 3.2: Pilihan Versi Python

Kemudian, buka file instalator python yang telah didownload, centang "Add Python 3.10 to PATH" dan klik *Customize Installation*

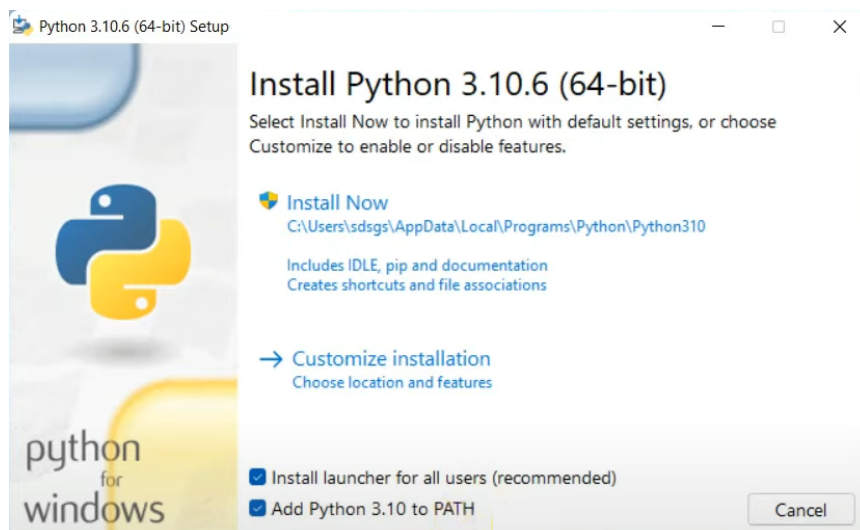


Figure 3.3: Instalator python

Pilih fitur yang akan digunakan, untuk saran pilih semua fitur agar instalasi python lengkap, lalu klik 'next'

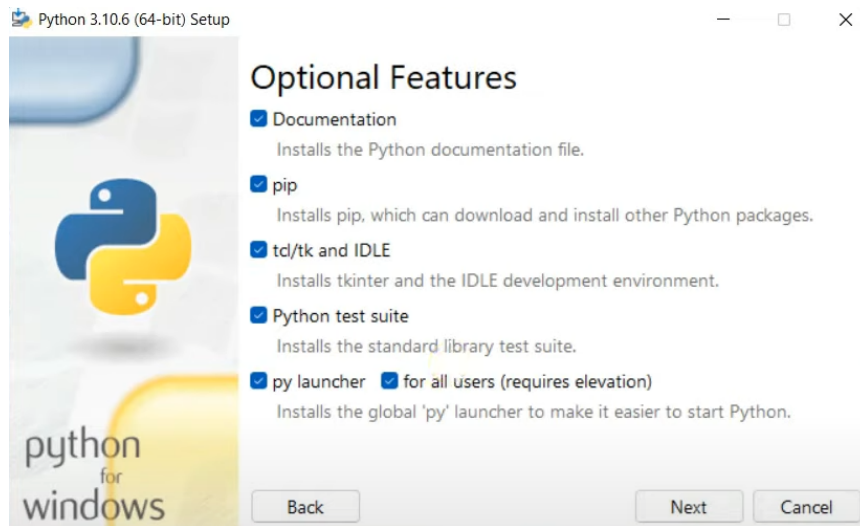


Figure 3.4: Pilihan fitur

Centang pilihan sesuai pada gambar, kemudian pilih direktori untuk lokasi penyimpanan instalasi python sesuai kebutuhan. Lalu klik "Install" dan tunggu hingga proses instalasi selesai.

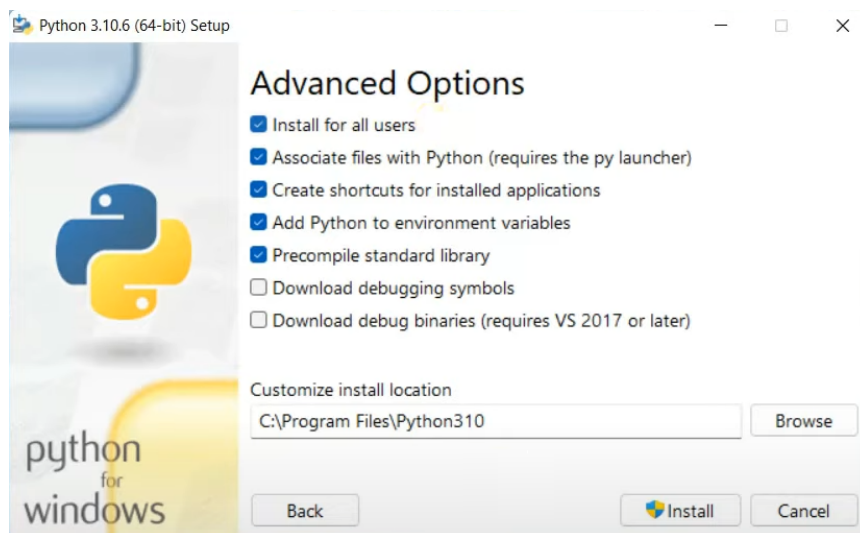


Figure 3.5: Pilihan lanjutan dan penyesuaian lokasi

### 3.1.2 Instalasi OpenCV

Untuk Instalasi openCV dapat dilakukan melalui CMD( *Command Prompt*). Buka direktori penyimpanan instalasi python, lalu menuju direktori 'Scripts' tempat pip.exe berada. Lalu tuliskan perintah *pip install opencv-contrib-python* untuk memulai instalasi openCV, tunggu instalasi hingga selesai.

```
Scripts> pip install opencv-contrib-python
```

Figure 3.6: Instalasi openCV

Setelah instalasi selesai, buka python IDLE atau pada CMD didalam direktori instalasi python, buka python.

Setelah python terbuka, ketik **import cv2** lalu enter, jika saat pengecekan openCV pada python tidak terjadi error, maka openCV berhasil diinstall.

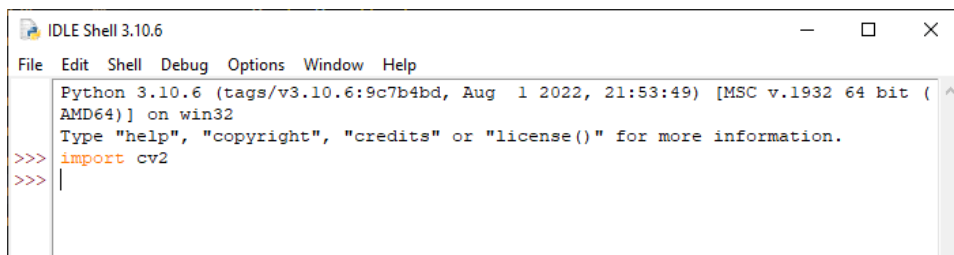


Figure 3.7: Cek openCV pada python IDLE

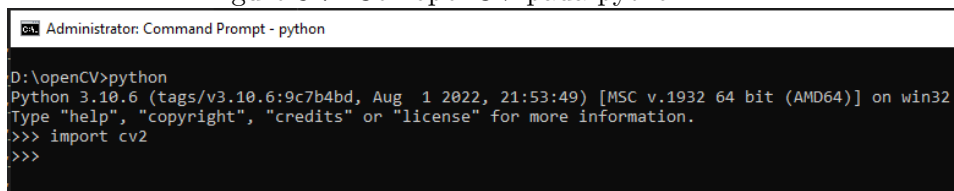


Figure 3.8: Cek openCV pada CMD

## 3.2 Pembuatan Sistem Face Detection

Pembuatan sistem Face Detection kali ini menggunakan algoritma Haar Cascade Classifier. Proses pertama yang dilakukan adalah dengan mengubah citra warna menjadi citra *grayscale*, selanjutnya melakukan pemindaian pada citra *grayscale* untuk mendapatkan nilai fitur citra dengan *Haar-Like Feature* yang menyatakan objek wajah.

Berikut ini merupakan bagian-bagian untuk membuat sistem face detection atau pendeteksi wajah.

- Memasukan library, **cv2** = merupakan *library* openCV

```
import cv2
```

Figure 3.9: Memasukan library openCV

- Proses face detection, untuk melakukan proses deteksi wajah akan menggunakan algoritma *haarcascade*. Dengan fungsi **cv2.CascadeClassifier**

```
faceDetect = cv2.CascadeClassifier(  
    'haarcascade_frontalface_default.xml')
```

Figure 3.10: Memasukan library openCV

- Memasukan video, **cv2.VideoCapture** merupakan fungsi untuk membaca video yang akan dijadikan *sample* dan bisa juga untuk menampilkan frame kamera yang terkoneksi dengan komputer.

```
cap = cv2.VideoCapture('data/video_test.mp4')
```

Figure 3.11: Memasukan video

- Membuka video/kamera dan merubah warna. Kode pada baris pertama digunakan untuk membuka video atau kamera yang terhubung, baris kedua merupakan kode untuk mengubah warna citra menjadi hitam putih menggunakan **cv2.cvtColor**

```
_, frame = cap.read()  
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

Figure 3.12: Membuka video dan ubah warna cira

- Fungsi untuk mendeteksi wajah, menggunakan fungsi **detectMultiScale**

```
face = faceDetect.detectMultiScale(gray, 1.3, 5);
```

Figure 3.13: Deteksi wajah

- Fungsi untuk membuat bingkai tanda jika wajah terdeteksi oleh sistem menggunakan **cv2.rectangle** jika berbentuk persegi. Dan fungsi **cv2.imshow** untuk menjalankan sistem yang telah dibuat.

```
for(x,y,w,h) in face:
    cv2.rectangle(frame, (x,y), (x+w, y+h), (255,0,0),2)
cv2.imshow('face', frame)
```

Figure 3.14: Kode bingkai wajah

Berikut adalah seluruh kode dan hasil untuk sistem deteksi wajah:

```
import cv2

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

cap = cv2.VideoCapture('data/video_test.mp4')
while cap.isOpened():
    _, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    face = faceDetect.detectMultiScale(gray, 1.3, 5);
    for(x,y,w,h) in face:
        cv2.rectangle(frame, (x,y), (x+w, y+h), (255,0,0),2)

    cv2.imshow('face', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

Figure 3.15: Kode deteksi wajah

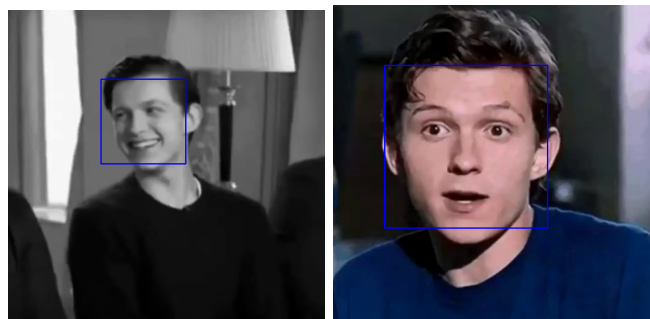


Figure 3.16: Hasil deteksi wajah

### 3.3 Pembuatan Sistem Face Recognition

Prose pertama untuk melakukan pengenalan wajah yaitu dengan mengumpulkan dataset yang akan di *training* dengan menangkap citra wajah pada saat awal deteksi wajah yang akan disimpan dan dikumpulkan berdasarkan id yang telah dimasukkan user. Setelah dataset terkumpul, selanjutnya sistem akan melakukan *training data* untuk mengenali wajah berdasarkan id. Kemudian proses penenalan wajah pun dilakukan dengan mendeteksi wajah menggunakan algoritma Haar-cascade classifier, lalu sistem akan melakukan pencocokan dengan menggunakan fitur LBPH untuk mencocokkan wajah yang terdeteksi dengan dataset yang sudah di *training* sebelumnya.

- Mengumpulkan dataset dengan metode *Haar-cascade classifier*
- Proses training dataset
- Proses pengenalan wajah dengan metode *Local Binary Patter Histogram (LBPH)*
- Testing aplikasi