

Artificial Intelligence

Pada Pengenalan Wajah

August 15, 2022

Contents

1	Pengantar	1
1.1	Artificial Intelligence Pada Pengenalan Wajah	1
1.2	Face Detection	2
1.3	Face Recognition	2
2	Implementasi Sistem	3
2.1	Proses Instalasi	3
2.1.1	Instalasi Python	3
2.1.2	Instalasi OpenCV	6
2.2	Pembuatan Sistem Face Detection	7

List of Figures

2.1	Website resmi python	3
2.2	Pilihan Versi Python	4
2.3	Instalator python	4
2.4	Pilihan fitur	5
2.5	Pilihan lanjutan dan penyesuaian lokasi	5
2.6	Instalasi openCV	6
2.7	Cek openCV pada python IDLE	6
2.8	Cek openCV pada CMD	6
2.9	Memasukan library openCV	7
2.10	Memasukan library openCV	7
2.11	Memasukan video	7
2.12	Membuka video	7
2.13	Deteksi wajah	8
2.14	Kode bingkai wajah	8
2.15	Kode deteksi wajah	8
2.16	Hasil deteksi wajah	8

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.¹

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

- *Acting Humanly* , sistem bertindak layaknya manusia.
- *Thinking Humanly* , sistem dapat berpikir seperti manusia.
- *Think Rationally* , sistem dapat berpikir secara rasional.
- *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 agar komputer atau mesin dapat meniru kemampuan perseptual mata manusia dan otak, atau bahkan dapat mengunggulinya untuk tujuan tertentu.²

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

²Wibowo, Ari. *Implementasi Teknik Computer Vision*. Universitas Widyatama

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.³

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 terang⁴

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. 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.⁵

³NUGROHO, 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>

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

⁵Putri, Monica. *Cara Kerja Face Recognition*. Universitas Binus

2

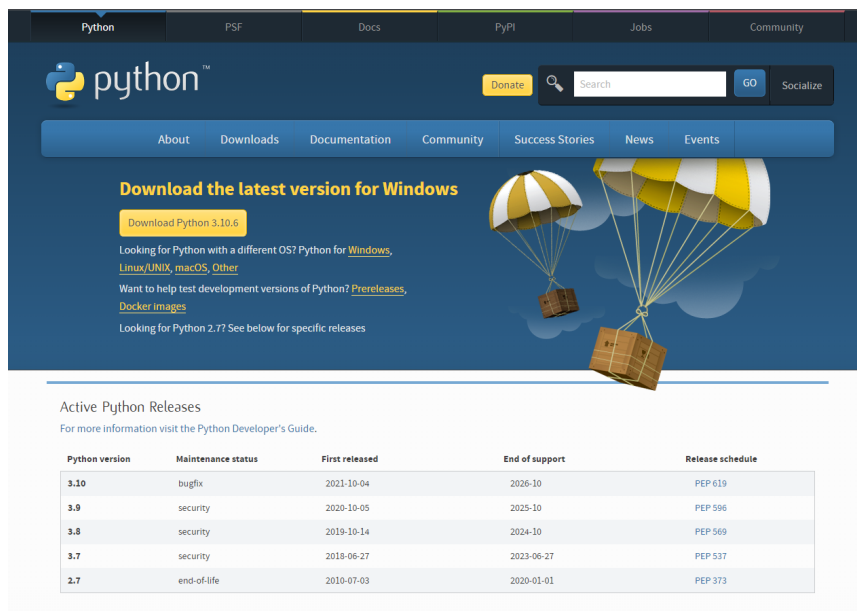
Implementasi Sistem

2.1 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.

2.1.1 Instalasi Python

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



The screenshot shows the official Python website. The top navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. Below this is a search bar and a 'Donate' button. The main content area features a large banner with the text 'Download the latest version for Windows' and a button to 'Download Python 3.10.6'. To the right of the banner is an illustration of two parachutes carrying boxes. Below the banner, there is a section titled 'Active Python Releases' with a table listing the versions, their maintenance status, release dates, end of support dates, and release schedules.

Python version	Maintenance status	First released	End of support	Release schedule
3.10	bugfix	2021-10-04	2026-10	PEP 619
3.9	security	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569
3.7	security	2019-06-27	2023-06-27	PEP 537
2.7	end-of-life	2010-07-03	2020-01-01	PEP 373

Figure 2.1: Website resmi python

Download instalator versi terbaru dari python atau sesuaikan dengan kebutuhan penggunaan

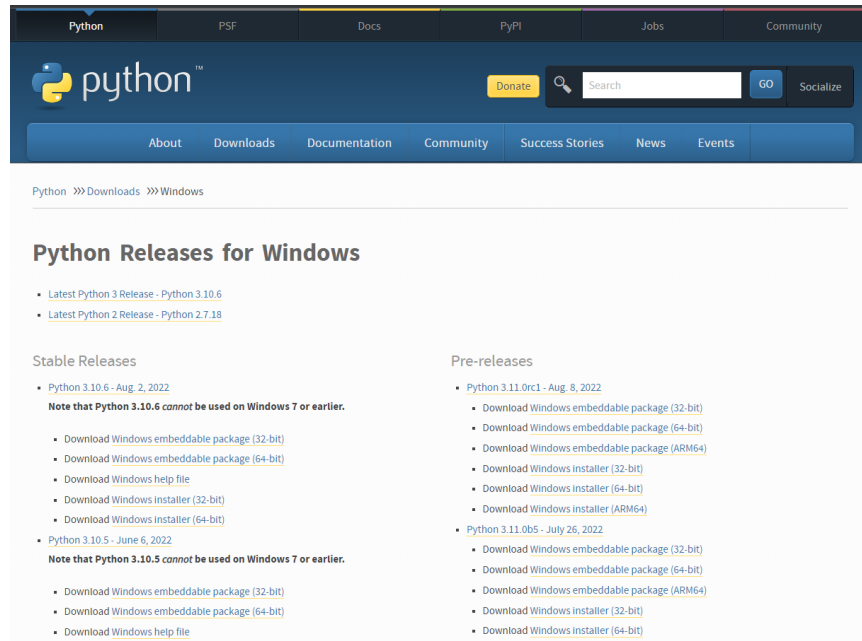


Figure 2.2: Pilihan Versi Python

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



Figure 2.3: Instalator python

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

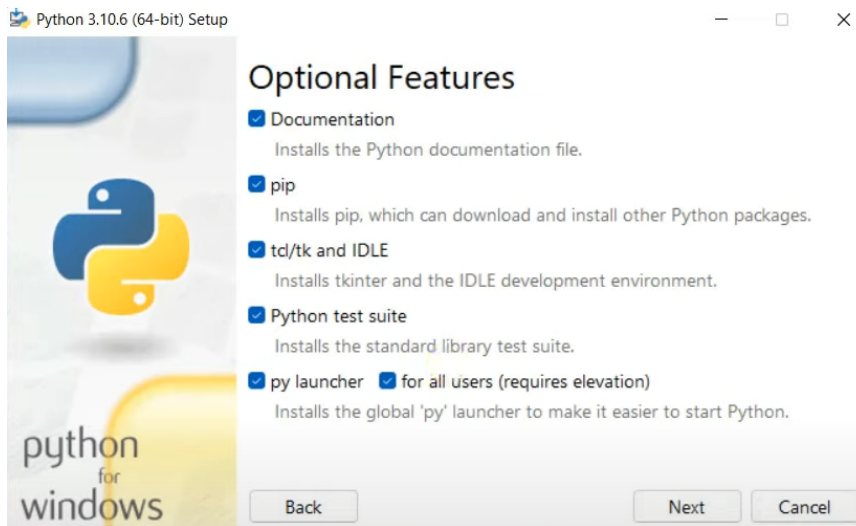


Figure 2.4: Pilihan fitur

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

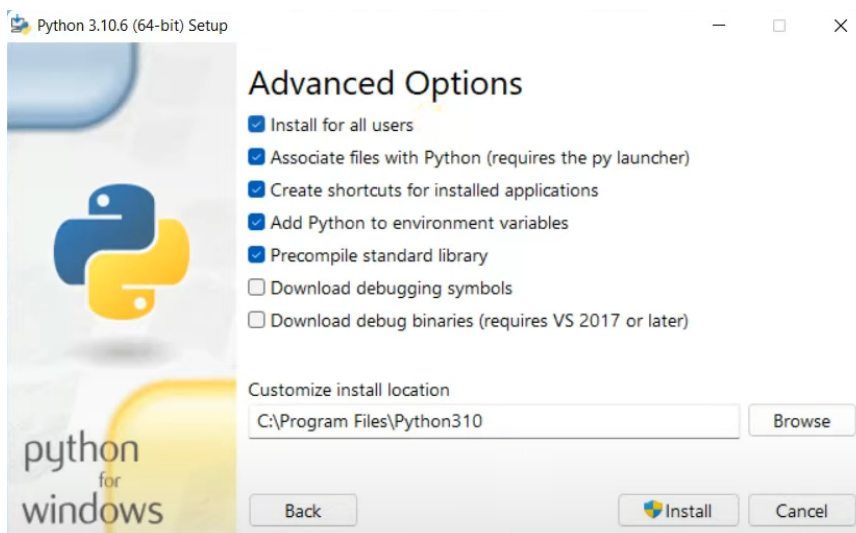


Figure 2.5: Pilihan lanjutan dan penyesuaian lokasi

2.1.2 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 2.6: 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.

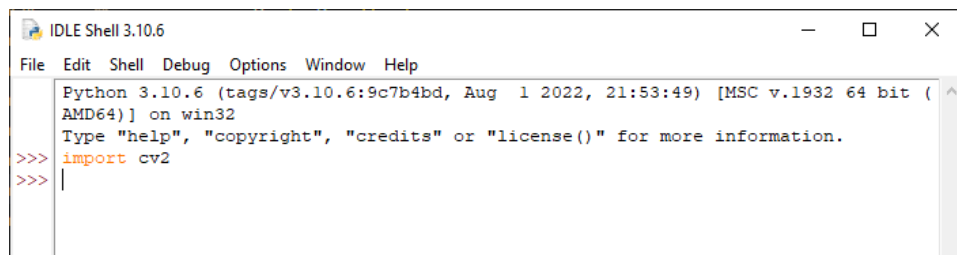


Figure 2.7: Cek openCV pada python IDLE

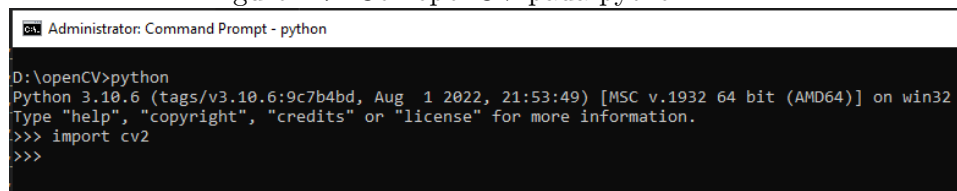


Figure 2.8: Cek openCV pada CMD

2.2 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 2.9: 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 2.10: 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.

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

Figure 2.11: 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 2.12: Membuka video

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

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

Figure 2.13: 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 2.14: 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 2.15: Kode deteksi wajah

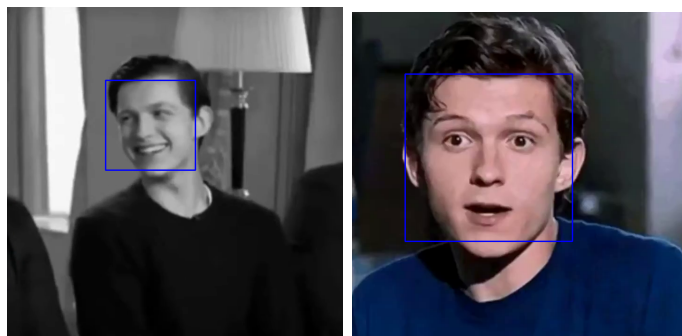


Figure 2.16: Hasil deteksi wajah