



D3TI-2023

INTERNET OF THINGS

SISTEM KEHADIRAN DENGAN PENGENALAN WAJAH PEOPLE IOT - 2024

Prepared by :

11323006 - Cheryl M Lovica

11323051 - Vinci G Baringbing

11323055 - Febyanti Hutahaeen



[github.com/IOT-D3TI23/People IoT FINAL](https://github.com/IOT-D3TI23/People_IoT_FINAL)



Introduction

Sistem Kehadiran dengan Pengenalalan Wajah

Definsi

Sistem Kehadiran dengan Pengenalalan Wajah adalah aplikasi berbasis Python yang memanfaatkan pustaka OpenCV untuk mendeteksi dan mengenali wajah. Hal ini sangat mudah dilakukan manusia tetapi komputer membutuhkan instruksi yang tepat. Sistem ini bekerja dengan mencocokkan wajah mahasiswa yang di-scan secara langsung dengan gambar yang sudah dilatih sebelumnya menggunakan trainer image. Tujuannya adalah membuat proses absensi jadi lebih praktis, cepat, dan akurat, sekaligus membantu dosen dalam melacak kehadiran mahasiswa secara otomatis tanpa perlu metode manual yang memakan waktu.

Perangkat Lunak yang digunakan

- ➔ Bahasa Pemograman : Python 3
- ➔ library : openCv

Keuntungan

- ➔ Sistem menyimpan wajah yang terdeteksi dan secara otomatis menandai kehadiran



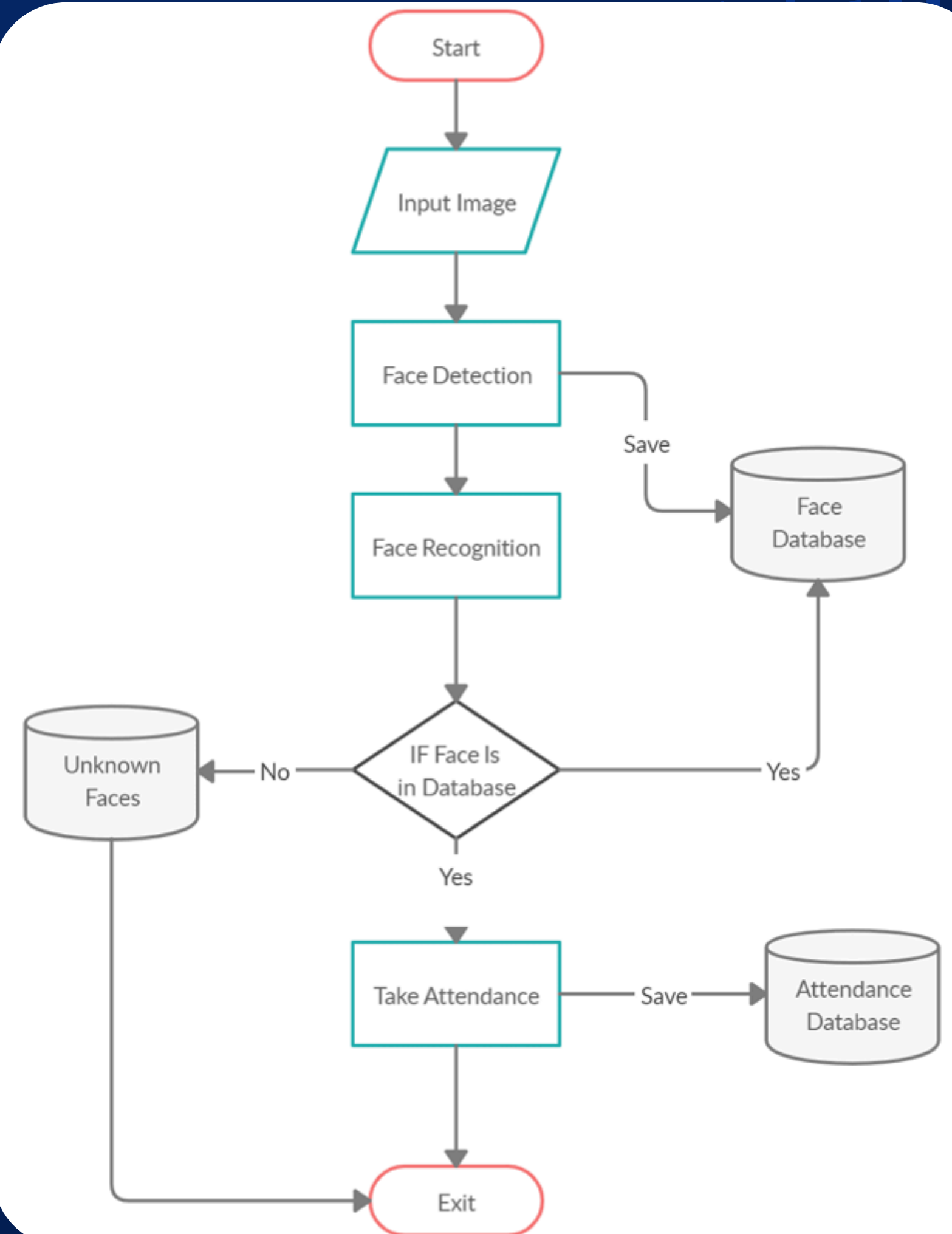
Kelemahan

- ➔ Keakuratan sistem ini tidak 100%. Sistem ini hanya dapat mendeteksi wajah dari jarak yang terbatas



Cara Kerja Sistem

1. Proses dimulai dengan sistem menerima input gambar wajah, (team kami menggunakan kamera diikuti lib openCv)
2. Setelah gambar diterima, sistem melanjutkan ke tahap pendeteksian wajah (Face Detection) untuk memastikan keberadaan wajah di dalam gambar.
3. Jika wajah terdeteksi, proses dilanjutkan ke pengenalan wajah (Face Recognition) untuk mencocokkan wajah tersebut dengan data yang telah tersimpan di dalam Face Database.
4. Pada tahap berikutnya, sistem melakukan pengecekan apakah wajah yang dikenali ada di dalam database. Jika wajah tersebut ditemukan, maka sistem mencatat kehadiran pengguna tersebut ke dalam Attendance Database.
5. Namun, jika wajah tidak ditemukan di database, wajah tersebut digolongkan sebagai Unknown Faces, yang memungkinkan wajah baru disimpan untuk keperluan pendaftaran di masa depan.
6. Setelah proses selesai, sistem keluar dan siap untuk melakukan pengenalan wajah berikutnya.



Kerja Algoritma LBPH

Lbph adalah salah satu metode populer dalam pengenalan wajah yang bekerja dengan mengekstrak fitur tekstur lokal dari gambar wajah.

Untuk melatih algoritma nya:

LBPH menggunakan 3 parameter :

1. radius, yang digunakan untuk membangun pola biner lokal melingkar dan mewakili radius disekitar piksel pusat.
2. grid x, jumlah sel pada arah horizontal. semakin banyak sel semakin halus kisi-kisi, semakin tinggi dimensi vektor yang dihasilkan
3. grid y, jumlah sel pada arah vertikal. semakin banyak sel semakin halus kisi-kisi, semakin tinggi dimensi vektor yang dihasilkan

Untuk melakukannya, kita perlu menggunakan dataset dengan gambar wajah orang yang ingin kita kenali. Kita juga perlu menetapkan ID (bisa berupa nomor atau nama orang) untuk setiap gambar, sehingga algoritme akan menggunakan informasi ini untuk mengenali gambar Input dan memberikan output. Gambar dari orang yang sama harus memiliki ID yang sama. Dengan set pelatihan yang telah dibuat, mari kita lihat langkah-langkah komputasi LBPH.

How to run code

```
python --version
```

```
pip install opencv-contrib-python
```

```
python
```

```
import cv2
```

```
print(cv2.__version__)
```

```
exit()
```

```
python
```

```
pip install pillow
```

```
from PIL import Image
```

```
print("Pillow is installed!")
```

```
exit()
```

```
python
```

```
pip install pandas
```

```
import pandas as pd
```

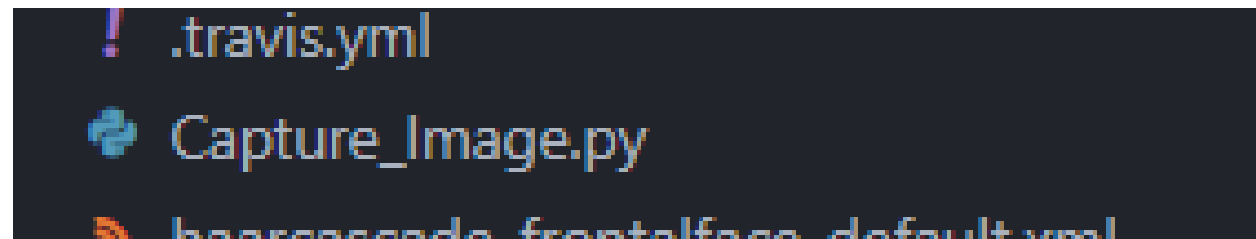
```
print(pd.__version__)
```

```
exit()
```

```
python main.py
```

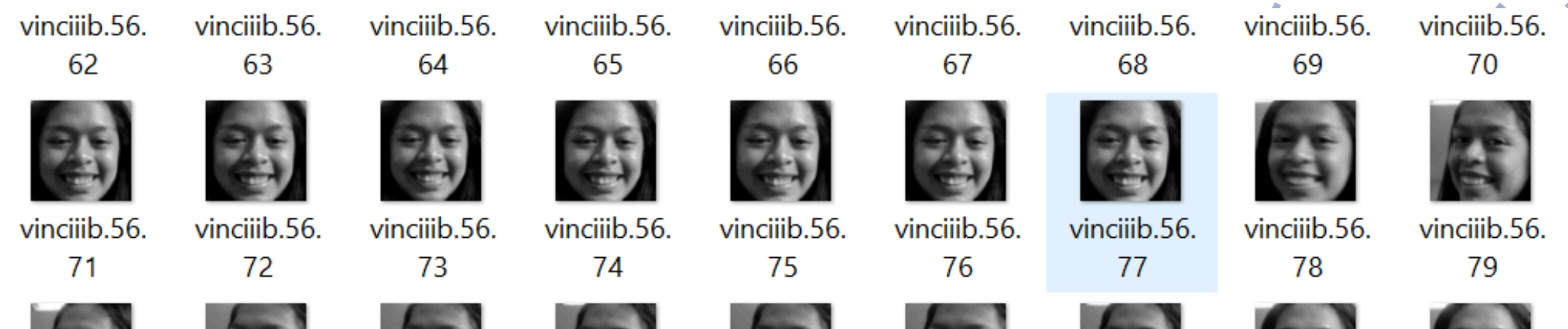
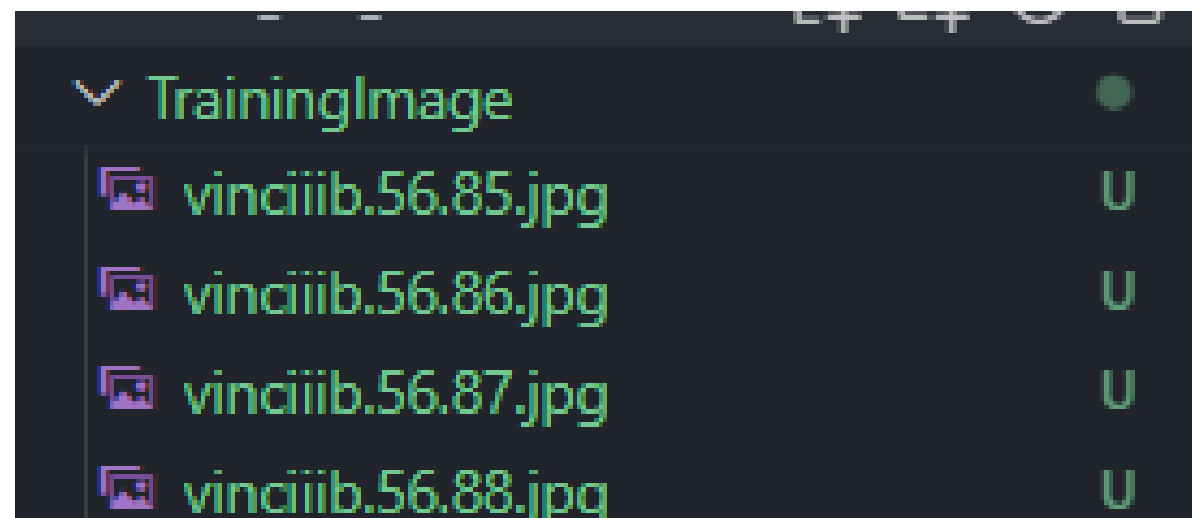
File yang berfungsi menjalankan

Capture_Image.py



Fungsi ini mengambil gambar wajah pengguna untuk membangun dataset yang digunakan dalam pelatihan model pengenalan wajah. Proses dimulai dengan meminta input ID dan Nama pengguna. Menggunakan webcam, gambar wajah diambil dan dideteksi dengan Haarcascade, lalu disimpan dalam folder TrainingImage dengan format {name}.{Id}.{sampleNumber}.jpg. Data pengguna berupa ID dan Nama juga disimpan dalam file DataMahasiswa/DataMahasiswa.csv.

yang dimana setiap tangkapan layar nanti akan masuk kedalam file ini, seperti berikut :



File yang berfungsi menjalankan

Train_Image.py

Fungsi ini melatih model pengenalan wajah menggunakan dataset yang diambil sebelumnya. Model menggunakan algoritma LBPH (Local Binary Patterns Histograms) untuk mengenali pola wajah. Dataset diambil dari folder TrainingImage, diolah menjadi data grayscale, dan dihubungkan dengan ID masing-masing wajah. Model yang telah dilatih kemudian disimpan dalam file TrainingImageLabel/Trainer.yml, sehingga dapat digunakan untuk proses pengenalan wajah berikutnya.

Recognize.py

Fungsi ini mengenali wajah pengguna dan mencatat kehadiran berdasarkan model yang telah dilatih. Proses dimulai dengan memuat file Trainer.yml dan Haarcascade untuk mendeteksi wajah. Kamera menangkap wajah pengguna, lalu model mencocokkan data wajah tersebut dengan dataset yang ada. Jika wajah dikenali, data kehadiran berupa ID, Nama, Tanggal, dan Waktu dicatat dalam file Kehadiran/DaftarHadir_{tanggal}.csv. Jika waktu deteksi melebihi batas tanpa pengenalan wajah, pesan "Kamu belum terdaftar!" akan ditampilkan.



Recognize.py

main.py

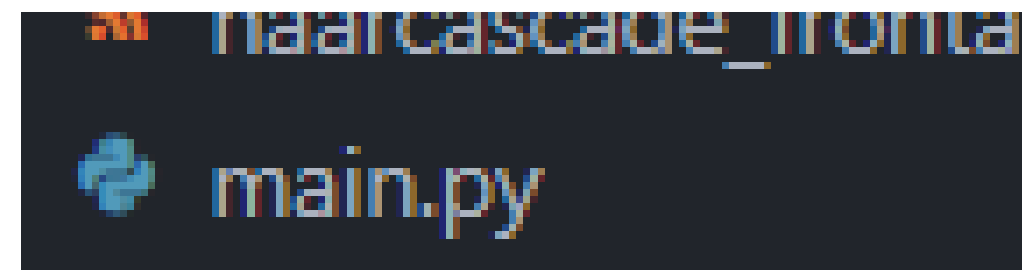
Merupakan antarmuka utama aplikasi dengan menu yang memungkinkan pengguna memilih fungsi yang diinginkan. Menu terdiri dari:

1. Tangkap Gambar: Memanggil fungsi dari `Capture_Image.py` untuk mengambil dataset wajah pengguna.

2. Latih Gambar: Memanggil fungsi dari `Train_Image.py` untuk melatih model pengenalan wajah.

3. Pengenalan & Catat Kehadiran: Memanggil fungsi dari `Recognize.py` untuk mengenali wajah pengguna dan mencatat kehadirannya.

4. Keluar: Mengakhiri aplikasi.



```
python main.py
```

THANK YOU

PEOPLE IOT - 2024

 [github.com/IOT-D3TI23/People IoT FINAL](https://github.com/IOT-D3TI23/People_IoT_FINAL)

