

# **LAPORAN PROYEK**

## **Internet of Things**

**Sistem Kehadiran dengan Pengenalan Wajah**



### **KELOMPOK-People IoT**

**Anggota :**

<b>Nama</b>	<b>NIM</b>
Cheryl Meilyn Lovica	11323006
Vinci G. Baringbing	11323051
Febyanti Hutahaeen	11323055

**Prodi: D-III Teknologi Informasi 2023**  
**Fakultas Vokasi**

**Institute Teknologi DEL**

## DAFTAR ISI

DAFTAR ISI .....	2
1    Pendahuluan .....	3
1.1    Pendahuluan .....	3
1.2    Latar Belakang .....	3
1.3    Pernyataan Masalah.....	4
1.4    Maksud dan Tujuan.....	4
1.5    Flow Chart.....	5
2    Tinjauan Pustaka .....	6
2.1    Sistem Kehadiran Siswa .....	6
2.2    Pemrosesan Gambar Digital .....	6
2.3    Langkah-langkah dalam Pemrosesan Gambar Digital .....	6
2.4    Definisi Istilah dan Sejarah .....	7
2.4.1    Face Detection .....	7
2.4.2    Face Recognition .....	7
2.4.3    Perbedaan antara Face Detection dan Face Recognition .....	7
2.4.4    Face Detection .....	8
3    Impelementasi & Analisis Model .....	9
3.1    Pendahuluan .....	9
3.2    Implementasi Model.....	9
3.3    Design Requirements .....	10
3.3.1    OpenCV: .....	10
3.4    Hasil Eksperimental .....	10
3.4.1    Deteksi Wajah:.....	10
3.4.2    Pengenalan Wajah: .....	11
4    Implementasi Kode .....	13
4.1    Capture_Image.py .....	13
4.1.1    Impor Modul.....	13
4.1.2    Fungsi is_number(s) .....	13
4.1.3    Fungsi takeImages().....	14
4.2    Recognize.py .....	16
4.2.1    Impor Modul.....	16
4.2.2    Memuat Model dan File Pendukung.....	16
4.2.3    Inisialisasi Kamera.....	17
4.2.4    Deteksi dan Pengenalan Wajah.....	17
4.2.5    Menampilkan Hasil.....	18
4.2.6    Mencatat Kehadiran.....	18
4.2.7    Penanganan Kesalahan .....	18
4.2.8    Proses Keseluruhan.....	18
4.3    Train_Image.py .....	19
4.3.1    Impor Modul.....	19
4.3.2    Fungsi getImagesAndLabels(path) .....	19
4.3.3    Fungsi TrainImages().....	20
4.3.4    Fungsi counter_img(path) (Opsional) .....	21
4.4    main.py.....	21
4.4.1    Fungsi title_bar() .....	21
4.4.2    Fungsi mainMenu().....	22
4.4.3    Fungsi CaptureFaces() .....	23
4.4.4    Fungsi Trainimages().....	23
4.4.5    Fungsi RecognizeFaces() .....	23
4.4.6    Driver Program: .....	24
5    Kesimpulan dan Saran.....	25
5.1    Kesimpulan .....	25
5.2    Saran.....	25

# **1 Pendahuluan**

## **1.1 Pendahuluan**

Kehadiran merupakan hal yang sangat penting bagi pengajar dan siswa di setiap lembaga pendidikan. Oleh karena itu, penting untuk mencatat kehadiran dengan tepat. Namun, sistem tradisional dalam mencatat kehadiran sering kali menimbulkan masalah, terutama dalam hal konsumsi waktu dan tenaga. Untuk mengatasi masalah ini, sistem absensi otomatis menjadi solusi yang lebih efisien. Beberapa institusi sudah menggunakan sistem otomatis seperti teknik biometrik dan RFID, meskipun kedua sistem ini lebih maju dari metode konvensional, mereka masih belum memenuhi batasan waktu yang dibutuhkan. Siswa sering harus mengantri untuk mencatat kehadiran, yang tentu menghabiskan waktu.

Proyek ini memperkenalkan sistem pencatatan kehadiran yang tidak mengganggu kegiatan pengajaran. Sistem ini dapat diterapkan dalam berbagai situasi, termasuk saat ujian, atau dalam kegiatan lain yang membutuhkan pencatatan kehadiran. Sistem ini menghilangkan metode konvensional seperti memanggil nama atau memeriksa kartu identitas siswa, yang dapat mengganggu proses pengajaran dan menyebabkan stres bagi siswa, terutama selama ujian. Untuk menggunakan sistem ini, siswa harus mendaftar terlebih dahulu dalam database, dan pendaftaran dapat dilakukan langsung di tempat dengan antarmuka yang mudah digunakan.

## **1.2 Latar Belakang**

Pengenalan wajah memainkan peran penting dalam kehidupan sehari-hari untuk mengenali keluarga, teman, atau orang yang kita kenal. Tanpa kita sadari, proses pengenalan wajah melibatkan banyak langkah yang dilakukan oleh sistem visual manusia. Melalui pemrosesan visual, kita menganalisis bentuk, ukuran, dan tekstur objek untuk mengenali wajah. Namun, tantangan besar adalah membangun sistem otomatis yang dapat meniru kemampuan manusia dalam mengenali wajah. Dalam lingkungan seperti universitas yang memiliki banyak mahasiswa dengan latar belakang berbeda, mengenali setiap wajah secara manual hampir tidak mungkin tanpa kesalahan. Oleh karena itu, komputer dengan memori besar dan kemampuan pemrosesan cepat digunakan dalam sistem pengenalan wajah.

Pengenalan wajah adalah metode biometrik di mana identifikasi dilakukan dengan mencocokkan gambar wajah secara langsung dengan gambar yang tersimpan di database. Penggunaan pengenalan wajah telah menyebar luas karena kesederhanaannya dan kinerja yang luar biasa. Sistem ini digunakan di berbagai bidang, seperti keamanan bandara, investigasi kriminal oleh FBI, hingga aplikasi sosial seperti Facebook yang memungkinkan pengenalan wajah dalam foto. Selain itu, perusahaan besar seperti Intel dan Apple juga mengimplementasikan teknologi ini untuk meningkatkan kenyamanan dan keamanan penggunaannya.

Pengenalan wajah dimulai pada tahun 1960 dengan sistem yang mengandalkan fitur wajah seperti mata, hidung, dan mulut. Seiring berjalannya waktu, penelitian semakin berkembang, dan teknik seperti PCA (Analisis Komponen Utama) mulai diterapkan untuk menyempurnakan sistem pengenalan wajah ini.

### **1.3 Pernyataan Masalah**

Metode tradisional dalam mencatat kehadiran siswa sering menimbulkan berbagai masalah. Sistem absensi berbasis pengenalan wajah menawarkan solusi sederhana dengan menghilangkan metode absensi konvensional seperti memanggil nama atau memeriksa kartu identitas siswa. Proses ini tidak hanya mengganggu pembelajaran, tetapi juga mengalihkan perhatian siswa, terutama selama ujian. Selain itu, distribusi lembar absensi di kelas besar bisa menjadi tantangan. Sistem ini menggantikan absensi manual yang memakan waktu dan menyebabkan gangguan bagi siswa. Lebih lanjut, sistem absensi otomatis berbasis pengenalan wajah mengatasi potensi kecurangan dan mengurangi kebutuhan dosen untuk menghitung kehadiran siswa secara berulang.

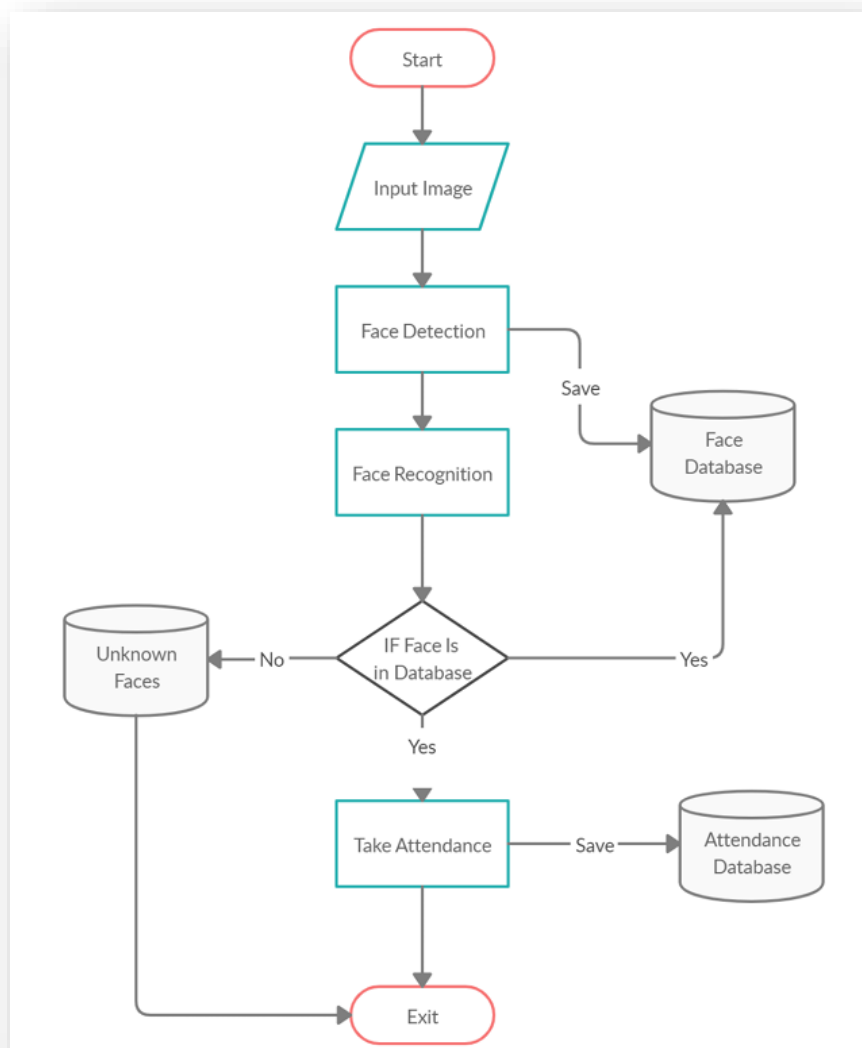
Beberapa penelitian telah mengidentifikasi masalah dalam sistem pengenalan wajah, termasuk kesulitan membedakan gambar yang dikenal dan tidak dikenal serta tantangan yang timbul selama pelatihan sistem yang memakan waktu. Selain itu, faktor seperti pencahayaan yang bervariasi dan perubahan pose kepala sering mempengaruhi kinerja sistem ini. Oleh karena itu, diperlukan pengembangan sistem absensi siswa yang dapat beroperasi secara real-time, dengan proses identifikasi yang cepat dan akurat, serta toleransi terhadap perubahan kondisi seperti pencahayaan dan pose.

### **1.4 Maksud dan Tujuan**

Proyek ini bertujuan untuk mengembangkan sistem absensi yang menggunakan teknologi pengenalan wajah. Tujuan yang ingin dicapai untuk menyelesaikan proyek ini antara lain:

- Mendeteksi bagian wajah dalam bingkai video.
- Mengekstraksi fitur penting dari wajah yang terdeteksi.
- Mengklasifikasikan fitur-fitur tersebut untuk mengenali wajah yang terdeteksi.
- Mencatat kehadiran siswa yang berhasil dikenali.

## 1.5 Flow Chart



Flowchart ini menunjukkan alur proses sistem absensi berbasis pengenalan wajah. Proses dimulai dengan input gambar atau video wajah ke sistem. Kemudian, sistem mendeteksi wajah dalam input tersebut. Jika wajah terdeteksi, data wajah akan disimpan ke dalam Face Database. Setelah itu, sistem melakukan pengenalan wajah (face recognition) untuk mencocokkan wajah dengan data yang ada di Face Database. Jika wajah ditemukan dalam database, sistem akan mencatat kehadiran individu tersebut ke Attendance Database. Namun, jika wajah tidak dikenali, data akan dianggap sebagai wajah tak dikenal (unknown faces). Setelah semua proses selesai, sistem mengakhiri operasi.

## 2 Tinjauan Pustaka

### 2.1 Sistem Kehadiran Siswa

Menurut Arun Katara dkk. (2017), sistem RFID, sidik jari, dan pengenalan iris memiliki sejumlah kelemahan. Meskipun sistem RFID sederhana, pengguna sering meminjamkan kartu identitasnya kepada teman sehingga memungkinkan terjadinya kecurangan. Sistem sidik jari, meskipun akurat, kurang efisien karena membutuhkan waktu yang cukup lama untuk proses verifikasi, membuat pengguna harus mengantri untuk memindai sidik jari satu per satu. Sementara itu, wajah manusia selalu terlihat dan memiliki informasi yang lebih sedikit dibandingkan iris, tetapi sistem pengenalan iris, meskipun lebih rinci, dapat dianggap melanggar privasi pengguna. Pengenalan suara juga merupakan alternatif, namun akurasi lebih rendah dibandingkan metode lainnya. Dengan mempertimbangkan berbagai kelebihan dan kekurangan, sistem pengenalan wajah dinilai lebih cocok untuk digunakan dalam sistem absensi siswa.

### 2.2 Pemrosesan Gambar Digital

Pemrosesan Citra Digital adalah proses pengolahan gambar yang bersifat digital dengan menggunakan komputer digital. Teknik pemrosesan citra digital didorong oleh tiga aplikasi utama, yaitu:

- Peningkatan informasi gambar untuk persepsi manusia.
- Pemrosesan gambar untuk aplikasi mesin otonom.
- Penyimpanan dan transmisi yang efisien.

### 2.3 Langkah-langkah dalam Pemrosesan Gambar Digital

Pemrosesan citra digital mencakup tugas-tugas dasar berikut:

- **Akuisisi Gambar** – Menggunakan sensor pencitraan dan kemampuan untuk mendigitalkan sinyal yang dihasilkan oleh sensor.
- **Prapemrosesan** – Meningkatkan kualitas gambar, seperti penyaringan, peningkatan kontras, dan lainnya.
- **Segmentasi** – Memisahkan gambar input menjadi bagian-bagian atau objek penyusunnya.
- **Deskripsi/Pemilihan Fitur** – Mengekstrak deskripsi objek gambar yang sesuai untuk pemrosesan komputer lebih lanjut.
- **Pengenalan dan Interpretasi** – Memberikan label pada objek berdasarkan informasi yang disediakan oleh deskriptornya. Interpretasi memberikan makna pada sekumpulan objek berlabel.
- **Basis Pengetahuan** – Membantu dalam pemrosesan yang efisien serta kerja sama antar modul.

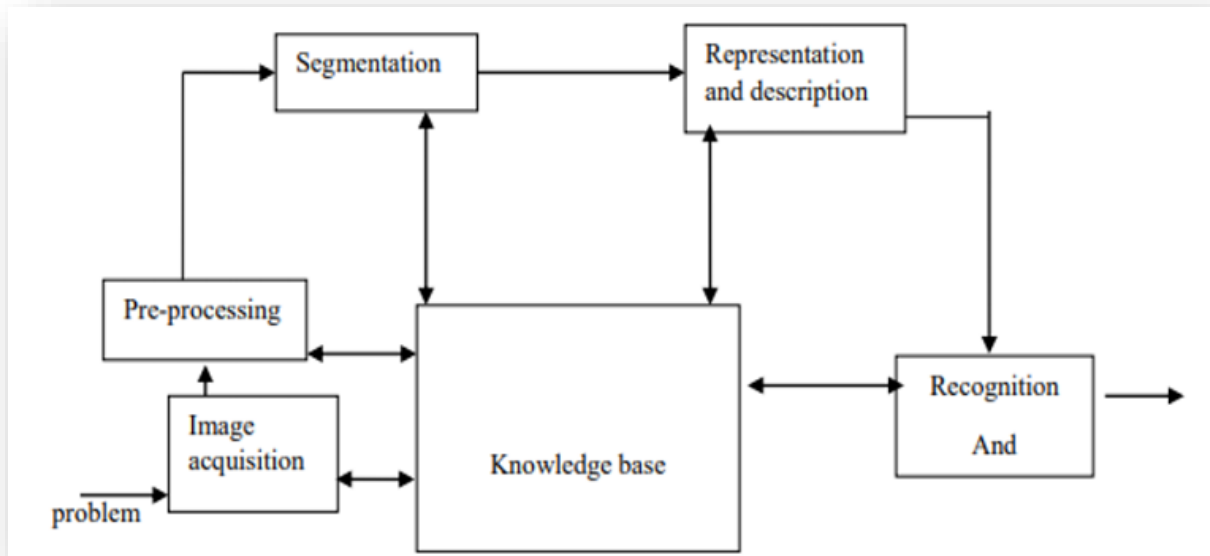


Diagram yang menunjukkan Langkah-langkah dalam pemrosesan gambar digital.

## 2.4 Definisi Istilah dan Sejarah

### 2.4.1 Face Detection

Face detection adalah proses untuk mengidentifikasi dan menemukan semua wajah yang ada dalam satu gambar atau video tanpa memperhatikan posisi, skala, orientasi, usia, dan ekspresi wajah. Selain itu, deteksi wajah harus tetap berfungsi meskipun terdapat kondisi pencahayaan yang tidak ideal serta variasi konten gambar dan video.

### 2.4.2 Face Recognition

Face recognition adalah masalah pengenalan pola visual, di mana wajah, yang direpresentasikan sebagai objek tiga dimensi dengan berbagai faktor seperti pencahayaan dan pose, perlu diidentifikasi berdasarkan gambar yang diperoleh. Pengenalan wajah merupakan tugas untuk mengidentifikasi wajah yang sudah terdeteksi sebagai wajah yang dikenal atau tidak dikenal, dan dalam kasus yang lebih maju, menentukan secara pasti wajah siapa itu.

### 2.4.3 Perbedaan antara Face Detection dan Face Recognition

Face detection menjawab pertanyaan, "Di mana wajahnya?" dengan mengidentifikasi objek sebagai "wajah" dan menentukan lokasinya di dalam gambar input. Sebaliknya, face recognition menjawab pertanyaan, "Siapa ini?" atau "Wajah siapa ini?" dengan menentukan apakah wajah yang terdeteksi adalah seseorang yang dikenal atau tidak dikenal berdasarkan database wajah yang digunakan untuk memvalidasi gambar input tersebut. Dapat dilihat bahwa output dari face detection (wajah yang terdeteksi) adalah input untuk face recognition, dan output dari face recognition adalah keputusan akhir, yaitu wajah dikenal atau tidak dikenal.

#### **2.4.4 Face Detection**

Sebuah detektor wajah harus mampu menentukan apakah sebuah gambar dengan ukuran acak mengandung wajah manusia, dan jika iya, di mana letaknya. Face detection dapat dilakukan berdasarkan beberapa parameter: warna kulit (untuk wajah dalam gambar dan video berwarna), gerakan (untuk wajah dalam video), bentuk wajah/kepala, penampilan wajah, atau kombinasi dari parameter-parameter ini. Sebagian besar algoritma pendeteksian wajah modern berbasis pada penampilan tanpa menggunakan parameter lainnya.

Gambar input dipindai di semua lokasi dan skala yang memungkinkan oleh sub-jendela. Face detection dianggap sebagai pengklasifikasian pola di sub-jendela, baik sebagai wajah atau bukan wajah. Pengklasifikasi wajah/bukan wajah ini dilatih menggunakan contoh wajah dan bukan wajah dengan metode pembelajaran statistik. Sebagian besar algoritma modern didasarkan pada kerangka kerja pendeteksian objek Viola Jones, yang berbasis pada Haar Cascades.



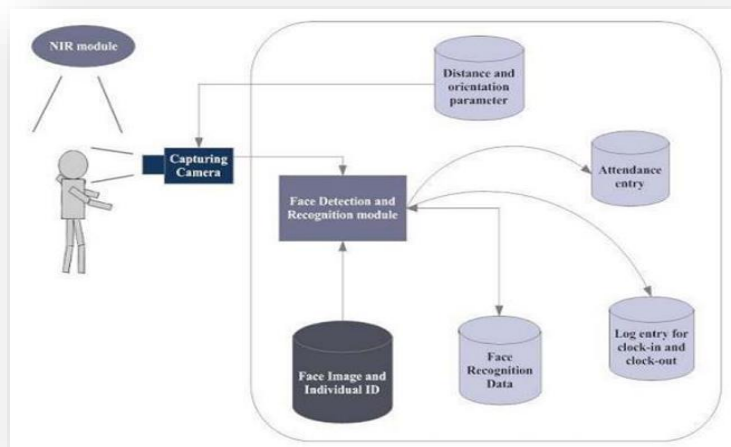
### 3 Implemtasi & Analisis Model

#### 3.1 Pendahuluan

Deteksi wajah melibatkan pemisahan jendela gambar menjadi dua kelas: satu yang mengandung wajah dan satu lagi yang dianggap sebagai latar belakang (gangguan visual). Proses ini sulit karena, meskipun terdapat kesamaan di antara wajah, variasi besar tetap ada dalam hal usia, warna kulit, dan ekspresi wajah. Masalah ini menjadi lebih rumit karena kondisi pencahayaan yang berbeda, kualitas gambar, geometri, serta kemungkinan adanya obstruksi parsial atau penyamaran. Oleh karena itu, sebuah detektor wajah yang ideal harus mampu mendeteksi keberadaan wajah apa pun dalam kondisi pencahayaan apa pun dan dengan latar belakang apa pun.

Tugas deteksi wajah dapat dibagi menjadi dua langkah. Langkah pertama adalah tugas klasifikasi, di mana gambar acak diambil sebagai masukan dan menghasilkan nilai biner "ya" atau "tidak" untuk menunjukkan apakah ada wajah dalam gambar tersebut. Langkah kedua adalah tugas pelokalan wajah, yang bertujuan mengambil gambar sebagai masukan dan memberikan lokasi wajah dalam bentuk kotak pembatas dengan koordinat (x, y, lebar, tinggi). Setelah gambar diambil, sistem akan membandingkan kesamaan gambar tersebut dengan gambar dalam basis datanya dan memberikan hasil yang paling relevan. Kami akan menggunakan sistem operasi Raspbian, platform OpenCV, dan melakukan pengkodean menggunakan bahasa Python.

#### 3.2 Implementasi Model



Komponen utama yang digunakan dalam pendekatan implementasi ini adalah pustaka penglihatan komputer sumber terbuka (OpenCV). Salah satu tujuan OpenCV adalah menyediakan infrastruktur visi komputer yang mudah digunakan untuk membantu orang membangun aplikasi visi yang cukup canggih dengan cepat. Pustaka OpenCV memiliki lebih dari 500 fungsi yang mencakup berbagai bidang dalam visi komputer. Teknologi utama di balik pengenalan wajah adalah OpenCV. Pengguna berdiri di depan kamera dengan jarak minimum 50 cm, dan gambarnya diambil sebagai input. Wajah depan diekstraksi dari gambar, kemudian dikonversi menjadi skala abu-abu dan disimpan. Algoritma **Principal Component Analysis (PCA)** diterapkan pada gambar, dan nilai eigen disimpan dalam file xml. Ketika pengguna

meminta pengenalan, wajah depan diekstraksi dari bingkai video yang diambil melalui kamera. Nilai eigen dihitung ulang untuk wajah uji dan dibandingkan dengan data yang disimpan untuk menemukan kecocokan terdekat.

### 3.3 Design Requirements

Untuk membangun sistem HFR (Human Face Recognition), kami memanfaatkan beberapa alat penting. Tanpa dukungan alat-alat ini, sistem tidak dapat diselesaikan dengan baik. Salah satu alat utama yang digunakan adalah sebagai berikut:

#### 3.3.1 OpenCV:

Kami menggunakan OpenCV versi 3 sebagai pustaka utama untuk Python 3. OpenCV adalah pustaka yang kaya akan fungsi pemrosesan gambar, sangat cocok untuk berbagai kebutuhan pengolahan citra. Bahkan, pustaka ini memungkinkan pengguna mendapatkan hasil yang memuaskan tanpa perlu menulis kode secara manual. OpenCV bersifat lintas platform dan tersedia secara gratis di bawah lisensi BSD sumber terbuka. Berikut adalah beberapa fitur utama yang didukung oleh OpenCV:

- **Derivasi:** Menghitung gradien, laplacian, dan membatasi kontur.
- **Transformasi Hough:** Mendeteksi garis, segmen, lingkaran, dan bentuk geometris.
- **Histogram:** Melakukan penghitungan, penyamaan, dan lokalisasi objek menggunakan algoritma proyeksi balik.
- **Segmentasi:** Penentuan ambang batas, transformasi jarak, deteksi latar depan/belakang, dan segmentasi watershed.
- **Penyaringan:** Melakukan filter linier, non-linier, dan operasi morfologi.
- **Detektor Cascade:** Digunakan untuk deteksi wajah, mata, atau plat nomor kendaraan.
- **Titik Minat:** Deteksi dan pencocokan fitur penting dalam gambar.
- **Pemrosesan Video:** Meliputi aliran optik, pengurangan latar belakang, dan pelacakan objek.
- **Fotografi:** Mendukung pembuatan panorama, pencitraan definisi tinggi (HDR), dan perbaikan gambar (inpainting).

### 3.4 Hasil Eksperimental

Berikut adalah langkah-langkah dalam proses eksperimen:

#### 3.4.1 Deteksi Wajah:

Gambar ditangkap melalui kamera web dari sisi klien.

##### Langkah-langkah:

- Lakukan **pra-pemrosesan** pada gambar yang ditangkap untuk mengekstrak wajah.
- Hitung nilai eigen dari wajah yang ditangkap, lalu bandingkan dengan nilai eigen wajah yang sudah ada di database.

- Jika nilai eigen tidak ditemukan cocok, informasi wajah baru akan disimpan ke database wajah dalam file xml.
- Jika nilai eigen cocok, sistem akan melanjutkan ke proses pengenalan wajah.

**Proses selesai.**

### 3.4.2 Pengenalan Wajah:

Langkah-langkah berikut dilakukan menggunakan algoritma PCA untuk proses pengenalan wajah:

#### Langkah-langkah:

- Identifikasi informasi wajah yang sesuai dengan gambar di database.
- Catat kehadiran siswa dengan memperbarui tabel log menggunakan gambar wajah dan waktu sistem.

**Proses selesai.**

Bagian ini menampilkan hasil eksperimen yang bertujuan mengonversi wajah menjadi gambar skala abu-abu dengan resolusi 50x50 piksel.

Test Data	Hasil yang diharapkan	Diamati Hasil	Lulus/Gagal
OpenCAM_CB()	Terhubung dengan kamera yang terpasang dan mulai bermain.	Kamera dimulai.	Lulus
LoadHaar Classifier()	Memuat Kaskade Pengklasifikasi Haar file untuk wajah depan	Bersiap untuk Ekstraksi.	Lulus
ExtractFace()	Memulai program Paul- Viola Bingkai ekstraksi wajah bekerja.	Wajah diekstraksi.	Lulus
Learn()	Memulai PCA Algoritma	Memperbaharui facedata. xml	Lulus
Recognize()	Ini membandingkan wajah input dengan wajah yang disimpan.	Wajah terdekat.	Lulus

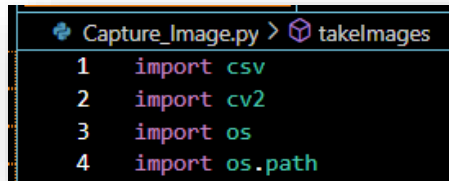
Berikut ini adalah sampel Kumpulan data kami:



## 4 Implementasi Kode

### 4.1 Capture\_Image.py

#### 4.1.1 Impor Modul

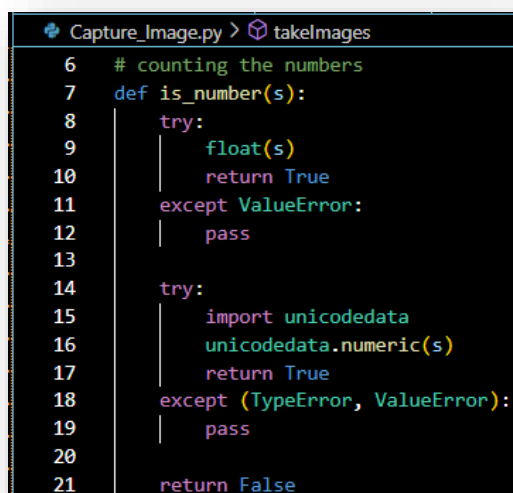


```
1 import csv
2 import cv2
3 import os
4 import os.path
```

Kode ini mengimpor modul yang diperlukan untuk menjalankan fungsi:

- **csv**: Untuk membaca dan menulis data dalam format CSV.
- **cv2**: Pustaka OpenCV digunakan untuk memproses gambar dan mendeteksi wajah.
- **os** dan **os.path**: Untuk bekerja dengan direktori dan file di sistem operasi.

#### 4.1.2 Fungsi is\_number(s)



```
6 # counting the numbers
7 def is_number(s):
8     try:
9         float(s)
10        return True
11    except ValueError:
12        pass
13
14    try:
15        import unicodedata
16        unicodedata.numeric(s)
17        return True
18    except (TypeError, ValueError):
19        pass
20
21    return False
```

**Fungsi ini memeriksa apakah input berupa angka. Langkah-langkah:**

- Mencoba mengonversi input ke tipe float.
- Jika gagal, mencoba mengonversi ke angka unicode.
- Jika semua langkah gagal, fungsi mengembalikan False.

### 4.1.3 Fungsi takelImages()

```
Capture_Image.py > ...
23 # Take image function
24 def takeImages():
25
26     Id = input("Masukkan ID Anda: ")
27     name = input("Masukkan Nama Anda: ")
28
29     if(is_number(Id) and name.isalpha()):
30         cam = cv2.VideoCapture(0)
31         harcascadePath = "haarcascade_frontalface_default.xml"
32         detector = cv2.CascadeClassifier(harcascadePath)
33         sampleNum = 0
34
35         while(True):
36             ret, img = cam.read()
37             gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
38             faces = detector.detectMultiScale(gray, 1.3, 5, minSize=(30,30), flags = cv2.CASCADE_SCALE_IMAGE)
39             for(x,y,w,h) in faces:
40                 cv2.rectangle(img, (x, y), (x+w, y+h), (10, 159, 255), 2)
41                 #incrementing sample number
42                 sampleNum = sampleNum+1
43                 #saving the captured face in the dataset folder TrainingImage
44                 cv2.imwrite("TrainingImage" + os.sep + name + "." + Id + "." +
45                     | | | str(sampleNum) + ".jpg", gray[y:y+h, x:x+w])
46                 #display the frame
47                 cv2.imshow('Tangkap Gambar', img)
48                 #wait for 100 milliseconds
49                 if cv2.waitKey(100) & 0xFF == ord('q'):
50                     break
51                 # break if the sample number is more than 100
52                 elif sampleNum > 100:
53                     break
54             cam.release()
55             cv2.destroyAllWindows()
56             res = "Gambar disimpan untuk ID : " + Id + " Nama : " + name
57             header=["Id", "Name"]
58             row = [Id, name]
59             if(os.path.isfile("DataMahasiswa"+os.sep+"DataMahasiswa.csv")):
60                 with open("DataMahasiswa"+os.sep+"DataMahasiswa.csv", 'a+') as csvFile:
61                     writer = csv.writer(csvFile)
62                     writer.writerow(j for j in row)
63                     csvFile.close()
64             else:
65                 with open("DataMahasiswa"+os.sep+"DataMahasiswa.csv", 'a+') as csvFile:
66                     writer = csv.writer(csvFile)
67                     writer.writerow(i for i in header)
68                     writer.writerow(j for j in row)
69                     csvFile.close()
70             else:
71                 if(is_number(Id)):
72                     print("Masukkan Nama dengan Huruf Alphabet")
73                 if(name.isalpha()):
74                     print("Masukkan ID dengan Angka")
```

Fungsi utama untuk menangkap gambar wajah dan menyimpannya dalam basis data.

- **Input:**

- Id: ID unik pengguna, harus berupa angka.
- name: Nama pengguna, harus berupa huruf alfabet.

**Langkah-langkah Utama dalam Fungsi:**

1. **Validasi Input:**

- Id harus berupa angka (dicek dengan `is_number()`).
- name harus berupa huruf alfabet (dicek dengan `.isalpha()`).

## 2. Inisialisasi Kamera:

- Kamera diaktifkan menggunakan `cv2.VideoCapture(0)`.

## 3. Deteksi Wajah:

- Menggunakan file Haar Cascade (`haarcascade_frontalface_default.xml`) untuk mendeteksi wajah.
- Wajah yang terdeteksi diberikan kotak pembatas berwarna.

## 4. Menyimpan Gambar Wajah:

- Gambar wajah diproses menjadi skala abu-abu (`cv2.cvtColor()`).
- Bagian wajah yang terdeteksi dipotong dan disimpan dalam folder **TrainingImage** dengan format `Nama.ID.NomorSample.jpg`.

## 5. Penghentian Proses:

- Proses berhenti jika tombol 'q' ditekan atau jumlah sampel lebih dari 100.

## 6. Menyimpan Data ke CSV:

- Informasi Id dan name disimpan ke file CSV:
  - Jika file **DataMahasiswa.csv** sudah ada, data baru ditambahkan.
  - Jika belum ada, file dibuat, dan header ditulis sebelum data ditambahkan.

## 7. Pesan dan Validasi Input:

- Jika validasi gagal:
  - Menampilkan pesan error ("Masukkan Nama dengan Huruf Alfabeta" atau "Masukkan ID dengan Angka").

## Penjelasan Perintah Penting

- **cv2.CascadeClassifier()**: Membuat detektor wajah menggunakan file XML Haar Cascade.
- **cv2.imwrite()**: Menyimpan gambar yang dipotong ke file dengan format tertentu.
- **os.sep**: Digunakan untuk memastikan pemisah direktori kompatibel lintas platform (\ di Windows, / di Linux/Mac).
- **cv2.imshow()**: Menampilkan gambar yang diproses pada jendela baru.
- **cv2.waitKey()**: Menunggu input keyboard untuk melanjutkan atau menghentikan proses.

## 4.2 Recognize.py

### 4.2.1 Impor Modul

```
Recognize.py > recognize_attendance
1 import datetime
2 import os
3 import time
4 import cv2
5 import pandas as pd
```

Kode ini menggunakan beberapa modul penting:

- **datetime**: Untuk mencatat waktu dan tanggal.
- **os**: Untuk bekerja dengan sistem file.
- **time**: Untuk menghitung batas waktu.
- **cv2**: OpenCV digunakan untuk pengolahan citra dan video.
- **pandas**: Digunakan untuk membaca dan menulis data dalam format CSV.

### 4.2.2 Memuat Model dan File Pendukung

- Recognizer

```
# Load recognizer and face cascade
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read("../TrainingImageLabel/Trainer.yml")
```

Memuat model **LBPH (Local Binary Patterns Histograms)** yang telah dilatih untuk pengenalan wajah.

- Haar Cascade

```
harcascadePath = "haarcascade_frontalface_default.xml"
```

Memuat file Haar Cascade yang digunakan untuk mendeteksi wajah.

- Data Siswa

```
# Read Data Mahasiswa
student_details_path = "DataMahasiswa" + os.sep + "DataMahasiswa.csv"
```

Membaca file CSV yang berisi data ID dan nama siswa.



### 4.2.3 Inisialisasi Kamera

- Kamera diaktifkan dengan:

```
# Initialize camera  
cam = cv2.VideoCapture(0)
```

Kamera akan digunakan untuk menangkap video.

- Resolusi video diatur ke **640x480**, dan ukuran minimum deteksi wajah dihitung:

```
minW = 0.1 * cam.get(3)  
minH = 0.1 * cam.get(4)
```

### 4.2.4 Deteksi dan Pengenalan Wajah

- **Deteksi Wajah:** Gambar diubah menjadi skala abu-abu, dan wajah dideteksi:

```
gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)  
faces = faceCascade.detectMultiScale(  
    gray, 1.2, 5, minSize=(int(minW), int(minH)), flags=cv2.CASCADE_SCALE_IMAGE)
```

- **Prediksi Wajah:** Model **LBPH** digunakan untuk memprediksi wajah, dengan confidence yang dihitung:

```
Id, conf = recognizer.predict(gray[y:y + h, x:x + w])
```

Jika **confidence** di atas 60%, wajah dianggap dikenali, dan nama diambil dari file CSV:

```
# Record attendance if confidence > 60%  
if (100 - conf) > 60:  
    ts = time.time()  
    date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')  
    timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')  
    aa = str(aa)[2:-2] # Clean up the name string  
    attendance.loc[len(attendance)] = [Id, aa, date, timeStamp]  
    print(f"Kehadiran yang tercatat : {Id}, {aa}, {date}, {timeStamp}")  
    attendance_recorded = True
```

#### 4.2.5 Menampilkan Hasil

- Gambar dari kamera ditampilkan dengan informasi ID dan tingkat confidence:

```
cv2.putText(im, str(tt), (x + 5, y - 5), font, 1, (255, 255, 255), 2)
```

- **Timeout:** Jika waktu melebihi 30 detik tanpa mencatat kehadiran, sistem menampilkan pesan:

```
cv2.putText(im, "Kamu belum terdaftar!", (100, 100), font, 1, (0, 0, 255), 2)
```

#### 4.2.6 Mencatat Kehadiran

- Kehadiran disimpan dalam file CSV dengan format:  
**Id, Nama, Tanggal, dan Waktu.**
- File baru dibuat jika belum ada, atau data baru ditambahkan ke file yang sudah ada:

```
attendance.to_csv(fileName, mode='a', header=False, index=False)  
print(f"Kehadiran berhasil dicatat!")
```

#### 4.2.7 Penanganan Kesalahan

- **Kesalahan File:** Jika file Haar Cascade atau CSV data siswa tidak ditemukan, sistem akan menampilkan pesan error.
- **Kesalahan Kamera:** Jika kamera gagal diakses, program menampilkan pesan kesalahan dan keluar.
- **Kesalahan Tak Terduga:** Kesalahan lain ditangani dengan blok try-except, memastikan kamera dilepaskan dan jendela OpenCV ditutup.

#### 4.2.8 Proses Keseluruhan

- a. Sistem memuat model pengenalan wajah dan data siswa.
- b. Kamera diaktifkan untuk menangkap gambar.
- c. Wajah terdeteksi dan dikenali:
  - Jika wajah dikenali, kehadiran dicatat di file CSV.
  - Jika tidak dikenali, sistem menampilkan pesan.
- d. Sistem menutup kamera dan menyimpan hasil kehadiran.

## 4.3 Train\_Image.py

### 4.3.1 Impor Modul

```
Train_Image.py > getImagesAndLabels
1 import os
2 import time
3 import cv2
4 import numpy as np
5 from PIL import Image
6 from threading import Thread
```

- **os**: Untuk bekerja dengan sistem file.
- **time**: Digunakan untuk menambahkan jeda waktu (misalnya, dalam animasi counter).
- **cv2**: OpenCV digunakan untuk pengenalan wajah dan pelatihan.
- **numpy**: Untuk menangani array numerik (diperlukan untuk memproses data gambar).
- **PIL**: Pustaka untuk memproses gambar (digunakan untuk konversi gambar ke skala abu-abu).
- **threading.Thread**: Digunakan untuk menjalankan fungsi dalam thread terpisah agar lebih efisien.

### 4.3.2 Fungsi getImagesAndLabels(path)

**Tujuan:** Memuat gambar dari folder dan mengembalikan daftar wajah dan ID masing-masing.

**Langkah-langkah Utama:**

#### 1. Membaca Semua Gambar:

- Mengambil jalur semua file di folder path menggunakan:

```
# get the path of all the files in the folder
imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
# print(imagePaths)
```

#### 2. Mengonversi Gambar ke Skala Abu-Abu:

- Membuka gambar menggunakan `PIL.Image.open()`.
- Mengonversinya menjadi skala abu-abu dengan:

```
# loading the image and converting it to gray scale
pilImage = Image.open(imagePath).convert('L')
```

#### 3. Mengonversi ke Array NumPy:

- Mengubah gambar menjadi array numerik untuk digunakan dengan OpenCV:

```
# Now we are converting the PIL image into numpy array
imageNp = np.array(pilImage, 'uint8')
```

#### 4. Ekstraksi ID dari Nama File:

- ID diambil dari nama file, diasumsikan memiliki format Name.ID.xxx.jpg:

```
# getting the Id from the image  
Id = int(os.path.split(imagePath)[-1].split(".")[1])
```

#### 5. Mengembalikan Wajah dan ID:

- Wajah dan ID disimpan dalam dua daftar, faces dan Ids, lalu dikembalikan.

### 4.3.3 Fungsi TrainImages()

**Tujuan:** Melatih model pengenalan wajah menggunakan gambar yang telah dikumpulkan.

**Langkah-langkah Utama:**

#### 1. Inisialisasi Model Pengenalan Wajah:

- Menggunakan model **LBPHFaceRecognizer**:

```
recognizer = cv2.face_LBPHFaceRecognizer.create()
```

#### 2. Memuat Data Gambar:

- Memuat gambar wajah dan ID yang terkait dari folder TrainingImage menggunakan fungsi `getImagesAndLabels()`.

#### 3. Melatih Model:

- Model dilatih dengan gambar wajah (faces) dan ID (Id):

```
Thread(target = recognizer.train(faces, np.array(Id))).start()
```

#### 4. Penyimpanan Model:

- Model yang dilatih disimpan dalam file Trainer.yml:

```
recognizer.save("TrainingImageLabel"+os.sep+"Trainer.yml")  
print("Semua gambar telah dilatih dan disimpan.")
```

#### 5. Menambahkan Counter Visual (Opsional):

- Fungsi `counter_img(path)` digunakan untuk menampilkan jumlah gambar yang telah dilatih secara real-time.

#### 4.3.4 Fungsi counter\_img(path) (Opsional)

**Tujuan:** Menampilkan animasi penghitung untuk gambar yang dilatih.

**Langkah-langkah Utama:**

1. **Mengambil Semua Jalur Gambar:**

- Sama seperti getImagesAndLabels(), membaca semua jalur gambar dalam folder.

2. **Penghitungan Gambar:**

- Menggunakan loop untuk mencetak jumlah gambar yang telah dilatih:

```
print(str(imgcounter) + " Gambar dilatih", end="\r")
```

3. **Jeda Animasi:**

- Menambahkan jeda singkat menggunakan time.sleep(0.008) untuk memberikan efek visual.

#### Proses Utama

1. Gambar wajah dimuat dari folder TrainingImage.
2. Data gambar diubah menjadi array skala abu-abu dan ID diambil dari nama file.
3. Model pengenalan wajah **LBPH** dilatih menggunakan data tersebut.
4. Model yang telah dilatih disimpan di file Trainer.yml untuk digunakan dalam pengenalan wajah di masa mendatang.

## 4.4 main.py

### 4.4.1 Fungsi title\_bar()

**Tujuan:** Menampilkan header atau judul program di terminal.

**Langkah-langkah:**

- Menggunakan os.system('cls') untuk membersihkan layar terminal (khusus untuk Windows).
- Menampilkan judul sistem absensi berbasis pengenalan wajah dengan format dekoratif.

```
# creating the title bar function
def title_bar():
    os.system('cls') # for windows

    # title of the program

    print("*****")
    print("**** Sistem Kehadiran Pengenalan Wajah ****")
    print("*****")
```

#### 4.4.2 Fungsi mainMenu()

**Tujuan:** Menyediakan menu utama bagi pengguna untuk memilih opsi yang diinginkan.

**Langkah-langkah:**

- **Menampilkan Menu:**
  - Memanggil fungsi title\_bar() untuk membersihkan layar dan menampilkan judul.
  - Menampilkan empat opsi:
    - [1] Tangkap Gambar
    - [2] Latih Gambar
    - [3] Pengenalan & Catat Kehadiran
    - [4] Keluar.
- **Menerima Input Pengguna:**
  - Menggunakan input() untuk menerima pilihan pengguna.
  - Mengonversi input menjadi tipe int.
- **Mengarahkan ke Fungsi yang Sesuai:**
  - Jika pengguna memilih:
    - **1:** Memanggil fungsi CaptureFaces() untuk menangkap gambar.
    - **2:** Memanggil fungsi TrainImages() untuk melatih gambar.
    - **3:** Memanggil fungsi RecognizeFaces() untuk mengenali wajah dan mencatat kehadiran.
    - **4:** Menampilkan pesan "Terima Kasih" dan keluar dari program.
  - Jika pilihan tidak valid (bukan 1-4), meminta pengguna untuk mencoba lagi.
- **Menangani Error:**

Jika input pengguna tidak valid (misalnya huruf atau simbol), menampilkan pesan kesalahan dan meminta input ulang.

```
# creating the user main menu function
def mainMenu():
    title_bar()
    print()
    print(19 * "=", " MENU", 19 * "=")
    print("[1] Tangkap Gambar")
    print("[2] Latih Gambar")
    print("[3] Pengenalan & Catat Kehadiran")
    print("[4] Keluar")

    while True:
        try:
            choice = int(input("Masukan pilihan: "))

            if choice == 1:
                CaptureFaces()
                break
            elif choice == 2:
                TrainImages()
                break
            elif choice == 3:
                RecognizeFaces()
                break
            elif choice == 4:
                print("Terima Kasih")
                break
            else:
                print("Pilihan tidak tepat. Pilih 1-4")
                mainMenu()
        except ValueError:
            print("Pilihan tidak tepat. Pilih 1-4\nCoba lagi!")
    exit
```

#### 4.4.3 Fungsi CaptureFaces()

**Tujuan:** Memanggil fungsi takeImages() dari file Capture\_Image.py untuk menangkap gambar wajah.

**Langkah-langkah:**

- Memanggil Capture\_Image.takeImages() untuk menangkap gambar.
- Setelah selesai, meminta pengguna menekan Enter untuk kembali ke menu utama.
- Memanggil mainMenu() untuk menampilkan menu kembali.

```
# calling the take image function form capture image.py file
def CaptureFaces():
    Capture_Image.takeImages()
    key = input("Klik Enter untuk kembali ke halaman Menu")
    mainMenu()
```

#### 4.4.4 Fungsi Trainimages()

**Tujuan:** Memanggil fungsi TrainImages() dari file Train\_Image.py untuk melatih gambar yang telah diambil.

**Langkah-langkah:**

- Memanggil Train\_Image.TrainImages() untuk melatih gambar.
- Meminta pengguna menekan Enter untuk kembali ke menu utama.
- Memanggil mainMenu().

```
# calling the train images from train_images.py file
def Trainimages():
    Train_Image.TrainImages()
    key = input("Klik Enter untuk kembali ke halaman Menu")
    mainMenu()
```

#### 4.4.5 Fungsi RecognizeFaces()

**Tujuan:** Memanggil fungsi recognize\_attendence() dari file Recognize.py untuk mengenali wajah dan mencatat kehadiran.

**Langkah-langkah:**

- Memanggil Recognize.recognize\_attendence() untuk mengenali wajah dan mencatat kehadiran.
- Meminta pengguna menekan Enter untuk kembali ke menu utama.
- Memanggil mainMenu().

```
# calling the recognize_attendence from recognize.py file
def RecognizeFaces():
    Recognize.recognize_attendence()
    key = input("Klik Enter untuk kembali ke halaman Menu")
    mainMenu()
```

#### 4.4.6 Driver Program:

- **mainMenu()** dipanggil pertama kali ketika program dijalankan.
- Menu utama akan terus ditampilkan sampai pengguna memilih untuk keluar dengan memilih opsi 4.

```
# -----main driver -----  
mainMenu()
```

#### Penjelasan Alur Program

1. **Menu Utama:**
  - Program memulai dengan menampilkan menu utama kepada pengguna.
2. **Pilihan Pengguna:**
  - Pengguna memilih salah satu dari empat opsi:
    - **Tangkap Gambar:** Memulai proses pengambilan gambar wajah.
    - **Latih Gambar:** Melatih model pengenalan wajah dengan gambar yang telah diambil.
    - **Pengenalan & Catat Kehadiran:** Mengenali wajah dari kamera dan mencatat kehadiran.
    - **Keluar:** Menutup program.
3. **Fungsi Tertentu Dipanggil:**
  - Berdasarkan pilihan, program menjalankan fungsi terkait.
4. **Kembali ke Menu:**
  - Setelah menyelesaikan satu tugas, program kembali ke menu utama kecuali pengguna memilih keluar.



## **5 Kesimpulan dan Saran**

### **5.1 Kesimpulan**

Berdasarkan implementasi sistem absensi berbasis pengenalan wajah yang telah dijelaskan, dapat disimpulkan bahwa:

#### **1. Efisiensi Pencatatan Kehadiran**

Sistem ini mampu menggantikan metode pencatatan kehadiran manual dengan cara yang lebih cepat dan otomatis melalui pengenalan wajah, mengurangi kemungkinan manipulasi data kehadiran.

#### **2. Integrasi Teknologi Komputer Vision**

Dengan memanfaatkan pustaka OpenCV dan model LBPH, sistem ini mampu mendeteksi dan mengenali wajah dengan tingkat akurasi yang baik dalam kondisi pencahayaan yang sesuai.

#### **3. Penyimpanan Data Terstruktur**

Data kehadiran dicatat dalam format CSV yang mudah diakses, memungkinkan proses analisis lebih lanjut menggunakan alat analitik atau perangkat lunak lain.

#### **4. Kemudahan Penggunaan**

Dengan antarmuka berbasis terminal dan menu sederhana, pengguna dapat dengan mudah menangkap gambar, melatih model, dan mencatat kehadiran tanpa memerlukan pengetahuan teknis mendalam.

#### **5. Penggunaan Modular**

Struktur program yang modular (dengan file terpisah untuk berbagai fungsi utama seperti capture, train, dan recognize) mempermudah pengembangan lebih lanjut atau penyesuaian sistem.

Namun, terdapat beberapa batasan, seperti ketergantungan pada kualitas pencahayaan, sudut wajah, dan kondisi kamera yang dapat memengaruhi akurasi pengenalan wajah.

### **5.2 Saran**

Untuk pengembangan lebih lanjut, berikut adalah beberapa saran untuk meningkatkan sistem ini:

#### **1. Peningkatan Akurasi Deteksi dan Pengenalan Wajah**

Implementasi model pengenalan wajah berbasis deep learning seperti FaceNet atau dlib untuk meningkatkan akurasi, terutama pada kondisi pencahayaan atau sudut wajah yang kurang ideal.

Penambahan fitur augmentasi data untuk menangani variasi ekspresi wajah, sudut, dan pencahayaan.

#### **2. Integrasi Antarmuka Grafis (GUI)**

Pengembangan antarmuka pengguna berbasis GUI menggunakan pustaka seperti Tkinter atau PyQt untuk meningkatkan pengalaman pengguna dan mempermudah interaksi dengan sistem.

### **3. Keamanan Data**

Meningkatkan keamanan data kehadiran dengan menyimpan file dalam database terenkripsi, seperti MySQL atau MongoDB.

Menambahkan autentikasi berbasis multi-faktor untuk mengakses sistem.

### **4. Peningkatan Performa dan Skalabilitas**

Mengoptimalkan pengelolaan sumber daya seperti kamera dan proses pelatihan untuk meningkatkan performa, khususnya pada perangkat dengan spesifikasi rendah.

Mengintegrasikan sistem ini ke dalam jaringan perusahaan sehingga dapat digunakan secara terpusat untuk beberapa lokasi.

### **5. Analitik Kehadiran**

Penambahan fitur visualisasi data untuk menyajikan statistik kehadiran (grafik mingguan/bulanan) guna memberikan wawasan kepada manajemen.

### **6. Penggunaan Kamera Khusus**

Memanfaatkan kamera dengan resolusi tinggi atau inframerah untuk deteksi yang lebih akurat dalam berbagai kondisi pencahayaan.

Dengan pengembangan ini, diharapkan sistem absensi berbasis pengenalan wajah dapat memberikan manfaat lebih luas dalam berbagai skenario, seperti institusi pendidikan, perusahaan, dan sektor lainnya.