

## BAB V

### IMPLEMENTASI DAN PENGUJIAN SISTEM

#### 5.1 HASIL IMLEMENTASI PROGRAM

Pada tahap ini penulis mengimplementasikan hasil dari perancangan program yang dibuat. Perancangan *software* ditulis menggunakan bahasa pemrograman *Phyton* yang dioperasikan pada *Windows*.

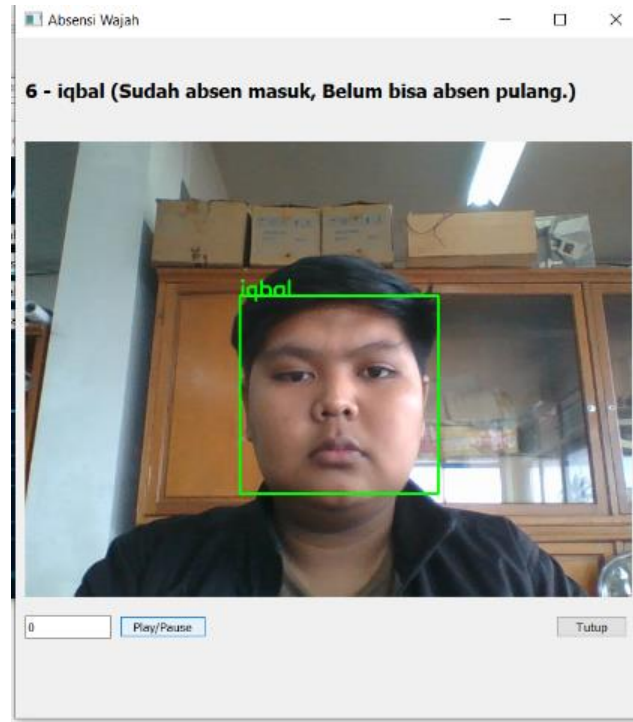
Tahap – tahap implementasi adalah sebagai berikut:

1. Menyiapkan *Libraries* yang diperlukan.
2. Menulis *code* program absensi pendeteksian wajah pada *Visual Studio Code*.
3. Menguji program pendeteksian wajah yang telah dibuat.

##### 5.1.1 Implementasi *Haar Cascade Classifier*

Pada bagian ini tiap subcitra akan diklasifikasi menggunakan satu fitur. Hasil dari klasifikasi pertama ini berupa *Detect* untuk objek yang memenuhi fitur Haar tertentu dan tidak mendeteksi apabila tidak dikenali. Dengan banyaknya tingkat klasifikasi, maka diperlukan *Training* yang lebih spesifik sehingga mendapatkan akurasi yang baik. Tahapan selanjutnya ialah menampilkan objek sampel gambar yang telah terdeteksi, dengan memberi tanda bujur sangkar.

Berikut ditampilkan hasil program perancangan sistem manajemen kehadiran karyawan menggunakan *face recognition* :



**Gambar 5.1** Pendeteksian Menggunakan Haar Cascade Classifier

### **5.1.2 Implementasi Program Absensi Jika Tidak Mendeteksi Wajah**

Berikut ditampilkan hasil program perancangan sistem manajemen kehadiran karyawan menggunakan *face recognition* :



**Gambar 5. 2 Hasil Program Pada Saat Tidak Mendeteksi Objek**

### **5.1.3 Tampilan Program Absensi Wajah**

Tampilan program dibuat sekomplek mungkin karena program ini bertujuan untuk mendeteksi wajah dan plat kendaraan sehingga banyak sekali code yang dituliskan dan mendapatkan akurasi yang baik ketika program dijalankan. Berikut tampilan program dapat dilihat pada gambar 5.3

```
py AbsensiWajah.py X
AbsensiWajah.py > VideoThread > run
# membaca kamera
ret, frame = cap.read()
if ret:
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # deteksi wajah
    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor=1.1,
        minNeighbors=5,
        minSize=(60, 60),
        flags=cv2.CASCADE_SCALE_IMAGE,
    )

    # lokasi wajah
    for x, y, w, h in faces:
        cv2.rectangle(frame, (x, y), (x + w, y + h),
            (0, 255, 0), 2)

    # jika wajah ditemukan
    if len(faces) > 0:
        rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        encodings = face_recognition.face_encodings(rgb)
        names = []
        for encoding in encodings:

            # pengenalan wajah
            matches = face_recognition.compare_faces(
                self.knownEncodings, encoding
            )
            name = "Unknown"
            if True in matches:
                matchedIdxs = [
                    i for (i, b) in enumerate(matches) if b]
                counts = {}
                for i in matchedIdxs:
                    name = self.knownNames[i]
                    counts[name] = counts.get(name, 0) + 1
                name = max(counts, key=counts.get)
                names.append(name)
```

**Gambar 5. 3 Program Pendeteksi Wajah**

#### **5.1.4 Program Pengenalan Wajah**

Program ini digunakan untuk mengidentifikasi wajah yang dideteksi dengan mencocokkan encoding wajah yang dideteksi dengan encoding wajah yang dikenal (`self.knownEncodings`). `matches=face_recognition.compare_faces(self.knownEncodings,encoding)` digunakan untuk membandingkan encoding wajah yang dideteksi (`encoding`) dengan encoding wajah yang dikenal (`self.knownEncodings`). Fungsi ini akan mengembalikan array of boolean yang menunjukkan hasil perbandingan

antara setiap encoding yang dikenal dengan encoding yang dideteksi. Jika sebuah wajah cocok dengan salah satu wajah yang dikenal, maka elemen yang sesuai dalam array akan bernilai True.

`name = "Unknown"` digunakan untuk menetapkan nama default sebagai "Unknown" jika wajah yang dideteksi tidak cocok dengan wajah yang dikenal. `if True in matches:` digunakan untuk memeriksa apakah ada wajah yang cocok dengan wajah yang dikenal.

`matchedIdxs = [i for (i, b) in enumerate(matches) if b]` digunakan untuk mengambil indeks dari wajah yang cocok dalam array `matches`. `counts = {}` digunakan untuk menyimpan jumlah wajah yang cocok dengan setiap nama yang dikenal.

`for i in matchedIdxs:` digunakan untuk iterasi setiap indeks wajah yang cocok. `name = self.knownNames[i]` digunakan untuk mengambil nama dari wajah yang cocok dari array `self.knownNames`. `counts[name] = counts.get(name, 0) + 1` digunakan untuk menambahkan satu ke jumlah wajah yang cocok dengan nama yang sesuai.

`name = max(counts, key=counts.get)` digunakan untuk menentukan nama yang muncul paling banyak dari wajah yang cocok. `names.append(name)` digunakan untuk menambahkan nama yang ditentukan ke dalam array `names`. Pada akhirnya, jika wajah dideteksi cocok dengan wajah yang dikenal, maka nama yang sesuai akan ditambahkan ke dalam array `names`. Jika tidak cocok, maka nama "Unknown" akan ditambahkan ke dalam array `names`.

```
matches = face_recognition.compare_faces(  
    self.knownEncodings, encoding  
)  
name = "Unknown"  
if True in matches:  
    matchedIdxs = [  
        i for (i, b) in enumerate(matches) if b]  
    counts = {}  
    for i in matchedIdxs:  
        name = self.knownNames[i]  
        counts[name] = counts.get(name, 0) + 1  
    name = max(counts, key=counts.get)  
    names.append(name)
```

**Gambar 5. 4 Program Pengenalan Wajah**

## 5.2 PENGUJIAN KEAKURATAN SCAN WAJAH ABSENSI

Pengujian ini dilakukan dengan tujuan agar melihat seberapa akurat kamera dapat mendeteksi wajah dengan baik saat menggunakan *Haar Cascade Classifier*. Berikut hasil pengujian dapat dilihat pada table 5.1

**Table 5.1 Pengujian akurasi deteksi**

Banyak Percobaan	Jarak (cm)	Mengenali		Akurasi terdeteksi
		ya	Tidak	
1.	30 cm	Ya		96%
2.	60 cm	Ya		87%
3.	90 cm	Ya		79%
4.	120 cm	Ya		75%
5.	150 cm		tidak	65%

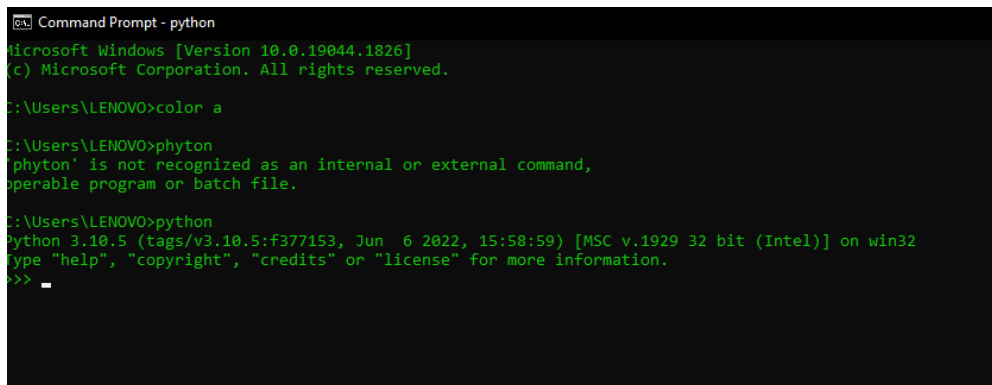
Pada tabel 5.1 dapat dilihat bahwa dari hasil 5 kali percobaan menunjukkan bahwa hanya pada jarak 30cm tingkat akurasi hampir mencapai 100% untuk melakukan deteksi wajah.

## 5.3 PENGUJIAN SISTEM

Pengujian sistem bertujuan untuk memastikan apakah semua fungsi sistem bekerja dengan baik dan mencari kesalahan yang mungkin terjadi. Pada pengujian sistem presensi karyawan.

### 5.2.1 Python

Dalam sistem manajemen kehadiran karyawan, peneliti menggunakan bahasa pemrograman *python*.



```
Command Prompt - python
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LENOVO>color a

C:\Users\LENOVO>pyhton
'pyhton' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\LENOVO>python
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 15:58:59) [MSC v.1929 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>> -
```

**Gambar 5.5 Menjalankan Python Pada Terminal**

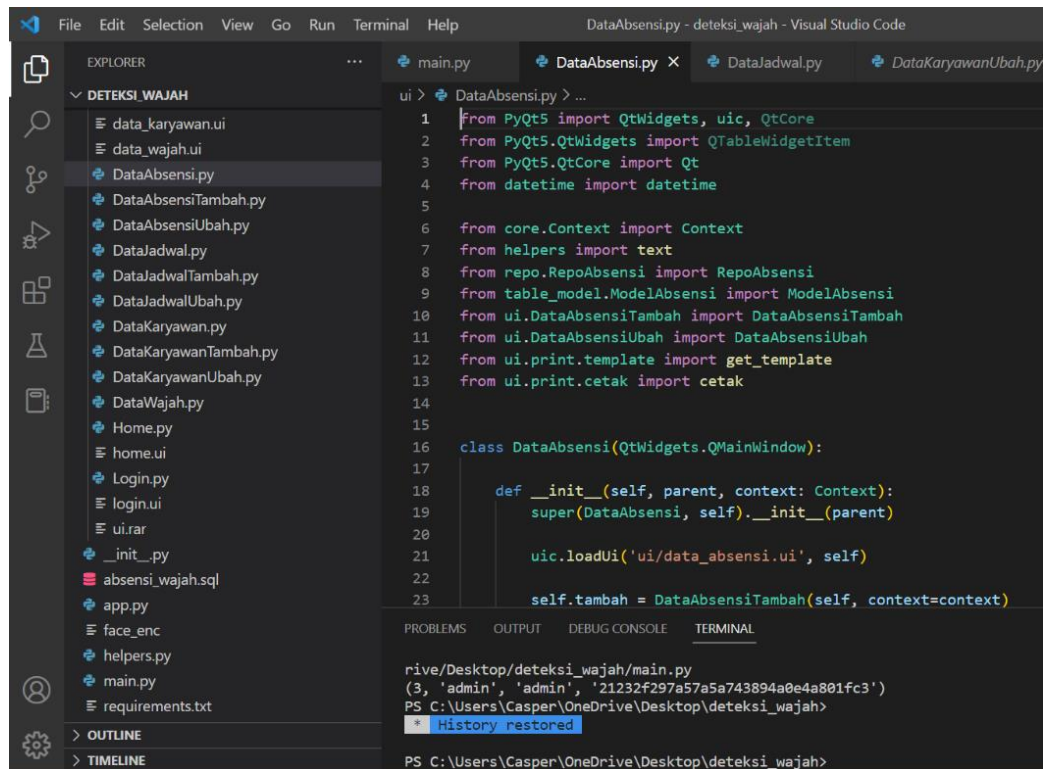
Untuk pengujian, dapat dimulai dari menjalankan *python* melalui terminal dengan mengetik *python* kemudian *enter* seperti pada gambar 5.4 :

Jika tampilannya sudah seperti pada gambar tersebut, berarti *python* telah berhasil diinstall. Selanjutnya menginstall semua library yang dibutuhkan dalam penelitian ini yaitu, Open CV, Numpy, Tesseract, Serial, Time.

### 5.2.2 Visual Studio Code

Setelah semua *Library* telah berhasil diinstall, selanjutnya dimulai penulisan kode sistem pendeteksian objek menggunakan text editor *Visual Studio Code*, buka VSCode dan buat folder baru kemudian simpan dengan nama *deteksi\_wajah* seperti pada gambar 5.6.



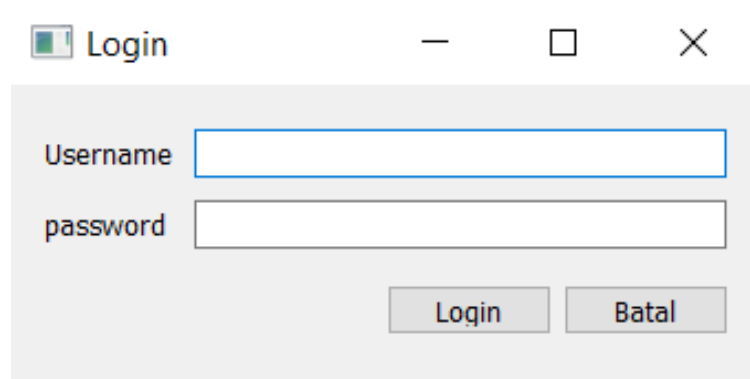


**Gambar 5. 6 Tampilan Visual Studio Code**

### 5.2.3 Tampilan From Login Admin

Halaman yang ditampilkan pada aplikasi ketika pertama kali dibuka, terdapat 1 user yang dapat melakukan login yaitu admin. Saat admin menekan button login, dan admin sudah memasukkan username dan password dengan benar, maka sistem akan menampilkan halaman daftarkan data wajah karyawan.

Pada tahap ini peneliti melakukan pengujian pada program dengan cara menjalankan program yang telah dibuat seperti gambar 5.7 :



**Gambar 5. 7 Tampilan Login**

#### **5.2.4 Tampilan Form Menu Utama Admin**

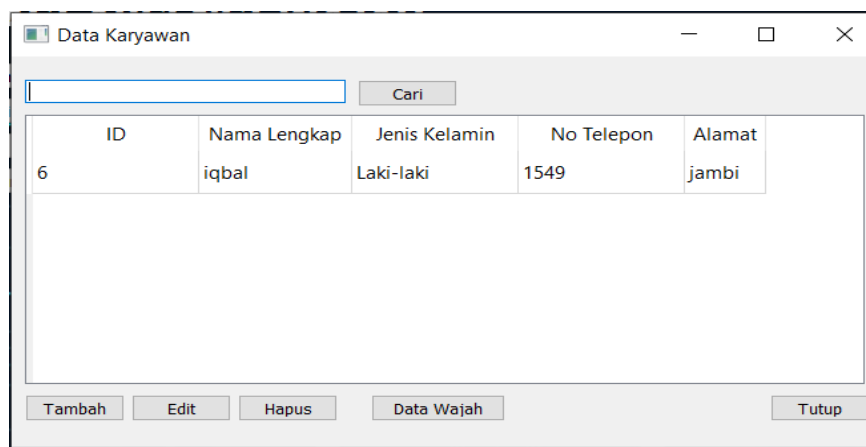
Form ini merupakan perancangan tampilan menu utama untuk admin. Pada tampilan ini terdapat sebuah informasi serta empat pilihan menu yaitu, pilihan Absen wajah, pilihan Data absensi, Data jadwal dan Data karyawan. Adapun formnya dapat pada gambar 5.8 berikut ini :



**Gambar 5. 8 Tampilan Menu Utama Admin**

### 5.2.5 Tampilan Form Data Karyawan

Form merupakan halaman yang ditampilkan pada aplikasi ketika admin memilih pilihan data karyawan pada tampilan antarmuka menu utama, halaman ini memuat fitur kelola data wajah dan data karyawan. fitur yang dimiliki antara lain fitur daftar data karyawan dan fitur daftarkan data set wajah. Adapun bentuk formnya seperti gambar 5.9 berikut ini:



The screenshot shows a window titled "Data Karyawan" with a search bar and a "Cari" button. Below is a table with the following data:

ID	Nama Lengkap	Jenis Kelamin	No Telepon	Alamat
6	iqbal	Laki-laki	1549	jambi

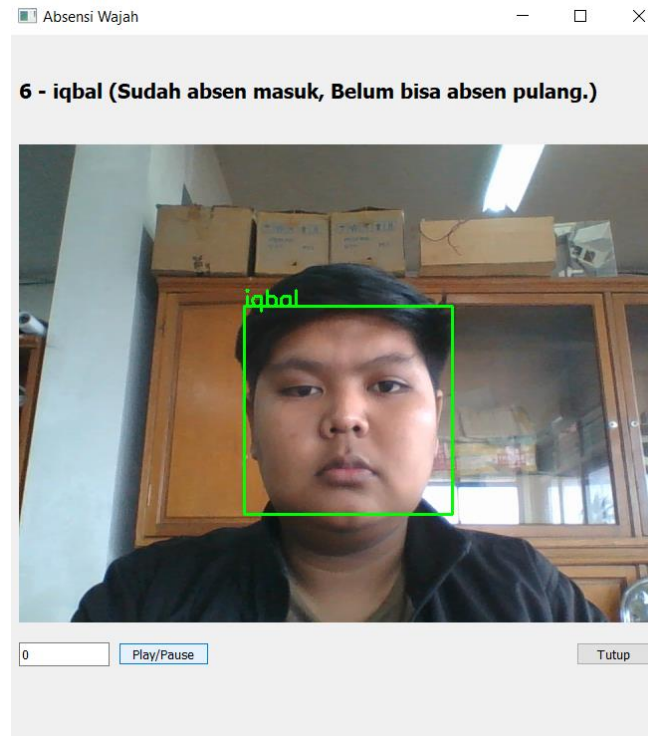
At the bottom of the window are buttons for "Tambah", "Edit", "Hapus", "Data Wajah", and "Tutup".

**Gambar 5. 9 Tampilan Menu Data Karyawan**

Pada gambar diatas user atau admin dapat melakukan penambahan data untuk karyawan dan melakukan registrasi wajah pada data wajah.

### 5.2.6 Tampilan Form Menu Absensi Kehadiran

Pada *form* tampilan pada gambar 5.10 ini merupakan tampilan ketika terdapat wajah yang terdeteksi dan dikenali oleh aplikasi. Pada *form* ini menampilkan Nama karyawan, keterangan kehadiran karyawan pada saat melakukan absensi terlambat atau tidak dan juga ditampilkan progres bar untuk menunjukkan sejauh mana proses pengenalan wajah dilakukan oleh aplikasi. Untuk menu absen wajah yang bisa dilihat pada gambar 5.10 :



**Gambar 5. 10 Menu Absensi Wajah**

### **5.2.7 Tampilan Form Laporan Presensi/ Kehadiran Jam kerja**

Pada gambar 5.11 merupakan form tampilan laporan presensi jam kerja karyawan. Pada form ini menampilkan filter tanggal, filter nama karyawan, filter absen masuk, absen pulang, filter status kehadiran dan buttonprint untuk mencetak laporan presensi pegawai.

Tanggal	Nama Karyawan	Absen Masuk	Absen Pulang	Status
2023-01-19	6 - iqbal	15:14:00	0:00:00	Terlambat
2023-01-18	6 - iqbal	14:41:00	18:41:00	Terlambat

**Gambar 5. 11 Laporan Presensi Karyawan**

### 5.3 ANALISIS SISTEM SECARA KESELURUHAN

Untuk mendeteksi kemungkinan terjadi kesalahan setelah uji coba, maka perlu dilakukan analisa rangkaian secara keseluruhan. Dari semua proses yang telah dilakukan, baik pengujian *hardware* maupun *software*, bisa dikatakan bahwa alat ini berfungsi dengan baik seperti apa yang diinginkan oleh peneliti.

Penelitian ini dilakukan guna mencoba apakah algoritma ini dapat digunakan untuk sistem manajemen kehadiran karyawan dan mencari tahu apakah dengan menggunakan metode ini dapat mendeteksi wajah dan plat secara bersamaan dan akurat. Penelitian ini dilakukan dengan cara sebagai berikut :

1. Memasang *camera webcam*
2. Menjalankan Program *Python* yang telah peneliti buat.
3. Kemudian melakukan presensi dengan mengarahkan wajah pada *camera webcam* untuk absen masuk atau pulang.