

BAB III

METODE PENELITIAN

3.1. Objek Penelitian

Penelitian ini menggunakan objek gambar wajah Mahasiswa angkatan IF16 program studi Teknik Informatika Universitas Buana Perjuangan Karawang. Pengambilan data dilakukan dengan posisi wajah menghadap kamera dengan jarak $\pm 50\text{cm}$ sebanyak 100 buah dan dengan berbagai ekspresi wajah seperti senang, sedih, tertawa, dan marah dengan keadaan pada siang hari pada ruangan cukup pencahayaan.

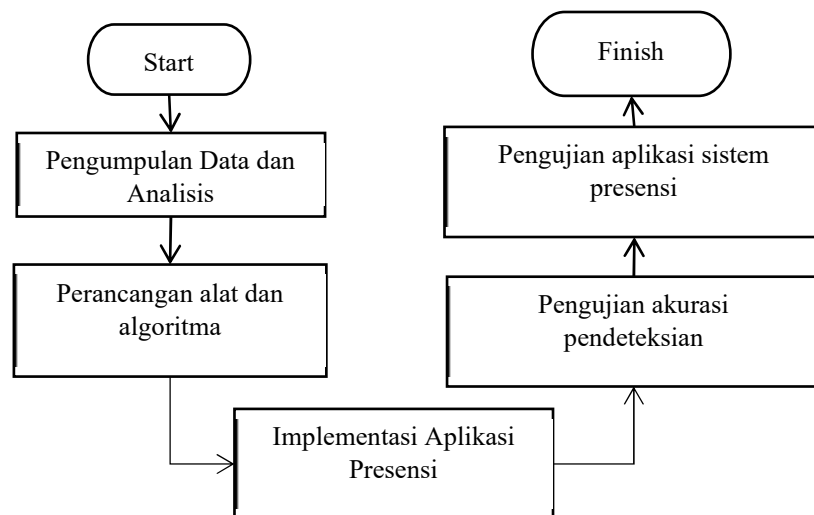
Lokasi dan waktu penelitian ini dilakukan di ruang kelas Teknik Informatika. Waktu penelitian Desember 2019.

Tabel 3.1 Tabel Rencana Penelitian

No	Kegiatan	Bulan					
		1	2	3	4	5	6
1.	Studi Literatur						
2.	Analisis Kebutuhan						
3.	Perancangan Sistem						
4.	Implementasi						
5.	Pengujian						

3.2. Prosedur Penelitian

Tahapan penelitian pada penelitian ini dilakukan terdiri dari beberapa tahap dimulai dengan menganalisis dan mengumpulkan data, perancangan sistem, implementasi, pengujian, dan evaluasi.



Gambar 3.1. Prosedur Penelitian

Penelitian diawali dengan pengumpulan referensi (studi literatur) yang berkaitan dengan sistem presensi pengenalan wajah, kemudian berdasarkan referensi tersebut dibuat aplikasi deteksi dan pengenalan wajah yang dibuat untuk menguji akurasi dari sistem presensi yang akan dibuat, setelah pengujian aplikasi selesai dilaksanakan sistem presensi berbasis pengenalan wajah kemudian dibuat untuk akhirnya diujicobakan di Universitas Buana Perjuangan Karawang.

Alur tahapan prosedur penelitian, diantaranya :

a. Pengumpulan Data dan Analisis

Proses pengumpulan data yang dilakukan menggunakan laptop sebagai alat pemrosesan serta kamera *webcam* sebagai alat penginputan data. Dengan menggunakan pemrograman *python* dengan *library cv2*, cara mengakses webcam pada laptop dapat menggunakan perintah sebagai berikut:

```
cam = cv2.VideoCapture(0)
```

Selanjutnya adalah proses menampilkan objek citra pada posisi *webcam* dapat menggunakan perintah sebagai berikut:

```
cv2.imshow('Capturing', img)
```



Gambar 3.2 Objek gambar yang tertangkap oleh webcam

Kemudian dengan bantuan dari algoritma *haarcascade* dapat mempermudah pendeteksian objek wajah dari sebuah citra. Sehingga data wajah akan tersimpan secara otomatis dan menjadi sebuah dataset. Untuk melakukannya dapat menggunakan perintah sebagai berikut:

```
harcascadePath = "haarcascade_frontalface_default.xml"
detector = cv2.CascadeClassifier(harcascadePath)
```

Sebelum disimpan citra wajah akan diubah menjadi citra *grayscale* terlebih dahulu untuk memudahkan proses selanjutnya. Berikut adalah baris perintahnya:

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
faces = detector.detectMultiScale(gray, 1.3, 5)
```



Gambar 3.3 Objek Gambar Hasil *Grayscale*

Selanjutnya adalah citra akan diubah kedalam matrik dengan algoritma *eigenface*. Berikut adalah perintahnya:

```
recognizer = cv2.face_LBPHFaceRecognizer.create()
recognizer.train(faces, np.array(Id))
recognizer.save(file.yml)
```

Selanjutnya adalah proses pengenalan dan pedeteksian objek. Berikut adalah perintahnya:

```
recognizer = cv2.face_LBPHFaceRecognizer.create()
recognizer.read(file.yml")
```

Terakhir adalah proses pembersihan tampilan. Perintah untuk melakukan proses pembersihan kembali tampilan adalah sebagai berikut:

```
cam.release()
cv2.destroyAllWindows()
```

b. Perancangan Aplikasi Deteksi dan Pengenalan Wajah

Tahapan ini berisikan perancangan aplikasi mulai dari bahan yang dibutuhkan dalam penelitian hingga desain *user interface* dari aplikasi yang akan dibangun.

Adapun bahan yang penulis gunakan:

1. *Hardware* :

- a. Laptop Merk ASUS A450C dengan spesifikasi prosesor Intel© Core™ i3-3217U CPU @ 1.80GHz × 2, Memory RAM 7.7 GB, Harddisk 162.9 GB, Grafik card Intel Corporation 3rd Gen Core processor Graphics Controller, Dimensi layar 14 inch resolusi 1366 x 768.
- b. Webcam Full HD 1080 dengan spesifikasi resolusi 1920 x1080, 110° wide angle, built-in microphone, adjustable Focus Lens, plug-and-play, USB connectivity, automatic Brightness Adjustment, frame Rate: 30 fps.

2. *Software* :

- a. OS Linux Mint 19.3 Cinnamon versi 4.4.8 linux cernel 5.0.0-32-generic
- b. Algoritma *Eigenface* & *Haar Cascade*
- c. Bahasa pemrograman Python version 2.7.17
- d. Library OpenCV

Adapun desain *interface* sendiri memiliki beberapa tahapan sebagai berikut:

1. Perancangan pada perangkat keras

Penelitian ini menggunakan laptop sebagai alat pemrosesannya yang terhubung ke *webcam* sebagai media inputannya. Hasil dari pendeteksian akan ditampilkan pada layar laptop dan akan tersimpan/terrecord sebagai file .csv. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini.



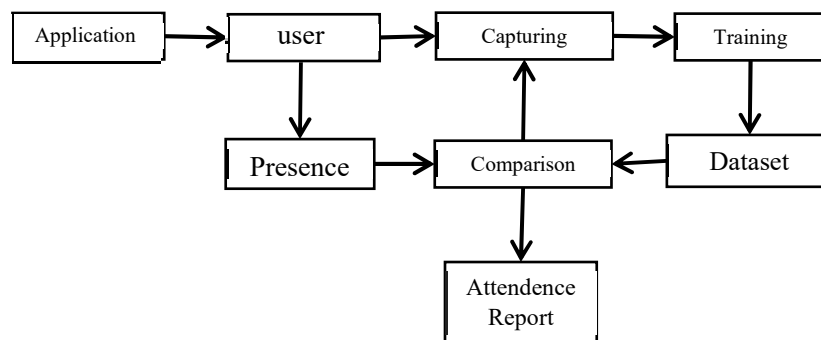
Gambar 3.4 Skema Perancangan Perangkat Keras

Fungsi komponen pada skema diatas:

- Webcam* sebagai media penginputan data berupa data wajah.
- Laptop berfungsi sebagai alat pemrosesan data dari *webcam*.
- Report* berfungsi sebagai media penyimpanan hasil *record* pemrosesan data dari laptop yang sudah terbentuk kedalam sebuah laporan berextensi .CSV.

2. Perancangan pada Perangkat Lunak

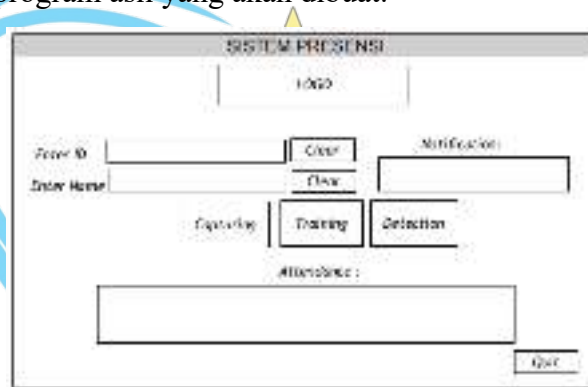
Bahasa pemrograman pada penelitian ini menggunakan bahasa pemrograman *python*. Berikut adalah diagram blok perancangan perangkat lunak.



Gambar 3.5 Diagram Block Perancangan Perangkat Lunak

Aplikasi ini memiliki dua buah proses, proses yang pertama adalah proses *capturing* atau proses pengambilan citra wajah dengan dibantu *haar cascade* untuk diambil kemudian disimpan kedalam dataset yang kemudian dapat digunakan sebagai data acuan pengenalan wajah. Dataset hasil *capturing* sebelum disimpan telah dirubah terlebih dahulu kedalam bentuk *grayscale*. proses yang kedua adalah Proses Presensi dimana pada proses ini *user* akan dideteksi dan disimpan/*record* kehadirannya dengan cara mencocokkan wajah *user* dengan data wajah yang sudah tersimpan sebelumnya pada *dataset*.

Desain *interface* ini mendeskripsikan desain antarmuka program atau aplikasi yang akan dibuat. Desainnya sendiri hanya berupa *mock-up* atau tiruan dari program asli yang akan dibuat.



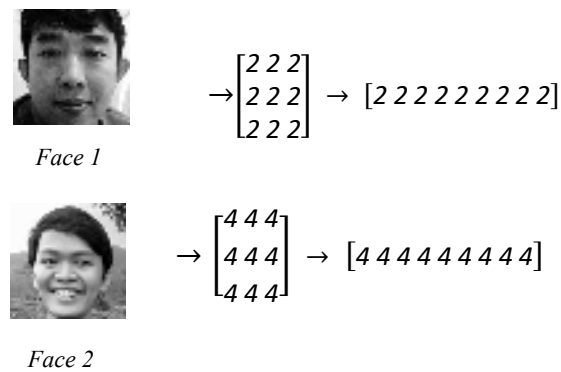
Gambar 3.6 Desain *Interface* Sistem

c. Implementasi aplikasi presensi

Proses pendeteksian wajah dengan menggunakan metode *eigenface* dilakukan dengan beberapa tahap berikut:

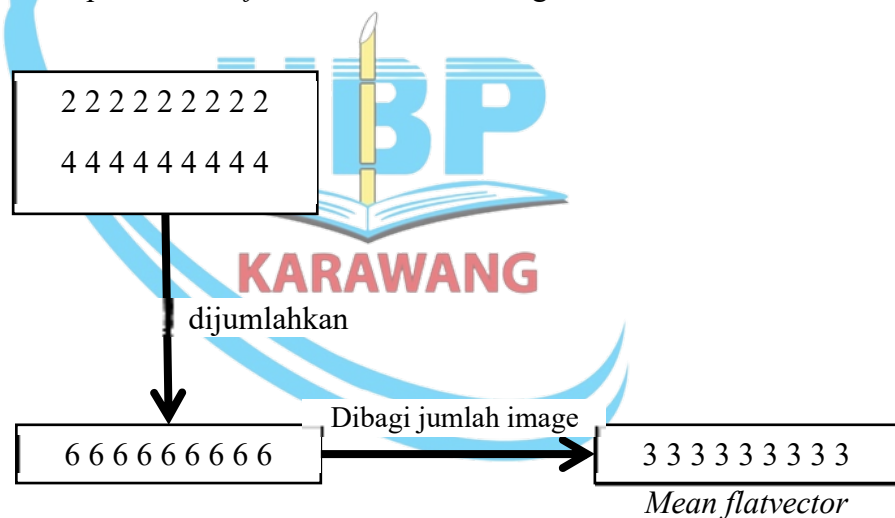
1. Penyusunan *flatvector*

Pertama ialah menyusun seluruh *image training* menjadi satu matriks tunggal. Misalnya *image* yang kita simpan berukuran HxW piksel dan jumlahnya N buah, maka kita memiliki *flatvector* dengan dimensi Nx(HxN). Jika didalam *image training* terdapat 2 *image* dengan ukuran 3x3 piksel maka kita akan memiliki *eigenvector* ukuran 2x9 untuk lebih jelasnya silahkan perhatikan berikut ini:

Gambar 3.7 Penyusunan *flatvector*

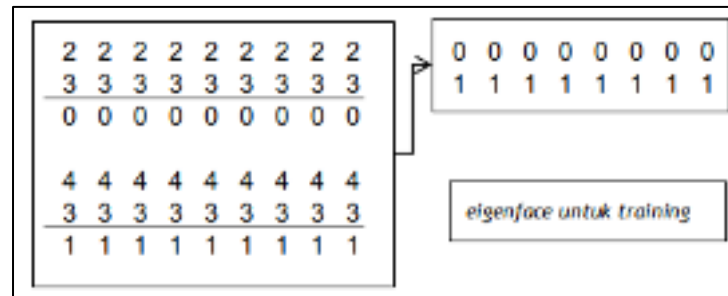
2. Menghitung *Mean Flatvector*

Setelah memperoleh *flatvector* kemudian dijumlah seluruh barisnya lalu diperoleh matriks ukuran $1 \times (H \times W)$ selanjutnya bagi dengan banyaknya *image* N untuk mendapatkan *mean flatvector*. Perhatikan gambar berikut ini.

Gambar 3.8 *Mean flatvector*

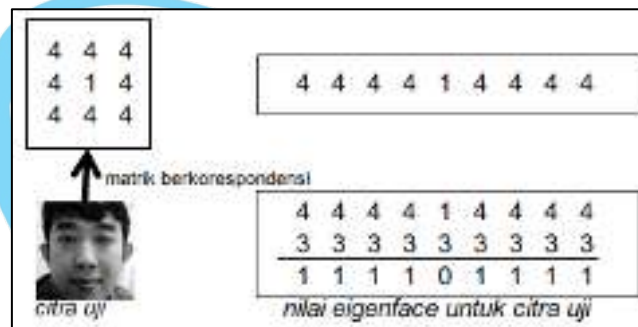
3. Menentukan Nilai *Eigenface*

Berikutnya kita akan menghitung nilai *eigenface* dengan menggunakan *mean flatvector* yang kita dapatkan sebelumnya. Dengan mengurangi baris-baris matriks *flatvector* dengan *mean flatvector*. Jika hasilnya < 0 ganti nilainya dengan 0.

Gambar 3.9 Nilai *Eigenface*

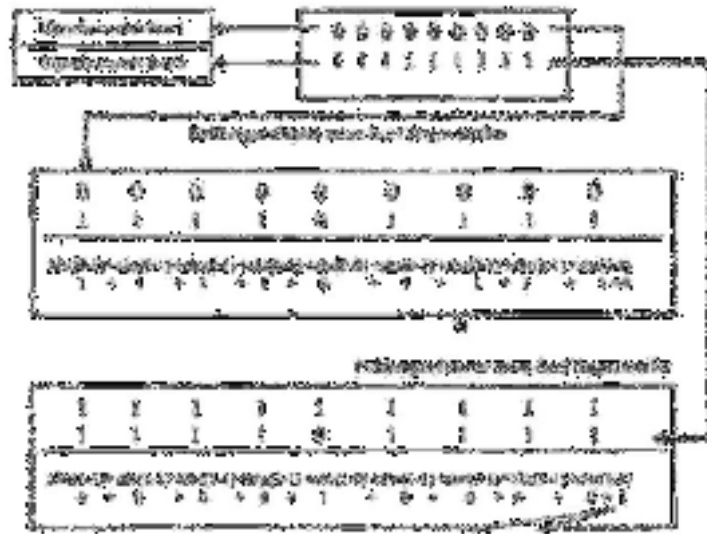
4. Proses Identifikasi

Langkah indentifikasi adalah langkah yang melakukan proses penghitungan nilai *eigenface* pada citra uji. Dengan cara yang sudah dijelaskan sebelumnya. Perhatikan gambar dibawah ini.



Gambar 3.10 Proses Identifikasi

Setelah nilai *eigenface* kita temukan, kita bisa menghitung nilai jarak terdekat dengan *eigenface* dengan *eigenvector image training*. dengan cara menentukan nilai absolut dari pengurangan baris ke 1 pada matriks *eigenface image training* dengan *eigenface* dari citra uji, selanjutnya jumlahkan elemen - elemen penyusun *vector* yang dihasilkan dari pengurangan tersebut sehingga diemukan jarak d indeks i . Lakukan pada semua baris, kemudian cari nilai d yang paling kecil.



Gambar 3.11 Perhitungan jarak.

d. Pengujian Akurasi Aplikasi Deteksi dan Pengenalan Wajah

Sebagai nilai acuan pada penelitian ini penulis menggunakan nilai akurasi. Untuk dapat melihat tingkat keberhasilan dalam pengujian sebagai bahan evaluasi. Nilai akurasi didapat dengan cara menghitung seluruh data yang dinyatakan benar dilakukan pembagian dengan seluruh jumlah data uji. Berikut adalah rumus penghitungan presentase pengujian adalah sebagai berikut:

$$\text{Persentase Pengujian} = \frac{\sum \text{sukses}}{\sum \text{Total uji}} \times 100\%$$

Parameter evaluasi yang akan dihitung persentase tingkat keberhasilannya dari tiap pengujian:

Tabel 3.2. Parameter Evaluasi

No.	Pengujian	Hasil		
		Total Uji	Sukses	Gagal
1	Citra wajah terdaftar dan tidak terdaftar			
2	Citra wajah menggunakan asesoris kacamata			
3	Citra wajah menggunakan asesoris topi			
4	Citra wajah dengan ekspresi normal			
5	Citra wajah dengan ekspresi senang			

-
- | | |
|----|--|
| 6 | Citra wajah dengan ekspresi sedih |
| 7 | Citra wajah dengan ekspresi marah |
| 8 | Citra wajah dengan ekspresi terkejut |
| 9 | Citra wajah dengan jarak dekat $\pm 30\text{cm}$ |
| 10 | Citra wajah dengan jarak jauh $\pm 60\text{cm}$ |
-

Selanjutnya penghitungan nilai persentase akurasi dengan rumus sebagai berikut :

$$\text{Persentase Akurasi} = \frac{\sum \% \text{ Pengujian}}{\sum \text{ Pengujian}} \times 100\%$$

