

BAB III

METODOLOGI PENELITIAN

3.1 Jenis dan Rancangan Penelitian

Rancangan penelitian bertujuan untuk mengidentifikasi masalah yang ada pada sistem yang akan dibuat oleh peneliti. Rancangan penelitian meliputi kebutuhan perangkat lunak, kebutuhan fungsional, dan kebutuhan non fungsional.

3.1.1 Spesifikasi Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak dibagi menjadi dua, yaitu spesifikasi kebutuhan perangkat lunak fungsional dan spesifikasi kebutuhan perangkat lunak non-fungsional

1. Spesifikasi kebutuhan perangkat lunak fungsional
 - a. Perangkat dapat mendeteksi area sekitar wajah secara baik
 - b. Perangkat dapat mengenali gambar wajah
 - c. Perangkat lunak dapat memproses data wajah menggunakan metode eigenface
2. Spesifikasi kebutuhan perangkat lunak non-fungsional
 - a. Pengguna perangkat lunak menggunakan wajahnya sebagai media deteksi wajah.
 - b. Kondisi cahaya harus memungkinkan atau terang
 - c. Kamera dan komputer berperan sebagai perangkat keras dalam proses pengolahan citra
 - d. Perangkat lunak dijalankan menggunakan system operasi MacOS

3.1.2 Analisa Kebutuhan Fungsional

Analisa kebutuhan fungsional akan membantu alur dan konstruksi perangkat lunak. Dalam perangkat lunak yang dibangun digunakan pengenalan wajah menggunakan metode pendekatan berorientasi objek.

Penulis penelitian ini membuat analisis kebutuhan fungsional dan berikut ini rincian setiap desain berdasarkan pendekatan berorientasi objek:

1. Pengguna menjalankan aplikasi pendeteksi dan pengenalan wajah lalu menekan form absensi maka form absensi akan ditampilkan oleh sistem.

2. Pengguna yang telah menekan form absensi melakukan pendeteksian dengan wajah yang menghadap ke kamera dengan tegap maka system akan menampilkan kotak areal pendeteksian wajah.
3. Pengguna yang telah terdeteksi maka informasi data pemilik akan ditampilkan oleh sistem pendeteksian wajah.

3.1.3 Analisa Kebutuhan Non-Fungsional

Analisa kebutuhan non-fungsional ini berfungsi untuk menjelaskan beberapa pendukung sistem yang akan dibangun. Adapun kebutuhan-kebutuhan yang diperlukan untuk mendukung kinerja dari perangkat lunak yaitu:

1. Analisa kebutuhan perangkat keras

Beberapa perangkat keras yang digunakan dalam membangun dan menjalankan sistem ini adalah sebagai berikut:

- PC

Komputer yang digunakan dalam membangun aplikasi adalah laptop dengan spesifikasi sebagai berikut:

- Processor : 1,4 GHz Quad-Core Intel Core i5
- RAM : 8 GB
- VGA : Intel Iris Plus Graphics 645 1536 MB
- SSD : 256 GB

- Kamera

Kamera yang digunakan untuk mengambil gambar adalah:

- Kamera Webcam 720p FaceTime HD Built-in
- Kecepatan frame 60fps
- Mampu menampilkan kualitas video HD 720p

2. Analisa kebutuhan perangkat lunak

Perangkat lunak yang digunakan untuk membangun sistem adalah sebagai berikut:

- Sistem Operasi

Sistem operasi yang digunakan peneliti adalah MacOS.

- *Sublime text*
Sublime text dipilih sebagai aplikasi pengkodean karena dapat menjalankan berbagai macam bahasa pemrograman.
- PHP
PHP merupakan bahasa pemrograman *server-side* untuk pembuatan aplikasi berbasis website.
- *Tensorflow*
Tensorflow digunakan untuk pendeteksian objek dan mempermudah proses *constructing, training* serta *deployment*.

3.2 Subjek Penelitian

Pihak atau partisipan penelitian digunakan sebagai sampel dalam suatu penelitian. Subyek penelitian juga dibahas, beserta deskripsi metodologi demografi, sampel, dan pengambilan sampel (acak/non-acak). Tiga tingkatan subjek penelitian, yaitu:

- 1) Partisipan penelitian mikro merupakan skala terkecil dan hanya berbentuk individu.
- 2) Meso adalah topik tingkat penelitian yang memiliki lebih banyak peserta, seperti keluarga dan kelompok.
- 3) Makro adalah topik tingkat penelitian dengan jumlah peserta yang banyak, termasuk masyarakat atau komunitas yang lebih besar.

Tanggung jawab subjek penelitian adalah menanggapi dan memberikan informasi mengenai data yang dibutuhkan peneliti, serta memberikan masukan kepada peneliti baik secara langsung maupun tidak langsung. Karena peserta dalam penelitian ini termasuk kelompok yang lebih besar, tingkat meso digunakan untuk penelitian ini. Mahasiswa IIB Darmajaya dan dosen program studi Teknik informatika IIB Darmajaya menjadi subjek pada penelitian ini.

3.2.1 Sampel/Populasi

Generalisasi yang dikenal sebagai "populasi" mengacu pada hal-hal atau orang-orang dengan jumlah dan sifat tertentu yang telah dipilih peneliti untuk

dipelajari dan kemudian diekstrapolasikan. Populasi suatu objek atau subjek mencakup semua ciri dan sifat-sifatnya, bukan hanya kuantitas objek atau subjek yang diteliti. Populasi penelitian yang digunakan adalah mahasiswa Prodi Teknik Informatika IIB Darmajaya.

Sampel mewakili representasi ukuran populasi dan susunannya. Jika populasinya besar dan peneliti tidak dapat menyelidiki seluruh populasi karena kekurangan sumber daya, orang, atau waktu, peneliti dapat menggunakan sampel dari populasi. Beberapa mahasiswa Prodi Teknik Informatika IIB Darmajaya menjadi sampel penelitian.

3.2.2 Teknik Pengambilan Sampel

Metode yang digunakan dalam pengumpulan data pada penelitian ini adalah sebagai berikut:

1. Pengamatan (Observasi)

Pengumpulan data dengan mengamati dan mencatat segala sesuatu yang dibutuhkan guna kebutuhan penelitian absensi menggunakan face recognition

2. Wawancara (Interview)

Metode ini dilakukan dua arah dengan tanya jawab antara peneliti dan juga narasumber untuk mengetahui permasalahan yang dihadapi objek penelitian. Pada proses wawancara ini peneliti melakukan tanya jawab dengan dosen dan juga mahasiswa terkait sistem absensi di IIB Darmajaya

3. Tinjauan Pustaka

Peneliti mengumpulkan data dari buku, jurnal dan artikel yang terkait dengan penelitian saat ini sebagai sumber pendukung dalam menyelesaikan penelitian.

3.3 Variabel Penelitian

Variabel penelitian merupakan suatu informasi yang nilainya tidak tetap. Dalam penelitian, terdapat klasifikasi variabel yang akan digunakan untuk mempersiapkan alat dan metode pengumpulan data, metode analisis atau pengolahan data, serta untuk melakukan pengujian. Karena penelitian ini

bersifat kualitatif, maka variabel penelitiannya adalah orang atau aktor yang berperan dalam sistem yang berjalan. Dan aktor yang dimaksud antara lain adalah mahasiswa IIB Darmajaya dan dosen program studi Teknik Informatika IIB Darmajaya.

Algoritma *Eigenface* merupakan satu di antara algoritma *face recognition* yang berdasarkan pada *Principle Component Analysis* (PCA). Prinsip dasar dari *face recognition* ialah dengan mengutip informasi unik wajah tersebut kemudian di-*encode* dan dibandingkan dengan hasil *de-code* yang sebelumnya dilakukan.

Algoritma *eigenface* secara umum cukup sederhana. Matriks himpunan ($\Gamma_1, \Gamma_2, \dots, \Gamma_M$) yang merepresentasikan Image matriks (Γ). Cari nilai rata-rata (Ψ) untuk mengekstrak *eigenvector* (v) dan *eigenvalue* (λ) dari matriks himpunan. Untuk menentukan nilai *eigenface* gambar, gunakan nilai *eigenvector*.

Berikut adalah tahapan perhitungan *Eigenface*:

- 1) Langkah pertama adalah menyiapkan data dengan membuat suatu himpunan S yang terdiri dari seluruh *training image*.

$$S = (\Gamma_1, \Gamma_2, \dots, \Gamma_M)$$

- 2) Langkah kedua adalah ambil nilai tengah atau *mean* (Ψ)

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n = 1$$

- 3) Langkah ketiga, kemudian cari selisih (Φ) antara nilai *training image* (Γ_i) dengan nilai tengah

$$\Phi_i = \Gamma_i - \Psi$$

- 4) Langkah keempat adalah menghitung nilai matriks *koyarian* (C)

$$C = \frac{1}{M} \sum_{n=1}^M \phi_n \phi_n^T = AA^T$$

$$L = A^T A \quad L = \phi_m^T \phi_n$$

- 5) Langkah kelima menghitung *eigenvalue* (λ) dan *eigenvector* (v) dari matriks *koyarian* (C)

$$C \times v_i = \lambda_i \times v_i$$

- 6) Langkah keenam, setelah *eigenvector* (v) diperoleh, maka *eigenface* (μ) dapat dicari dengan persamaan

$$\mu_l = \sum_{k=1}^M v_{ik} \phi_i$$

$$l = 1, \dots, M$$

Berikut adalah tahapan pengenalan wajah:

- 1) Sebuah *image* wajah baru atau *test face* (Γ_{new}) akan dicoba untuk dikenali, pertama terapkan cara pada tahapan pertama perhitungan *Eigenface* untuk menapatkan nilai eigen dari *image* tersebut.

$$\mu_{new} = v \times (\Gamma_{new} - \psi)$$

$$\Omega = [\mu_1, \mu_2, \dots, \mu_M]$$

- 2) Gunakan metode *Euclidean Distance* untuk mencari jarak (*distance*) terpendek antara nilai *eigen* dari *training image* dalam database dengan nilai *eigen* dari *image testface*.

$$\varepsilon_k = \|\Omega - \Omega_k\|$$

Eigenface memiliki beberapa kelebihan yang membuat metodenya lebih unggul dibandingkan metode pengenalan wajah lainnya, seperti:

1. Efisiensi Kalkulasi: *Eigenface* menggunakan teknik transformasi untuk mengubah wajah menjadi feature yang lebih sederhana dan mudah dikenali. Ini membuat kalkulasi lebih efisien dan cepat dibandingkan dengan metode pengenalan wajah lainnya.
2. Fleksibilitas: *Eigenface* dapat menangani variasi wajah seperti perubahan sudut, iluminasi, dan ekspresi. Hal ini membuat metodenya lebih fleksibel dan dapat bekerja dengan baik meskipun ada perubahan pada wajah yang diidentifikasi.
3. Robust terhadap Noise: *Eigenface* mampu meminimalisir efek dari noise atau gangguan pada gambar, sehingga metodenya lebih robust dan dapat memberikan hasil identifikasi yang akurat.
4. Skalabilitas: *Eigenface* memiliki skalabilitas yang baik, sehingga dapat bekerja dengan baik pada dataset yang besar atau kecil.

Oleh karena itu, *Eigenface* menjadi metode pengenalan wajah yang populer dan digunakan dalam aplikasi industri, seperti sistem pengenalan

wajah untuk pengontrol akses, pengenalan wajah untuk aplikasi multimedia, dan lain-lain.

3.4 Jadwal Penelitian

Dalam tahap ini penelitian memfokuskan pada penjadwalan pengerjaan penelitian, dan analisis kebutuhan perangkat lunak dan perangkat lunak yang digunakan untuk menjalankan aplikasi.

Pada penelitian ini terdapat beberapa proses yang harus dilakukan dari tahap *communication* hingga *implementation and testing*, maka dari itu di perlukan penjadwalan yang tepat agar penelitian ini dapat selesai tepat pada waktunya. Berikut penjadwalan penelitian berdasarkan aktivitas yang dilakukan dengan skala waktu 1 minggu.

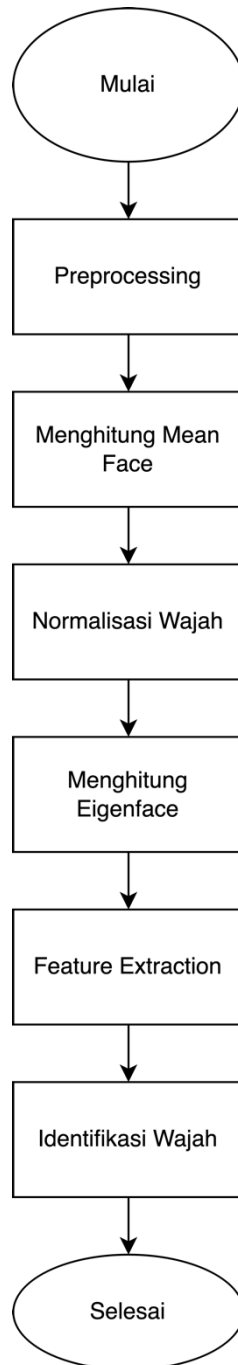
Tahap pengumpulan data, studi literatur, analisis masalah memerlukan waktu 1 minggu untuk penyelesaiannya. Analisa metode *eigenface* memerlukan waktu 2 minggu untuk mempelajari metode tersebut. Melakukan Analisa spesifikasi *hardware* dan *software* yang digunakan dalam membangun aplikasui membutuhkan waktu 1 minggu. Penjadwalan dilakukan penelitian selama 2 minggu. Perancangan *flowchart* membutuhkan waktu 3 minggu untuk mebuat alur sitem yang akan dibuat. Perancangan *user interface* membutuhkan waktu 1 minggu. Pengkodean program dilakukan selama 1 minggu dan pengujian aplikasi dilakukan selama 1 minggu. Jadwat tersebut dapat dilihat pada tabel 3.1.

3.5 Perancangan

Dalam tahap ini perancangan aplikasi meliputi perancangan *flowchart* sederhana pada setiap *user* yang akan di bangun.

3.5.1 Flowchart Eigenface

Berikut ini merupakan tahapan dasar dalam identifikasi wajah menggunakan metode eigenface, yang ditunjukkan pada gambar 3.1.



Gambar 3.1 Flowchart Dasar Eigenface

1. *Preprocessing*:
 - Membaca *dataset* wajah
 - Menyamakan ukuran wajah pada *dataset*
 - Menghapus ruang latar belakang
2. Menghitung *Mean Face*:
 - Menghitung rata-rata dari setiap pixel pada semua wajah dalam *dataset*
 - Menyimpan *Mean Face* sebagai referensi untuk normalisasi
3. Normalisasi Wajah:
 - Mengurangi *Mean Face* dari setiap wajah pada dataset untuk membuat wajah *centered*
4. Menghitung *Eigenface*:
 - Menghitung *covariance* matrix dari wajah *centered*
 - Menghitung *eigenvector* dan *eigenvalue* dari *covariance* matrix
 - Menyimpan *eigenvectors* dengan *eigenvalue* tertinggi sebagai *Eigenfaces*
5. Feature Extraction:
 - Menentukan jumlah *Eigenfaces* yang digunakan
 - Menghitung bobot setiap *Eigenface* untuk setiap wajah pada *dataset*
 - Menyimpan bobot *Eigenface* sebagai feature wajah
6. Identifikasi Wajah:
 - Membaca wajah yang akan diidentifikasi
 - Melakukan *preprocessing* dan normalisasi seperti pada tahap 2 dan 3
 - Menghitung bobot *Eigenface* baru untuk wajah yang akan diidentifikasi
 - Membandingkan bobot *Eigenface* baru dengan bobot *Eigenface* pada *dataset*
 - Menentukan wajah yang paling mirip dan memberikan hasil identifikasi.

3.5.2 Flowchart Admin

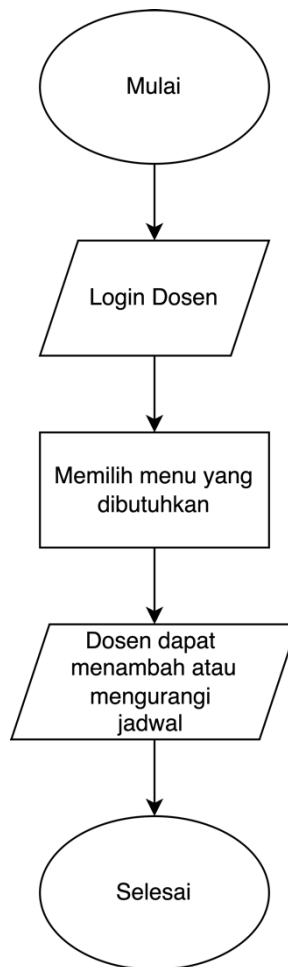
Flowchart ini dimaksudkan untuk mempermudah gambaran sistem sederhana yang akan dibangun untuk *user* admin untuk mengetahui sekilas fungsi apa yang akan diimplementasikan. Dapat dilihat pada gambar 3.2.



Gambar 3.2 Flowchart Admin

3.5.3 Flowchart Dosen

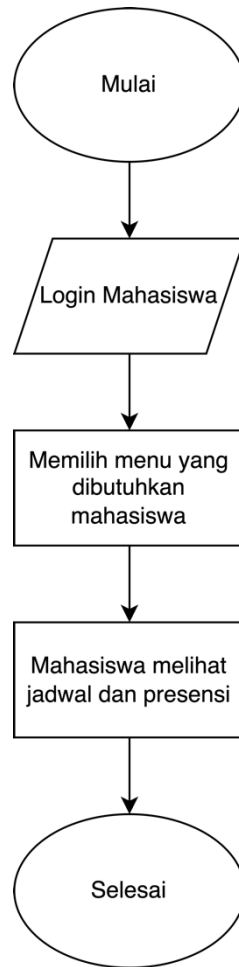
Flowchart dosen memiliki akses sistem yang lebih sedikit daripada admin, terdapat beberapa menu yang hanya diakses oleh dosen untuk membatasi otoritas penggunaan aplikasi. Rancangan *flowchart* sederhana dapat dilihat pada gambar 3.3.



Gambar 3.3 Flowchart Dosen

3.5.4 Flowchart Mahasiswa

Mahasiswa memiliki akses yang paling sederhana pada sistem yang dibangun, mahasiswa tidak dapat mengubah sistem apapun kecuali melakukan presensi pada matakuliah yang telah diambil. Mahasiswa hanya dapat melihat menu yang telah dibangun tanpa dapat memodifikasinya seperti gambar 3.4.



Gambar 3.4 Flowchart Mahasiswa