

# Rancang Bangun Sistem Absensi *Online* Berbasis *Face Recognition* Menggunakan Platform Android

Muliyadi<sup>a,1</sup>; Siska Anraeni<sup>a,2</sup>; Herman<sup>a,3</sup>

<sup>a</sup> Prodi Teknik Informatika Fakultas Ilmu Komputer Universitas Muslim Indonesia, JL.Urip Sumoharjo No.Km 5, Makassar 90231, Indonesia

<sup>1</sup> muliyadi.ultras21@gmail.com; <sup>2</sup> siska.anraeni@umi.ac.id; <sup>3</sup> herman@umi.ac.id  
\*corresponding author

INFORMASI ARTIKEL	ABSTRAK
Diterima : 07 – 02 – 2022 Direvisi : 21 – 02 – 2022 Diterbitkan : 28 – 02 – 2022	Pendidikan sangatlah penting bagi kemajuan bangsa. Dalam proses pembelajaran secara <i>Online</i> pada saat musim pandemi ini, dosen pasti akan sulit untuk mendata setiap Mahasiswa yang hadir. Tujuan dalam penelitian ini adalah untuk membuat sebuah aplikasi, yang mana nantinya dosen tidak repot lagi melakukan absensi secara manual, dan Mahasiswa dapat dengan mudah absensi dengan cara <i>scan</i> wajah atau <i>Face Recognition</i> , agar absensi dapat dilakukan secara jujur. <i>Eigenface</i> adalah salah satu algoritma pengenalan pola wajah yang berdasarkan pada <i>Principle Component Analysis</i> (PCA). Dalam metode <i>Eigenface</i> , <i>decoding</i> dilakukan dengan menghitung <i>Eigenvector</i> kemudian direpresentasikan dalam sebuah matriks yang berukuran besar. Setiap wajah dipresentasikan dalam kombinasi linear <i>Eigenface</i> . Berdasarkan hasil uji coba sistem yang telah dibuat maka dapat disimpulkan bahwa Aplikasi Absensi mampu melakukan proses absensi secara <i>realtime</i> dengan mendeteksi wajah melalui metode pengenalan wajah yaitu <i>Eigenface</i> PCA ( <i>Principle Component Analysis</i> ) dengan tingkat total akurasi keseluruhan dari Segi Pencahayaan adalah 75% dan Total Akurasi keseluruhan dari segi posisi Wajah adalah 33,34%.
<b>Kata Kunci:</b> Absensi <i>Online</i> Pengenalan Citra wajah <i>Eigenface</i>	
	This is an open access article under the <a href="#">CC-BY-SA</a> license
	

## I. Pendahuluan

Saat ini dunia sedang dilanda penyakit *Corona Virus (COVID)*, sehingga tidak diperbolehkan untuk bertemu atau melakukan kontak fisik, karena ditakutkan adanya penularan penyakit. Oleh karena itu berbagai macam sektor mengalami masalah, terutama di bidang pendidikan. Pendidikan sangatlah penting bagi kemajuan bangsa, oleh karena itu pemerintah tetap menghimbaukan kepada seluruh sekolah di Indonesia agar tetap menjalankan proses mengajar/belajar secara daring/*E-Learning* melalui *zoom* atau aplikasi lainnya.

Dalam proses pembelajaran, Dosen pasti akan sulit untuk mendata setiap Mahasiswa yang hadir dikarenakan harus mengecek satu persatu nama yang hadir dalam pembelajaran tersebut. Melihat kondisi demikian, Saya berfikir untuk membuat sebuah aplikasi, yang mana nantinya, Dosen tidak repot lagi melakukan absensi secara manual, dan Mahasiswa dapat dengan mudah absensi dengan cara *scan* wajah atau *Face Recognition*, agar absensi dapat dilakukan secara jujur.

*Face recognition* adalah sebuah metode yang diterapkan pada teknologi seperti smartphone, komputer, dan lain sebagainya sehingga teknologi tersebut dapat mengenali wajah[1]. Salah satu hal yang menarik untuk dirancang dan direalisasikan adalah pengaplikasian pengenalan wajah untuk sistem absensi *online*. Hal itu karena sistem absensi secara manual terkadang kurang simpel. Pada sistem absensi *online* yang dimaksud adalah mendata tiap mahasiswa yang hadir dalam Matakuliah secara otomatis berdasarkan pengenalan wajah. Sebelum suatu sistem dapat melakukan pengenalan wajah, maka sistem tersebut harus terlebih dahulu melakukan pendeteksian wajah untuk menemukan wajah yang direkam oleh kamera. Hal ini dikarenakan terkadang gambar wajah yang terambil saat direkam tercampur oleh objek lain yang tidak ingin dideteksi [2].

Salah satu metode pengenalan wajah[3] yang digunakan adalah metode *Eigenface*. *Eigenface* adalah suatu metode pengenalan wajah[4] yang berdasarkan pada algoritma *Principal component analysis* (PCA). Secara singkat prosesnya adalah citra direpresentasikan dalam sebuah gabungan vektor yang dijadikan satu matriks tunggal. Dari matriks tunggal ini akan diekstraksi suatu ciri utama yang akan membedakan antara citra wajah satu dengan citra wajah lainnya [5].

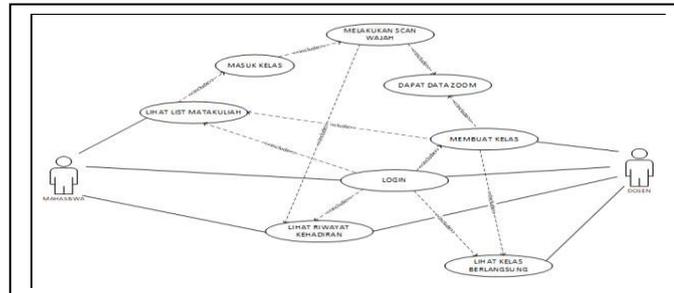
*Principal Componen Analysis* (PCA) adalah sebuah cara untuk mengidentifikasi pola pada data dan kemudian mengekspresikan data tersebut ke bentuk yang lain untuk menunjukkan perbedaan dan persamaan

antar pola. Tujuan dari PCA adalah untuk mereduksi dimensi yang besar dari ruang data (*observed variables*) menjadi dimensi yang lebih kecil dari ruang fitur (*independent variables*), yang dibutuhkan untuk mendeskripsikan data lebih sederhana [6].

## II. Metode

### A. Rancangan Sistem

Berikut ini adalah gambaran tentang tahapan Sistem Aplikasi absensi *online* berbasis *Face Recognition* pada Dosen dan Mahasiswa dengan menggunakan metode *Eigenface PCA*.



Gambar 1. Use case diagram

### B. Cara Kerja Sistem

Pertama, Dosen membuat kelas dan memasukkan data kelas yang akan diajarkan, beserta *link zoom* yang nanti akan diperoleh Mahasiswa setelah berhasil melakukan *scan wajah*[7] pada absensi kelas yang dibuat oleh Dosen, Dosen juga dapat melihat data Mahasiswa yang telah hadir dalam proses pembelajaran[8], dan juga dapat melihat riwayat absensi yang telah dilakukan Mahasiswa, Mahasiswa juga dapat melihat data absensi tiap pertemuan pembelajaran[9][10].

### C. Persamaan Matematika

Dalam sistem ini penyimpanan data dapat disimpan dalam database dan juga dataset. Database pada program ini menggunakan *SQLitstudio*. Adapun Rancangan Database nya dapat dilihat pada tabel dibawah ini.

Tabel 1. Rancangan Database

Name	Data Type
Id	Integer
Matkul	Varchar
Kelas	Varchar
Tanggal	Date
waktu_mulai	Time
waktu_selesai	Time
Zoom	Varchar
Id_dosen	Integer
Nip	Varchar
Password	Varchar
Alamat	Varchar
No_hp	Varchar
Id_mahasiswa	Integer
Hadir	Varchar
Id_jadwal	Integer
Semester	Varchar
Nama_kelas	Varchar
Nim	Varchar
Nama_mahasiswa	Varchar

Sedangkan untuk menyimpan citra wajah tiap mahasiswa akan tersimpan pada *storage* Hp berupa ekstensi *Xml*.

#### D. Jenis Penelitian

Untuk menyempurnakan data – data yang dibutuhkan dalam penyusunan Penelitian ini, maka penulis akan melakukan pengumpulan data dengan menggunakan 2 cara yaitu:

- 1) *Penelitian Kepustakaan (Library Research)*  
Penelitian Kepustakaan (*Library Research*), yaitu pengumpulan data dengan cara membaca buku melalui literature, tutorial – tutorial maupun artikel dari internet yang bersifat ilmiah yang ada hipotesisnya dengan materi pembahasan.
- 2) *Penelitian Lapangan (Field Research)*  
Penelitian Lapangan (*Field Research*), yaitu dilakukan dengan cara mengumpulkan data secara langsung kepada objek penelitian yaitu pada Fakultas Ilmu Komputer Universitas Muslim Indonesia.

#### E. Waktu dan Tempat Penelitian

Penelitian ini akan dilaksanakan selama 4 bulan, mulai Maret sampai Agustus 2021 di Fakultas Ilmu Komputer Universitas Muslim Indonesia.

#### F. Target/Subjek Penelitian

Target penelitian ini adalah Mahasiswa Fakultas Ilmu Komputer Universitas Muslim Indonesia.

#### G. Algoritma Eigenface

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

Berikut adalah tahapan perhitungan *Eigenface*.

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

$$S = (\Gamma_1, \Gamma_2, \dots, \Gamma_M) \quad (1)$$

- 2) Langkah kedua adalah ambil nilai tengah atau *mean* ( $\Psi$ )

$$\psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \quad (2)$$

- 3) Langkah ketiga, kemudian cari selisih ( $\Phi$ ) antara nilai *training image* ( $\Gamma_i$ ) dengan nilai tengah

$$\Phi_i = \Gamma_i - \Psi \quad (3)$$

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

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

$$L = A^T A \quad L = \Phi_m^T \Phi_n \quad (4)$$

- 5) Langkah kelima menghitung *Eigenvalue* ( $\lambda$ ) dan *Eigenvector* ( $v$ ) dari matriks *koyarian* (C)

$$C \times v_i = \lambda_i \times v_i \quad (5)$$

- 6) Langkah keenam, setelah *Eigenvector* ( $v$ ) diperoleh, maka *Eigenface* ( $\mu$ ) dapat dicari dengan persamaan

$$\mu_i = \sum_{k=1}^M v_{ik} \Phi_k$$

$$l = 1, \dots, M \quad (6)$$

Berikut adalah tahapan pengenalan wajah:

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

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

$$\Omega = [\mu_1, \mu_2, \dots, \mu_M] \quad (7)$$

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

$$\epsilon_k = \| \Omega - \Omega_k \| \quad (8)$$

### III. Hasil dan Pembahasan

#### A. Implementasi Sistem

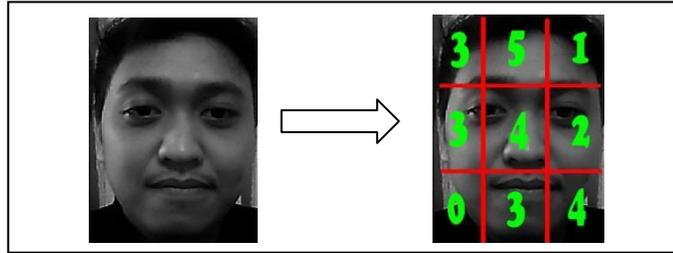
Aplikasi ini dibuat dengan menggunakan Bahasa pemrograman Java dan dikombinasikan dengan menggunakan *Library OpenCV Versi 4.0*. Sistem yang dibuat ini merupakan *system* yang berbasis *Android* yang dibuat pada aplikasi *Android Studio*.

#### B. Tahapan-Tahapan Algoritma Eigenface

##### 1) Perhitungan Eigenface Untuk Gambar Trainig

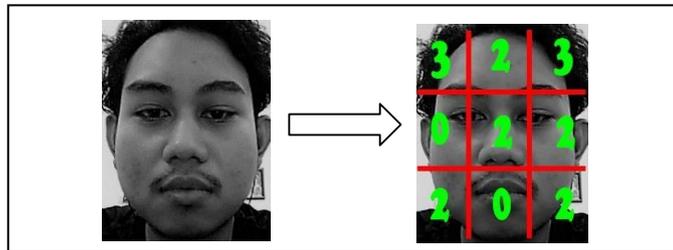
##### a) Penyusunan *flatvektor* matriks citra

Dari data *training image* ( $\Gamma$ ) yang tersedia Langkah selanjutnya adalah menyusun seluruh data *training* menjadi satu matriks tunggal. Representasikan semua matriks *training* menjadi matriks linier  $N \times 1$  atau *flatvektor*.



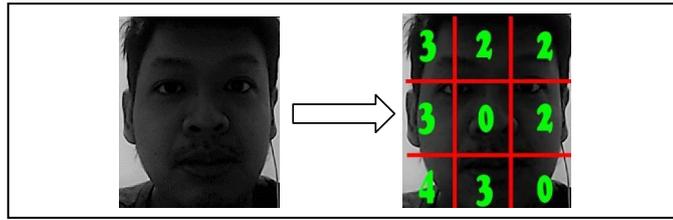
Gambar 2. *Training 1*

$$\Gamma 1 = \begin{bmatrix} 3 & 5 & 1 \\ 3 & 4 & 2 \\ 0 & 3 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 3 \\ 5 \\ 1 \\ 3 \\ 4 \\ 2 \\ 0 \\ 3 \\ 4 \end{bmatrix}$$



Gambar 3. *Training 2*

$$\Gamma 2 = \begin{bmatrix} 3 & 2 & 3 \\ 0 & 2 & 2 \\ 2 & 0 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 3 \\ 2 \\ 3 \\ 0 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 0 \\ 2 \end{bmatrix}$$



Gambar 4. Training 3

$$\Gamma_3 = \begin{bmatrix} 3 & 2 & 2 \\ 3 & 0 & 2 \\ 4 & 3 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 3 \\ 2 \\ 2 \\ 3 \\ 0 \\ 2 \\ 4 \\ 3 \\ 0 \end{bmatrix}$$

b) Menghitung nilai tengah atau *mean* ( $\Psi$ )

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

$$\Psi = \frac{1}{3} \left( \begin{bmatrix} 3 \\ 5 \\ 1 \\ 3 \\ 4 \\ 2 \\ 0 \\ 3 \\ 4 \end{bmatrix} + \begin{bmatrix} 3 \\ 2 \\ 3 \\ 0 \\ 2 \\ 2 \\ 2 \\ 0 \\ 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 2 \\ 2 \\ 3 \\ 0 \\ 2 \\ 4 \\ 3 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 3 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix}$$

c) Menghitung nilai selisih antara *training image* ( $\Gamma$ ) dengan nilai *mean* ( $\Psi$ )

$$\phi_1 = \begin{bmatrix} 3 \\ 5 \\ 1 \\ 3 \\ 4 \\ 2 \\ 0 \\ 3 \\ 4 \end{bmatrix} - \begin{bmatrix} 3 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ -1 \\ 1 \\ 2 \\ 0 \\ -2 \\ 1 \\ 2 \end{bmatrix}$$

$$\phi_2 = \begin{bmatrix} 3 \\ 2 \\ 3 \\ 0 \\ 2 \\ 2 \\ 2 \\ 0 \\ 2 \end{bmatrix} - \begin{bmatrix} 3 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 1 \\ -2 \\ 0 \\ 0 \\ 0 \\ -2 \\ 0 \end{bmatrix}$$

$$\phi_3 = \begin{bmatrix} 3 \\ 2 \\ 2 \\ 3 \\ 0 \\ 2 \\ 4 \\ 3 \\ 0 \end{bmatrix} - \begin{bmatrix} 3 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \\ -2 \\ 2 \\ 2 \\ 1 \\ -2 \end{bmatrix}$$

d) Menghitung nilai matriks *kovarian*

$$L = \begin{bmatrix} 0 & 2 & -1 & 1 & 2 & 0 & -2 & 1 & 2 \\ 0 & -1 & 1 & -2 & 0 & 0 & 0 & -2 & 0 \\ 0 & -1 & 0 & 1 & 2 & 2 & 2 & 1 & -2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 2 & -1 & -1 \\ -1 & 1 & 0 \\ 1 & -2 & 1 \\ 2 & 0 & -2 \\ 0 & 0 & 2 \\ -2 & 0 & 2 \\ 1 & -2 & 1 \\ 2 & 0 & -2 \end{bmatrix} = \begin{bmatrix} 19 & -7 & -12 \\ -7 & 10 & -3 \\ -12 & -3 & 19 \end{bmatrix}$$

e) Hitung nilai *eigenvalue* ( $\lambda$ ) dan *eigenvector* ( $v$ ) dari matriks *kovarian*

$$0 = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} \cdot \begin{bmatrix} 19 & -7 & -12 \\ -7 & 10 & -3 \\ -12 & -3 & 19 \end{bmatrix} = \begin{vmatrix} \lambda - 19 & 7 & 12 \\ 7 & \lambda - 10 & 3 \\ 12 & 3 & \lambda - 19 \end{vmatrix}$$

- f) Nilai *eigenvector*( $\lambda$ ) diperoleh dengan cara mensubstitusi *eigenvalue*( $v$ ) kedalam persamaan  $(\lambda I - L)v = 0$ . *Eigenvector* masing-masing *eigenvalue* didapat berdasarkan masing – masing kolom *eigenvalue* dan kemudian kembali menjadi satu matriks.

**Untuk  $\lambda_1 = 1.1641$**

$$\begin{bmatrix} \lambda - 19 & 7 & 12 \\ 7 & \lambda - 10 & 3 \\ 12 & 3 & \lambda - 19 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow v_1 = \begin{bmatrix} -0.5870 \\ -0.6354 \\ -0.5018 \end{bmatrix}$$

**Untuk  $\lambda_1 = 15.4227$**

$$\begin{bmatrix} \lambda - 19 & 7 & 12 \\ 7 & \lambda - 10 & 3 \\ 12 & 3 & \lambda - 19 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow v_2 = \begin{bmatrix} 0.3529 \\ -0.7586 \\ 0.5477 \end{bmatrix}$$

**Untuk  $\lambda_1 = 31.4131$**

$$\begin{bmatrix} \lambda - 19 & 7 & 12 \\ 7 & \lambda - 10 & 3 \\ 12 & 3 & \lambda - 19 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow v_3 = \begin{bmatrix} -0.7286 \\ 0.1444 \\ 0.6695 \end{bmatrix}$$

$$\text{Maka } C = \begin{bmatrix} -0.5870 & 0.3529 & -0.7286 \\ -0.6354 & -0.7586 & 0.1444 \\ -0.5018 & 0.5477 & 0.6695 \end{bmatrix}$$

- g) Menghitung nilai *eigenface*

$$\mu_1 = vx\phi_1$$

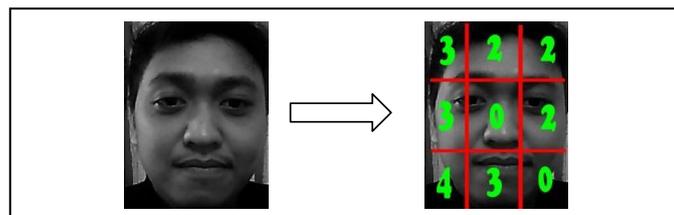
$$\mu_1 = \begin{bmatrix} -0.5870 & 0.3529 & -0.7286 \\ -0.6354 & -0.7586 & 0.1444 \\ -0.5018 & 0.5477 & 0.6695 \end{bmatrix} \times \begin{bmatrix} 0 & 2 & -1 \\ 1 & 2 & 0 \\ -2 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1.8102 & -1.1967 & -0.8703 \\ -1.0474 & -2.6435 & 0.9242 \\ -0.7913 & 0.7614 & 1.8404 \end{bmatrix}$$

$$\mu_2 = \begin{bmatrix} -0.5870 & 0.3529 & -0.7286 \\ -0.6354 & -0.7586 & 0.1444 \\ -0.5018 & 0.5477 & 0.6695 \end{bmatrix} \times \begin{bmatrix} 0 & -1 & 1 \\ -2 & 0 & 0 \\ 0 & -2 & 0 \end{bmatrix} = \begin{bmatrix} -0.7058 & 2.0442 & -0.5870 \\ 1.5172 & 0.3466 & -0.6354 \\ -1.0954 & -0.8372 & -0.5018 \end{bmatrix}$$

$$\mu_3 = \begin{bmatrix} -0.5870 & 0.3529 & -0.7286 \\ -0.6354 & -0.7586 & 0.1444 \\ -0.5018 & 0.5477 & 0.6695 \end{bmatrix} \times \begin{bmatrix} 0 & -1 & 0 \\ 1 & -2 & 2 \\ 2 & 1 & -2 \end{bmatrix} = \begin{bmatrix} -1.1044 & -0.8475 & 2.1631 \\ -0.4698 & 2.2969 & -1.8060 \\ 1.8867 & 0.0758 & -0.2436 \end{bmatrix}$$

## 2) Proses Pengenalan Wajah

- a) Proses pengenalan wajah dilakukan dengan cara mengenali gambar tes, kemudian mencocokkan dengan *training image* yang telah tersimpan. Tahap pertama dimulai dari mengubah matriks persegi menjadi *flatvector* hingga memperoleh nilai *eigenface* ( $\mu$ ).



Gambar 5. Gambar test

$$\Gamma_{new} = \begin{bmatrix} 3 & 2 & 2 \\ 3 & 0 & 2 \\ 4 & 3 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 3 \\ 2 \\ 2 \\ 3 \\ 0 \\ 2 \\ 4 \\ 3 \\ 0 \end{bmatrix}$$

- b) Setelah merepresentasikan gambar *test* ke *flatvector*, kemudian mencari selisih ( $\phi$ ) antara gambar *test* dengan nilai *mean* ( $\Psi$ ).

$$\phi_{new} = \begin{bmatrix} 3 \\ 2 \\ 2 \\ 3 \\ 0 \\ 2 \\ 4 \\ 3 \\ 0 \end{bmatrix} - \begin{bmatrix} 3 \\ 3 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ -2 \\ 1 \\ 2 \\ 0 \\ -2 \\ 1 \\ 3 \end{bmatrix}$$

Dari nilai selisih, maka nilai *eigenface* dapat di hitung.

$$\mu_1 = \begin{bmatrix} -0.5870 & 0.3529 & -0.7286 \\ -0.6354 & -0.7586 & 0.1444 \\ -0.5018 & 0.5477 & 0.6695 \end{bmatrix} \times \begin{bmatrix} 0 & 2 & -2 \\ 1 & 2 & 0 \\ -2 & 1 & 3 \end{bmatrix} = \begin{bmatrix} 1.8102 & -1.1967 & -1.0120 \\ -1.0474 & -2.6435 & 1.7039 \\ -0.7913 & 0.7614 & 3.0120 \end{bmatrix}$$

- c) Setelah nilai *eigenface* untuk gambar *test* diperoleh maka kita bisa melakukan identifikasi dengan menentukan jarak terpendek (ecludian *distance*) dengan *eigenface* dari *eigenvector training image*.

$$\epsilon_1 = \left\| \begin{bmatrix} -0.5870 & 0.3529 & -0.7286 \\ -0.6354 & -0.7586 & 0.1444 \\ -0.5018 & 0.5477 & 0.6695 \end{bmatrix} \begin{bmatrix} -0.5870 & 0.3529 & -0.7286 \\ -0.6354 & -0.7586 & 0.1444 \\ -0.5018 & 0.5477 & 0.6695 \end{bmatrix} \right\|$$

$$\epsilon_1 = \sqrt{(0)^2 + (0)^2 + (-0.1417)^2 + (0)^2 + (0)^2 + (0.7797)^2 + (0)^2 + (0)^2 + (1.1712)^2}$$

$$\epsilon_1 = \sqrt{1.9997} = 1.4141$$

$$\epsilon_2 = \left\| \begin{bmatrix} 1.8102 & -1.1967 & -1.0120 \\ -1.047 & -2.6435 & 1.7039 \\ -0.791 & 0.7614 & 3.0120 \end{bmatrix} \begin{bmatrix} -0.7058 & 2.0442 & -0.5870 \\ 1.5172 & 0.3466 & 0.6354 \\ -1.09 & -0.8372 & 0.5018 \end{bmatrix} \right\|$$

$$\epsilon_1 = \sqrt{(1.2197)^2 + (10.5034)^2 + (0.1806)^2 + (6.5772)^2 + (4.3685)^2 + (5.4723)^2 + (0.0924)^2 + (2.5555)^2 + (12.4171)^2}$$

$$\epsilon_1 = \sqrt{43.386} = 6.5868$$

$$\epsilon_3 = \left\| \begin{bmatrix} 1.8102 & -1.1967 & -1.0120 \\ -1.047 & -2.6435 & 1.7039 \\ -0.791 & 0.7614 & 3.0120 \end{bmatrix} \begin{bmatrix} -1.1044 & -0.8475 & 2.1631 \\ -0.4698 & 2.2969 & -1.8060 \\ -1.8867 & 0.0758 & -0.2436 \end{bmatrix} \right\|$$

$$\epsilon_3 = \sqrt{(2.9146)^2 + (0.3492)^2 + (-3.1751)^2 + (-0.5776)^2 + (-4.9404)^2 + (3.5099)^2 + (-2.678)^2 + (0.6856)^2 + (3.2556)^2}$$

$$\epsilon_3 = \sqrt{73.999} = 8.6022$$

Dari hasil perhitungan nilai ecludian *distance* gambar *training 1*, *training 2* dan *training 3* terhadap gambar *test*, maka nilai jarak *eigenface* yang terkecil diidentifikasi lebih mirip antara *training 1* dengan gambar tes, dibandingkan dengan *training 2* dan *training 3*.

### C. Pengujian Black Box

Pada tahap ini merupakan kelanjutan dari tahap implementasi. Yang dimana pengujian sistem dilakukan dengan metode pengujian black box yang terfokus dari persyaratan fungsional perangkat lunak.

- 1) *Pengujian Aplikasi Dosen*
  - a) Registrasi

Tabel 2. Pengujian form registrasi dosen

Test Factor	Hasil	Data Type
Registrasi dosen	√	Dosen dapat melakukan registrasi

- b) *Login*

Tabel 3. Pengujian form *login* dosen

Test Factor	Hasil	Data Type
<i>Login</i> dosen	√	Dosen dapat <i>login</i> menggunakan NIP dan <i>password</i> yang telah diregistrasi

- c) *Buat kelas*

Tabel 4. Pengujian form buat kelas dosen

<i>Test Factor</i>	<i>Hasil</i>	<i>Data Type</i>
Buat kelas	√	Dosen dapat membuat kelas sesuai yang diinginkan

d) Lihat Riwayat absensi

Tabel 5. Pengujian form lihat riwayat absensi dosen

<i>Test Factor</i>	<i>Hasil</i>	<i>Data Type</i>
Lihat riwayat absen	√	Dosen dapat melihat data mahasiswa yang telah melakukan absensi

e) Lihat kelas berlangsung

Tabel 6. Pengujian form kelas berlangsung dosen

<i>Test Factor</i>	<i>Hasil</i>	<i>Data Type</i>
Lihat kelas berlangsung	√	Dosen dapat melihat absensi kelas yang sementara berlangsung

f) Edit profil

Tabel 7. Pengujian form edit profil

<i>Test Factor</i>	<i>Hasil</i>	<i>Data Type</i>
Dosen edit profil	√	Dosen dapat mengubah data <i>profil</i>

2) *Pengujian Aplikasi Mahasiswa*

a) Registrasi

Tabel 8. Pengujian form registrasi mahasiswa

<i>Test Factor</i>	<i>Hasil</i>	<i>Data Type</i>
Registrasi mahasiswa	√	Mahasiswa dapat melakukan registrasi

b) Registrasi data wajah

Tabel 9. Pengujian form registrasi data wajah mahasiswa

<i>Test Factor</i>	<i>Hasil</i>	<i>Data Type</i>
Registrasi data wajah	√	Mahasiswa dapat melakukan pengambilan <i>scan</i> wajah,

c) *Login*

Tabel 10. Pengujian form *login* mahasiswa

<i>Test Factor</i>	<i>Hasil</i>	<i>Data Type</i>
<i>Login</i> mahasiswa	√	mahasiswa dapat melakukan pengambilan <i>scan</i> wajah mahasiswa dapat <i>login</i> menggunakan NIM dan <i>password</i> yang telah di registrasi

d) Absen kelas

Tabel 11. Pengujian form absen kelas mahasiswa

<i>Test Factor</i>	<i>Hasil</i>	<i>Data Type</i>
Absen kelas mahasiswa	√	Mahasiswa dapat melakukan absensi saat pelajaran di mulai

e) *Scan* wajah

Tabel 12. Pengujian form *scan* wajah mahasiswa

<i>Test Factor</i>	<i>Hasil</i>	<i>Data Type</i>
<i>Scan</i> wajah	√	Mahasiswa dapat melakukan <i>scan</i> wajah pada saat absensi kelas

f) Lihat Riwayat absensi

Tabel 13. Pengujian form lihat Riwayat absen mahasiswa

Test Factor	Hasil	Data Type
Lihat riwayat absen	√	Mahasiswa dapat melihat riwayat absensi kehadiran

g) Edit profil

Tabel 14. Pengujian form edit profil

Test Factor	Hasil	Data Type
Mahasiswa edit profil	√	Mahasiswa dapat mengubah data profil

3) Pengujian Laporan Hasil Absensi



Gambar 6. Hasil Absensi

4) Pengujian Citra Berdasarkan Cahaya

Tabel 15. Pengujian form edit profil

Citra Wajah Yang Diuji	Pencahayaannya	Citra Wajah Pencitraan
	Terang	Terdeteksi
	Normal	Terdeteksi
	Redup	Tidak terdeteksi
	Terang	Terdeteksi
	Normal	Terdeteksi
	Redup	Terdeteksi
	Terang	Terdeteksi
	Normal	Terdeteksi
	Redup	Terdeteksi

Berdasarkan hasil pengujian yang telah dilakukan, maka diperoleh tingkat keberhasilan akurasi citra wajah berdasarkan segi pencahayaan Terang, Normal, dan Redup adalah sebagai berikut:

$$\text{Depan} : \frac{8}{8} \times 100\% = 100\% \quad \text{Kanan} : \frac{6}{8} \times 100\% = 75\% \quad \text{Kiri} : \frac{4}{8} \times 100\% = 50\%$$

$$\text{Total akurasi dari keseluruhan Segi Posisi Wajah adalah } \frac{18}{24} \times 100\% = 75\%$$

5) Pengujian Citra Berdasarkan Posisi Wajah

Tabel 15. Pengujian form edit profil

Citra Wajah Yang Diuji	Pencahayaannya	Citra Wajah Pencitraan
	Depan	Terdeteksi
	Kanan	Tidak terdeteksi

	Kiri	Tidak terdeteksi
	Depan	Terdeteksi
	Kanan	Tidak terdeteksi
	Kiri	Tidak terdeteksi
	Depan	Terdeteksi
	Kanan	Tidak terdeteksi
	Kiri	Tidak terdeteksi

Berdasarkan hasil pengujian yang telah dilakukan, maka diperoleh tingkat keberhasilan akurasi citra wajah berdasarkan Posisi Wajah Depan, Kanan, dan Kiri adalah sebagai berikut:

$$\text{Depan} : \frac{8}{8} \times 100\% = 100\% \quad \text{Kanan} : \frac{0}{8} \times 100\% = 0\% \quad \text{Kiri} : \frac{0}{8} \times 100\% = 0\%$$

$$\text{Total akurasi dari keseluruhan Segi Posisi Wajah adalah } \frac{8}{7} \times 100\% = 33,34\%$$

#### IV. Kesimpulan dan saran

Berdasarkan hasil uji coba *system* yang telah dibuat maka dapat disimpulkan bahwa Aplikasi Absensi mampu melakukan proses absensi secara realtime dengan mendeteksi wajah melalui metode pengenalan wajah yaitu *Eigenface PCA (Principle Component Analysis)* dengan tingkat total akurasi keseluruhan dari segi pencahayaan adalah 75% dan total akurasi keseluruhan dari segi posisi wajah 33,34%. *System* ini dapat mengenali citra wajah dengan baik dalam posisi lurus. Tingkat keberhasilan akurasi sangat dipengaruhi oleh pencahayaan, semakin terang pencahayaan maka tingkat keberhasilan akurasi juga semakin tinggi. Dan proses pendeteksian citra wajah juga berpengaruh terhadap jarak citra dengan kamera yang digunakan.

#### Daftar Pustaka

- [1] S. Sugeng And A. Mulyana, "Sistem Absensi Menggunakan Pengenalan Wajah (Face Recognition) Berbasis Web Lan," *J. Sisfokom (Sistem Inf. Dan Komputer)*, Vol. 11, No. 1, Pp. 127–135, 2022, Doi: 10.32736/Sisfokom.V11i1.1371.
- [2] C. Lesmana, R. Lim, And L. W. Santoso, "Implementasi Face Recognition Menggunakan Raspberry Pi Untuk Akses Ruang Pribadi," *J. Infra Petra*, Pp. 2–5, 2019.
- [3] D. Aryani, M. N. Ihsan, And P. Septiyani, "Prototype Sistem Absensi Dengan Metode Face Recognition Berbasis Arduino Pada Smk Negeri 5 Kabupaten Tangerang," *Semin. Nas. Teknol. Inf. Dan Multimed. 2017*, Vol. 1, Pp. 37–42, 2017.
- [4] I. K. S. Widiakumara, I. K. G. D. Putra, And K. S. Wibawa, "Aplikasi Identifikasi Wajah Berbasis Android," *Lontar Komput. J. Ilm. Teknol. Inf.*, Vol. 8, No. 3, P. 200, 2017, Doi: 10.24843/Lkjiti.2017.V08.I03.P06.
- [5] M. R. Muliawan, B. Irawan, And Y. Brianorm, "Metode Eigenface Pada Sistem Absensi.03," *J. Coding, Sist. Komput. Untan*, Vol. 03, No. 1, Pp. 41–50, 2015.
- [6] D. E. Pratiwi And A. Harjoko, "Implementasi Pengenalan Wajah Menggunakan Pca (Principal Component Analysis)," *Ijeis*, Vol. 3, No. 2. Pp. 175–184, 2013.
- [7] H. Mulyawan, M. Z. H. Samsono, And Setiawardhana, "Identifikasi Dan Tracking Objek Berbasis Image," Pp. 1–5, 2011, [Online]. Available: [Http://Repo.Pens.Ac.Id/1324/1/Paper\\_Ta\\_Mbah.Pdf](http://Repo.Pens.Ac.Id/1324/1/Paper_Ta_Mbah.Pdf)
- [8] W. A. Siswanto, "Aplikasi Absensi Siswa Menggunakan Fingerprint Dan Penjadwalan Mata Pelajaran Di Sekolah Dasar Negeri Sumberongko Jombang (Application Of Student Absentence Using Fingerprint And Scheduling Schedule In Sumberongko Jombang State School)," 2019.
- [9] A. Husain, A. H. A. Prastian, And A. Ramadhan, "Perancangan Sistem Absensi Online Menggunakan Android Guna Mempercepat Proses Kehadiran Karyawan Pada Pt. Sintech Berkah Abadi," *Technomedia J.*, Vol. 2, No. 1, Pp. 105–116, 2017, Doi: 10.33050/Tmj.V2i1.319.
- [10] R. M. Z. Shery Christian, Enny Dwi Oktaviyani, Deddy Ronaldo, "Aplikasi Absensi Siswa Berbasis Android.Pdf." 2019.