

## **IMPLEMENTASI FACE-RECOGNITION BERBASIS SMARTPHONE ANDROID SEBAGAI MEDIA ABSENSI MAHASISWA STMIK PROFESIONAL MAKASSAR**

**Muhammad Faisal**

Program Studi Sistem Informasi, STMIK Profesional Makassar

muh.faisal.art@gmail.com

### ***Abstrak***

*Kehadiran Mahasiswa (school attendance) adalah keikutsertaan mahasiswa secara fisik dan mental terhadap aktivitas kampus pada jam-jam perkuliahan. Sedangkan ketidakhadiran adalah ketiadaan partisipasi secara fisik mahasiswa terhadap kegiatan perkuliahan dalam kampus. Solusi yang tepat untuk mendukung administrasi pengolahan data kehadiran mahasiswa yang lebih akurat yaitu memanfaatkan kemampuan face recognition pada perangkat smartphone android melalui aplikasi yang dirancang agar dapat menjadi perantara mahasiswa dan dosen sehingga proses pencatatan dapat langsung terlaksana pada komputer yang dimiliki oleh dosen. Hasil penelitian menunjukkan bahwa proses administrasi kehadiran mahasiswa yang akurat dapat terlaksana dengan baik melalui face recognition menggunakan perangkat smartphone Android. Selain itu keutamaan menggunakan face recognition pada proses absensi mahasiswa dapat pula mengatasi kejadian 'titip tandatangan' oleh mahasiswa yang kurang disiplin mengikuti proses perkuliahan.*

***Kata Kunci :*** kehadiran, face recognition, absensi

### **A. PENDAHULUAN**

Belajar merupakan salah satu cara mengasah kemampuan akal pikiran, Dengan belajar yang tekun maka dapat membuat seseorang menjadi lebih berwawasan luas serta dapat dengan mudah menyerap segala hal yang bermanfaat. Pada jenjang perkuliahan, mahasiswa tidak akandiintervensi untuk mengikuti proses belajar dalam kelas karena mahasiswa sudah dianggap dewasa dalam berpikir dan bertindak. Mahasiswa harus sadar jika tidak hadir maka yang rugi adalah dirinya sendiri. Hal yang perlu ditanamkan kepada mahasiswa sekarang ini ialah menumbuhkan semangat kebutuhan akan ilmu sehingga prioritasnya bukan nilai yang dicari, melainkan ilmu. Nilai hanyalah sebuah konsekuensi, Jika belajar dengan tekun maka nilainya akan tinggi, jika kurang perhatian kepada ilmu tersebut maka akan rendah pula nilainya.

Berkaitan dengan kedisiplinan mahasiswa dalam hal kehadiran, adapun permasalahan yang sering ditemukan dikelas yaitu kehadiran semu. Mahasiswa seperti hadir namun sebenarnya tidak hadir. Praktek kehadiran semu ini antara lain ialah titip tandatangan, yaitu mahasiswa yang tidak berangkat kuliah dan hanya menitip kepada teman untuk

membubuhkan tandatangan palsu dalam absensinya. Cara yang lain adalah mengisi absensi sebelum dosen hadir, yaitu mahasiswa datang ke ruang kelas lebih awal, selanjutnya mengisi absensi lalu pergi setelah membubuhkan tandatangan, atau dengan cara menghadiri kelas sampai absensi beredar, setelah membubuhkan tanda tangan mahasiswapun izin keluar ruangan dan tidak kembali.

Berdasarkan permasalahan yang telah ditemukan maka penulis berinisiatif untuk membangun suatu perangkat lunak absensi mahasiswa menggunakan face recognition pada perangkat smartphone sebagai media absensi. Keragaman bentuk wajah antar mahasiswa dapat menjamin keabsahan data kehadiran mahasiswa yang bersangkutan, disisi lain data digital mengenai absensi mahasiswa dapat dimanfaatkan untuk kebutuhan administrasi sehingga setiap dosen memiliki data histori mahasiswa secara digital.

## **B. METODE PENELITIAN**

### **Metode Haar Cascade Classifier**

Face recognition merupakan sebuah proses mengenali wajah dari seseorang yang sebelumnya sudah tersimpan dalam database/sudah dipelajari sebelumnya. Salah satu metode yang cukup terkenal ialah metode Algoritma Haar Cascade Classifier. Haar Feature adalah fitur yang didasarkan pada Wavelet Haar. Wavelet Haar adalah gelombang tunggal bujur sangkar (satu interval tinggi dan satu interval rendah). Untuk dua dimensi, satu terang dan satu gelap. Selanjutnya kombinasi kotak yang digunakan untuk pendeteksian objek visual yang lebih baik. Setiap Haar-like feature terdiri dari gabungan kotak - kotak hitam dan putih.

Metode Haar ini mengadopsi Adaboost Algoritma (Viola and Jones, 2004), dimana integral image dapat dihitung dari suatu gambar dengan menggunakan beberapa operasi per pixel. Setelah dihitung, salah satu fitur tersebut Haar-like dapat dihitung pada setiap skala atau lokasi dalam waktu yang konstan (Viola dan Jones, 2004).

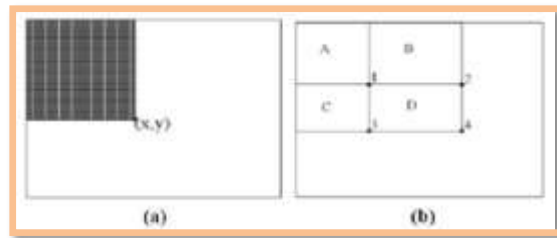
Seleksi fitur ini dicapai dengan menggunakan algoritma pembelajaran AdaBoost dengan membatasi setiap classifier lemah tergantung pada hanya fitur tunggal (Viola dan Jones, 2004).

Adanya fitur Haar ditentukan dengan cara mengurangi rata-rata piksel pada daerah gelap dari rata-rata piksel pada daerah terang. Jika nilai perbedaannya itu diatas nilai ambang atau treshold, maka dapat dikatakan bahwa fitur tersebut ada. Nilai dari Haar-like feature adalah perbedaan antara jumlah nilai-nilai piksel gray level dalam daerah kotak

Faisal – Implementasi Face Recognition Berbasis Smartphone Android hitam dan daerah kotak putih (Viola dan Jones, 2004). dimana untuk kotak pada Haar-like feature dapat dihitung secara cepat menggunakan “integral image” .

$$f(x) = SBRectangle - SWRectangle$$

Integral Image digunakan untuk menentukan ada atau tidaknya dari ratusan fitur Haar pada sebuah gambar dan pada skala yang berbeda secara efisien (Viola dan Jones, 2004). Pada umumnya, pengintegrasian tersebut berarti menambahkan unit-unit kecil secara bersamaan. Dalam hal ini unit-unit kecil tersebut adalah nilai-nilai piksel. Nilai integral untuk masing-masing piksel adalah jumlah dari semua piksel-piksel dari atas sampai bawah. Dimulai dari kiri atas sampai kanan bawah, keseluruhan gambar itu dapat dijumlahkan dengan beberapa operasi bilangan bulat per piksel.



**Gambar 1.** Ilustrasi Pixel

Gambar 1. Value Integral Image dalam point (x,y), b. Komputasi jumlah pixel gambar (Viola dan Jones, 2004). Seperti yang ditunjukkan oleh gambar 1 di atas setelah pengintegrasian, nilai pada lokasi piksel (x,y) berisi jumlah dari semua piksel di dalam daerah segiempat dari kiri atas sampai pada lokasi (x,y) atau daerah yang diarsir. Guna mendapatkan nilai rata-rata piksel pada area segiempat (daerah yang diarsir) ini dapat dilakukan hanya dengan membagi nilai pada (x,y) oleh area segiempat (Viola dan Jones, 2004)

$$s(x, y) = s(x, y - 1) + i(x, y)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y)$$

$$ii(x, y) = \sum_{x', sx, sy} i(x', y')$$

Dimana  $ii(x, y)$  adalah integral image dan  $i(x, y)$  adalah original image.

Guna mengetahui nilai piksel untuk beberapa segiempat yang lain, seperti segiempat D, dapat dilakukan dengan cara menggabungkan jumlah piksel pada area segiempat  $A+B+C+D$ , dikurangi jumlah dalam segiempat  $A+B$  dan  $A+C$ , ditambah jumlah piksel di dalam A. Dengan,  $A+B+C+D$  adalah nilai dari integral image pada lokasi 4,  $A+B$  adalah nilai pada lokasi 2,  $A+C$  adalah nilai pada lokasi 3, dan A pada lokasi 1. Sehingga hasil dari D dapat dikomputasikan (Viola dan Jones, 2004)

$$D = (A+B+C+D) - (A+B) - (A+C) + A$$

Cascade classifier adalah sebuah rantai stage classifier, dimana setiap stage classifier digunakan untuk mendeteksi apakah didalam image sub window terdapat obyek yang diinginkan (object of interest). stage classifier dibangun dengan menggunakan algoritma adaptive-boost (AdaBoost). Algoritma tersebut mengkombinasikan performance banyak weak classifier untuk menghasilkan strong classifier. Weak classifier dalam hal ini adalah nilai dari haar-like feature. Jenis Ada Boost yang digunakan adalah Gentle AdaBoost (Dodit S, et al., 2013). Adapun Algoritma Cascade Clasifier Haar Like Features yaitu :

1. Input nilai  $f$  dan  $d$ , dimana nilai  $f$  merupakan nilai positif maksimum yang dapat diterima dalam sebuah *layer*, dan  $d$  merupakan nilai minimum yang dapat diterima dalam sebuah *layer*.
2. Select target  $F_{target}$ ;
3.  $P$  = set dari sampel positif
4.  $N$  = set dari sampel negatif

$$F_0 = 1.0; D_0 = 1.0$$

$$i = 0$$

$$\text{while } F_i > F_{target}$$

$$i \leftarrow i + 1$$

$$n_i = 0; F_i = F_{i-1}$$

$$\text{while } F_i > f \times F_{i-1}$$

$$n_i \leftarrow n_i + 1$$

5. Use  $P$  and  $N$  to train a classifier with  $n_i$  features using AdaBoost
6. Evaluasi Clasifier untuk memvalidasi nilai  $F_i$  and  $D_i$ .
7. Penurunan nilai threshold untuk mendapatkan ambang batas minimum

Eigenface merupakan salah satu algoritma pengenalan pola wajah yang berdasarkan pada Principle Component Analysis (PCA), Algoritma Eigenface banyak digunakan pada pengenalan tulisan tangan, pembacaan bibir, pengenalan suara dan pencitraan medis (Turk, et al., 2003). Menurut Layman, Eigenface adalah sekumpulan standard face ingredient yang diambil dari analisis statistic dari banyak gambar wajah (Dodit S, et al., 2013).

Untuk menghasilkan eigenface, sekumpulan citra digital dari wajah manusia diambil pada kondisi pencahayaan yang sama kemudian dinormalisasikan dan diproses pada resolusi yang sama (misal  $m \times n$ ), kemudian citra tadi diperlakukan sebagai vector dimensi  $m \times n$  dimana komponennya diambil dari nilai piksel citra (Turk, et al., 2003).

Algoritma Eigenface secara keseluruhan cukup sederhana. Image matriks ( $\Gamma$ ) direpresentasikan ke dalam sebuah himpunan matriks ( $T_1, T_2, T_3, \dots, T_M$ ). Cari nilai rata-rata ( $\psi$ ) dan gunakan untuk mengekstraksi eigenvector ( $v$ ) dan eigenvalue ( $\lambda$ ) dari himpunan matriks. Gunakan nilai eigenvector untuk mendapatkan nilai eigenface dari image. Apabila ada sebuah image baru atau test face ( $T_{new}$ ) yang ingin dikenali, proses yang sama juga diberlakukan untuk image ( $T_{new}$ ), untuk mengekstraksi eigenvector ( $v$ ) dan eigenvalue ( $\lambda$ ), kemudian cari nilai eigenface dari image test face ( $T_{new}$ ). Setelah itu barulah image baru ( $T_{new}$ ) memasuki tahapan pengenalan dengan menggunakan metode euclidan distance (Dodit S, et al., 2013).

Euclidean distance adalah salah satu metode yang dapat digunakan untuk melakukan klasifikasi citra wajah yang baru (tidak diketahui) ke salah satu dari citra wajah yang telah diketahui. Nilai bobot antara dua vektor  $i$  dan  $j$  yang dihitung dengan menggunakan Euclidean distance menunjukkan ukuran kesamaan antara citra  $i$  dan  $j$  (Turk, et al., 2003).

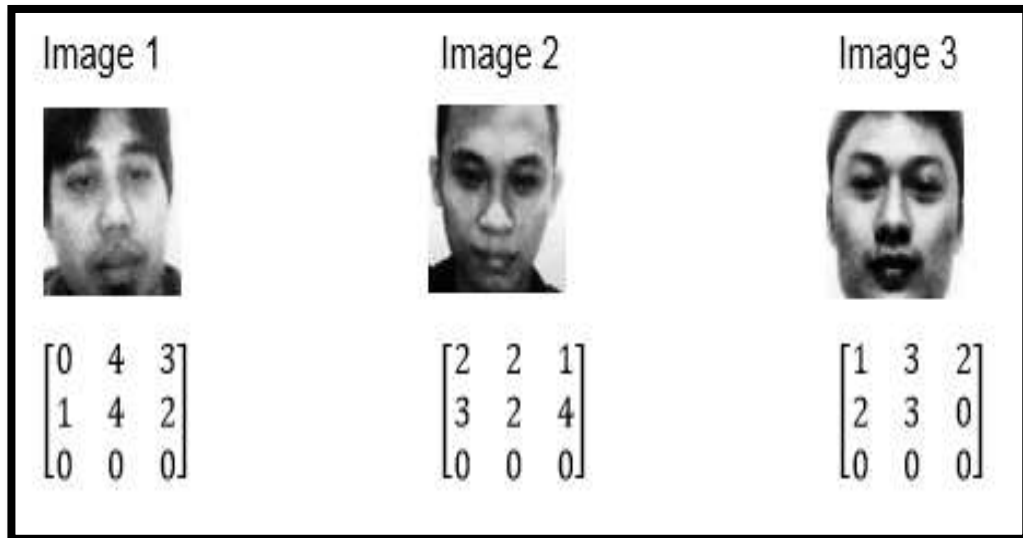
Perhitungan Euclidean distance dilakukan menurut rumus berikut:

Kemudian dicari nilai terkecil dari kumpulan jarak data citra wajah masukan dengan data training yang telah disimpan sebelumnya.

Dalam PCA proses klasifikasi menjamin bahwa vector-vector wajah yang dihasilkan harus jatuh berada di sekitar citra rekonstruksinya. Sebaliknya, untuk vector – vector bukan wajah haruslah jatuh diluar citra – citra rekonstruksi. Sehingga untuk itu, didefinisikan suatu persamaan ukuran jarak, dimana pengurangan jarak antara citra yang sudah dikurangi dengan mean ( mean subtracted image) dan citra rekonstruksinya dengan persamaan berikut :

**Tahapan perhitungan Eigen Face :**

- Langkah pertama adalah menyiapkan data dengan membuat suatu himpunan S yang terdiri dari seluruh training image, Buat himpunan image (S) dai total M training\_image, dimana setiap image adalah p x q piksel. Misal di dalam himpunan image terdapat tiga image ukuran 3 x 3 piksel maka :



Gambar 2. Himpunan Gambar

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

Cari nilai tengah atau mean ( $\Psi$ )

$$\Psi = \frac{1}{3} \sum_{n=1}^3 \Gamma_n \Gamma_n = \frac{1}{3} \left[ \begin{bmatrix} 0 & 4 & 3 \\ 1 & 4 & 2 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 2 & 2 & 1 \\ 3 & 2 & 4 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 3 & 2 \\ 2 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right]$$

$$\Psi = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- Langkah ketiga kemudian cari selisih ( $\Phi$ ) antara training image ( $\Gamma_1$ ) dengan nilai tengah ( $\Psi$ )

$$\Phi = \Gamma_1 - \Psi$$

Cari selisih ( $\Phi$ ) antara training image ( $\Gamma$ ) dengan nilai tengah ( $\Psi$ ), apabila ditemukan nilai dibawah nol ganti nilainya dengan nol.

$$\Phi_1 = \Gamma_1 - \Psi = \begin{bmatrix} 0 & 4 & 3 \\ 1 & 4 & 2 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 3 & 2 \\ 2 & 3 & 2 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Phi_2 = \Gamma_2 - \Psi = \begin{bmatrix} 2 & 2 & 1 \\ 3 & 2 & 4 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 3 & 2 \\ 2 & 3 & 2 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Phi_3 = \Gamma_3 - \Psi = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 3 & 2 \\ 2 & 3 & 2 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

4. Langkah keempat adalah menghitung nilai matriks kovarian (C)

$$Cx \ v_i = \lambda_i \ x \ v_i$$

Cari nilai *eigenvalue* ( $\lambda$ ) dan *eigenvector* ( $v$ ).

$$L \ x \ v = \lambda \ x \ v$$

$$L \ x \ v = \lambda I \ x \ v$$

$$(L - \lambda I) = 0 \ \text{atau} \ (\lambda I - L) = 0$$

Maka *eigenvalue* ( $\lambda$ ) dapat dihitung,  $\det(\lambda I - L) = 0$

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

Akan dihasilkan nilai  $\lambda = 1$ , dan  $\lambda = 4$

$$V = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

*eigenvector*( $v$ ) dihasilkan dengan mensubstitusi nilai *eigenvalue* ( $\lambda$ ) kedalam persamaan

$$(\lambda I - L) \ v = 0$$

untuk  $\lambda = 4$ , maka

$$\begin{bmatrix} 4 - 4 & 2 & 1 \\ 2 & 4 - 3 & 1 \\ 1 & 1 & 4 - 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 2 & 1 \\ 2 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 2v_1 & 2v_2 & v_3 \\ 2v_1 & v_2 & v_3 \\ v_1 & v_2 & 3v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Dihasilkan *eigenvector*  $\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$

Sehingga *eigenvector* yang dihasilkan dari matriks  $L$  adalah :

$$\begin{bmatrix} 1 & -3 & 1 \\ -1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Langkah keenam, setelah *eigenvector* ( $v$ ) diperoleh, maka *eigenface* ( $\mu$ ) dapat dicari dengan :

$$i = 1, \dots, \mathcal{M}$$

Cari nilai *eigenface* ( $\mu$ )

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

$$\mu_1 = v \cdot \Phi_1 = \begin{bmatrix} 1 & -3 & 1 \\ -1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -2 & 1 \\ 0 & -1 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\mu_2 = v \cdot \Phi_2 = \begin{bmatrix} 1 & -3 & 1 \\ -1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -2 & 0 & -3 \\ -1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\mu_3 = v \cdot \Phi_3 = \begin{bmatrix} 1 & -3 & 1 \\ -1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Untuk proses matching image penjelasannya sebagai berikut :

Sebuah image wajah baru atau test face ( $\Gamma_{new}$ ) akan dicoba untuk dikenali, pertama terapkan cara pada tahapan pertama perhitungan *eigenface* untuk mendapatkan nilai *eigenface* dari image tersebut :

Cari nilai *eigenface image* baru ( $\Gamma_{new}$ ) yang akan dikenali.

- a. Cari selisih ( $\Phi$ ) antara *test face* ( $\Gamma_{new}$ ) dengan nilai tengah ( $\Psi$ ), apabila ditemukan nilai dibawah nol ganti nilainya dengan nol. Misal *test face* ( $\Gamma_{new}$ ) terdiri dari matriks 3 x 3 :

$$\begin{bmatrix} 2 & 6 & 5 \\ 0 & 1 & 3 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\Phi_{new} = \begin{bmatrix} 2 & 6 & 5 \\ 0 & 1 & 3 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 3 & 2 \\ 2 & 3 & 2 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 3 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

b. Cari nilai *eigenface* dari *test face* ( $\Gamma_{new}$ )

$$\mu_{new} = v \cdot \Phi_{new}$$

$$\mu_{new} = \begin{bmatrix} 1 & -3 & 1 \\ -1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 3 & 3 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 0 \\ 2 & -3 & -3 \\ 0 & 0 & 1 \end{bmatrix}$$

Gunakan *Euclidan Distance* untuk mencari selisih terkecil antara *eigenface training image* ( $\Gamma_i$ ) dalam *database* dengan *eigenface test face* ( $\Gamma_{new}$ ).

$$\varepsilon_1 = \|\Omega_1 - \Omega_{new}\| = \left\| \begin{bmatrix} 0 & -2 & 1 \\ 0 & -1 & -1 \\ 0 & 1 & 0 \end{bmatrix} - \begin{bmatrix} 2 & 3 & 0 \\ 2 & -3 & -3 \\ 0 & 0 & 1 \end{bmatrix} \right\| = 16$$

$$\varepsilon_2 = \|\Omega_2 - \Omega_{new}\| = \left\| \begin{bmatrix} 2 & 0 & -3 \\ -1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 2 & 3 & 0 \\ 2 & -3 & -3 \\ 0 & 0 & 1 \end{bmatrix} \right\| = 20$$

$$\varepsilon_3 = \|\Omega_3 - \Omega_{new}\| = \left\| \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 2 & 3 & 0 \\ 2 & -3 & -3 \\ 0 & 0 & 1 \end{bmatrix} \right\| = 12$$

Karena jarak (*distance*) *eigenface image 3* dengan *eigenface tes face* yang paling kecil, maka hasil identifikasi menyimpulkan bahwa *test face* lebih mirip dengan *image 3* daripada *image 1* atau *image 2*.

adapun algoritma selengkapnya adalah sebagai berikut :

1. Buat kumpulan *training image*  $T_1, T_2, T_3, \dots, \dots, T_M$  (*Training Face*) dengan ukuran citra yang sama dan posisi tengah (*center*) pada citra.
2. Gabungkan semua citra  $\Gamma_i$  menjadi elemen *vector*.
3. Hitung nilai rata-rata *vector face*  $\psi$  dari citra wajah asli  $T_i$  dan simpan ke dalam *variable*  $\Phi_i$  :

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

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

4. Hitung *covariance matrix*:

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

5. Hitung *eigenvector* dan *eigenvalues* dari *covariance matrix* dan selanjutnya diurutkan berdasarkan nilai *eigen* terbesar.

6. Semakin tinggi nilai *eigen*, akan menentukan semakin banyak ciri atau karakteristik wajah yang akan dikenali.

### C. HASIL

Adapun hasil perancangan dan pengujian terhadap aplikasi yang digunakan adalah sebagai berikut :

#### Form Registrasi Mahasiswa



The screenshot shows a registration form with the following fields and values:

NIM	Ketik NIM Anda
Nama	Ketik Nama
J-Kelamin	<input checked="" type="radio"/> Pria <input type="radio"/> Wanita
Kelas	Ketik Kelas
Waktu	P, 26/04/2017, 15:38:50
Jurusan	Sistem Informasi

Below the form, there is a 'Foto' section with a 'Preview' button and a 'SIMPAN' button. At the bottom, there are icons for 'Send To Desktop' and 'Cancel'.

Gambar 4. Form Registrasi Mahasiswa

#### Registrasi Wajah



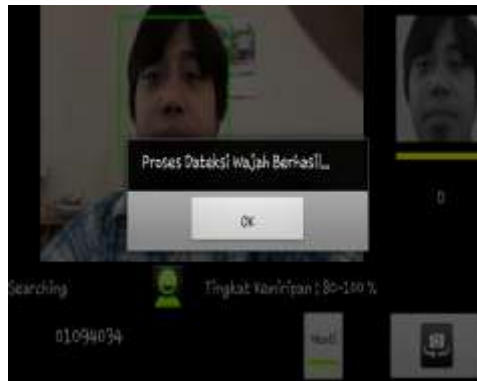
Gambar 5. Registrasi Wajah Mahasiswa

#### Proses Training Wajah



Gambar 6. Proses Training Wajah

**Proses Absensi Berhasil**



**Gambar 7.** Data Absensi Terkirim

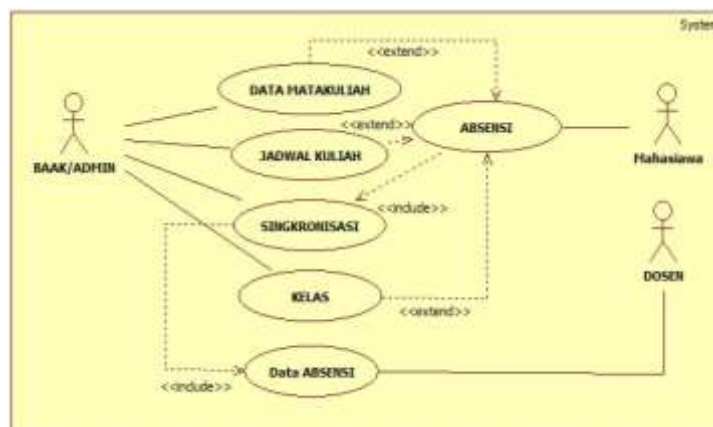
**Rekapitulasi Kehadiran Mahasiswa**

STMIK PROFESIONAL MAKASSAR		
DATA MAHASISWA TAP KELAS PER MATA KULIAH		
Kode Kelas :	DFR	
- Kode Mata Kuliah :	MK-002	
Mata Kuliah :	VISUAL II	
<b>No.</b>	<b>Nim</b>	<b>Nama</b>
1	04.124.209	Sonny
2	12.091.124	Yudistira
		Jumlah Mahasiswa : 2
Kode Kelas :	DFR	
- Kode Mata Kuliah :	MK-003	
Mata Kuliah :	Kecerdasan Buatan	
<b>No.</b>	<b>Nim</b>	<b>Nama</b>
1	ichal	Muhammad Faisal
		Jumlah Mahasiswa : 1

**Gambar 8.** Rekapitulasi Kehadiran

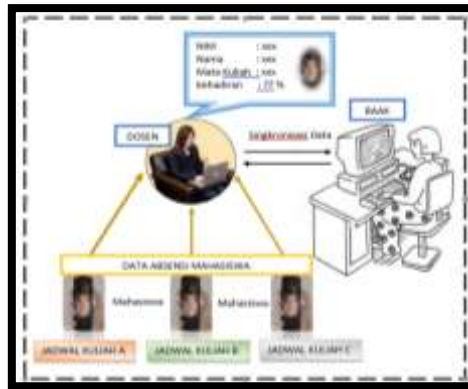
**D. PEMBAHASAN**

**Use Case**



**Gambar 9.** UML Sistem Berjalan.

### Proses pengiriman data absensi



**Gambar 10.** Proses Absensi

Adapun Pembahasan terhadap sistem yang berjalan adalah:

1. Dalam sistem yang dibangun terdapat 3 aktor yaitu BAAK, Mahasiswa, Dosen.
2. Pada tahap awal BAAK yang juga bertindak sebagai Administrator melakukan input data Matakuliah, Jadwal Kuliah, Kelas.
3. Tahapan selanjutnya yaitu Mahasiswa melakukan absensi melalui perangkat smartphone berdasarkan jadwal matakuliah yang sedang berjalan.
4. Dosen melakukan rekam data absensi pada komputer / laptop.
5. Dosen mengirim data absensi ke BAAK dan dilakukan proses sinkronisasi.
6. Dosen melihat data absensi mahasiswa berdasarkan matakuliah

### DAFTAR PUSTAKA

- [1] Russell d. Archibald. 2003. *Life cycle models for high-technology projects – applying systems thinking to managing projects*, 5 - 8.
- [2] Lily Wulandari, I Wayan Simri Wicaksana. 2005. *Semantic-web solusi Interoperabilitas sebagai Penunjang Jaringan Sistem Produksi*. 2 - 8.
- [3] Dodit S. 2013. *Penerapan Face Recognition dengan Metode Eigen Face Dalam Intelligent Home Security*. Surabaya.
- [4] Damayanti, F. 2010. *Pengenalan Citra Wajah Menggunakan Metode Linear Discriminant Analysis*. Surabaya.
- [5] Gunadi, K. 2002. *Face Recognition Menggunakan metode Linear Discriminant Analysis*. Surabaya
- [6] Joko Hartono. 2009. *Pengenalan Pola Wajah dalam Perspektif Jarak Deteksi*. Program Ganda Teknik Informatika dan Matematika, Universitas Bina Nusantara.

Jakarta.

- [7] Samuel , L. 1999. *Paradigma Kontekstual Pengenalan Wajah Manusia*. Unesa University Press. Semarang.
- [8] Turk, E., 2003. *Desain Sistem Pengenalan Wajah Dengan Variasi Ekspresi dan Posis menggunakan metode Eigen Face*. Balai Pustaka. Jakarta.
- [9] Viola and Jones. 2004. *Translator Haar Cascade for Object Detection..* Department of Electrical Engineering and Computer Science-Compute Vision Lab University of Tennessee Knoxville. USA.