

# METODE ALGORITMA EIGENFACE DAN EUCLIDEAN DISTANCE DALAM SISTEM PENGENALAN WAJAH

Norhikmah<sup>1)</sup>, Aditya Dharma Satria S<sup>2)</sup>

<sup>1, 2)</sup> Ilmu Komputer UNIVERSITAS AMIKOM Yogyakarta  
Jl Ring road Utara, Condongcatut, Sleman, Yogyakarta 55281  
Email : [hikmah@amikom.ac.id](mailto:hikmah@amikom.ac.id)<sup>1)</sup>, [aditya.simanjuntak@students.amikom.ac.id](mailto:aditya.simanjuntak@students.amikom.ac.id)<sup>2)</sup>

## Abstrak

Metode Algoritma Eigenface dalam sistem pengenalan atau pendeteksi wajah yang termasuk dalam ranah pengolahan citra. Proses pengenalan citra digital dalam sistem pengenalan wajah ini menggunakan algoritma Eigenface yang terdapat pada OpenCV. Hasil pencocokannya, diperoleh rentang selisih jarak eigenface citra wajah satu (1,73) lebih kecil daripada eigenface citra wajah (2,82) dengan menggunakan rumus euclidean distance

**Kata kunci:** Eigenface, Citra Digital, OpenCV

## 1. Pendahuluan

Sistem olah citra digital merupakan salah satu sistem yang saat ini tengah giat dikembangkan karena dalam alasan keamanan. Yang mana, proses pengenalan citra digital akan dilakukan dengan olah data wajah seseorang. Namun, yang menjadi pokok permasalahan adalah bagaimana hal tersebut dapat terwujud dalam sebuah sistem. Oleh karena itu, dibutuhkan sebuah algoritma yang mana nantinya dapat mengenali wajah dan dapat dipadukan dengan sistem keamanan dengan pengenalan citra digital.

**Tujuan :** 1). Untuk menerapkan Algoritma Eigenface sebagai pengenalan citra digital dalam sistem pengenalan wajah 2). Untuk mengetahui rentang kecocokan antar image wajah dengan euclidean distance

## Tinjauan Pustaka

Aplikasi Absensi dapat dibuat menggunakan pengenalan wajah dengan menggunakan metode eigenface untuk pengenalan wajah. Faktor-faktor yang mempengaruhi hasil pengenalan pada sistem ini adalah pencahayaan, jarak *capture* antara objek. Kesimpulan ini didapat dari hasil perbandingan data pengamatan yang dicoba pada 5 orang pada intensitas cahaya yang berbeda dan jarak pengambilan gambar. Terdapat perbedaan keakuratan[4].

Metode viola jonnes memiliki kelebihan tepat dibandingkan metode deteksi wajah lainnya dengan akurasi 90,9 %. Namun memiliki kelemahan dari sistem deteksi wajah ini yaitu tidak dapat menentukan wajah

pada gambar yang memiliki wajah tidak tegak atau frontal[5].

Rata-rata tingkat keberhasilan pengenalan wajah dengan metode *Adaboost* dan *Eigenfaces PCA* mencapai 80% pada berbagai kondisi berbeda (jarak objek dengan sensor, pencahayaan, posisi, atribut, dan mimik muka) [6].

## Pengenalan Wajah (Face-Recognition)

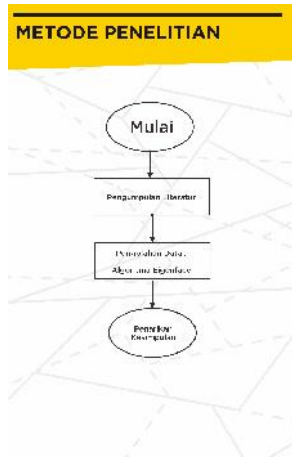
Pengenalan wajah merupakan salah satu bentuk keamanan biometrik yang sudah giat dikembangkan dan diimplementasikan dalam sistem keamanan lain seperti pengenalan retina dan iris mata (*retina and irish scan*), pengenal sidik jari (*finger-print*). Dalam wujudnya sendiri pengenalan wajah (*face recognition*) ini akan menggunakan alat penangkap citra seperti kamera untuk menangkap data wajah seseorang kemudian dibandingkan dengan wajah yang sebelumnya telah tersimpan di dalam *database* yang telah dibuat. [5]

## Algoritma Eigenface

*Eigenface* merupakan suatu algoritma pengenalan pola wajah berdasarkan pada *Principle Component Analysis (PCA)*. Prinsip dasar dari pengenalan wajah adalah dengan mengutip informasi unik wajah tersebut kemudian *encode* dan dibandingkan dengan hasil *decode* yang sebelumnya telah dilakukan.<sup>[1]</sup> Dalam metode algoritma *eigenface*, *decoding* dilakukan dalam menghitung sebuah vektor yaitu *eigenvector* dan setelah itu direpresentasikan dalam sebuah matrik yang berukuran besar. *Eigenvector* inilah yang kemudian dinyatakan sebagai *wert* atau karakteristik sebuah wajah. Setiap wajah akan direpresentasikan dalam kombinasi linear *eigenface*. Metode ini pertama kali dikembangkan oleh Matthew Turk dan Alex Pentland dari Vision and Modeling Group, The Media Laboratory, Massachusetts Institute of Technology pada tahun 1987. Metode ini disempurnakan lagi oleh Turk dan Pentland pada tahun 1991. [7]

**Metodologi**

Metodologi penelitian yang dilakukan untuk menganalisa permasalahan di atas dilakukan dalam beberapa tahapan, sebagai berikut :



**Gambar 1.** Diagram Alur Penelitian

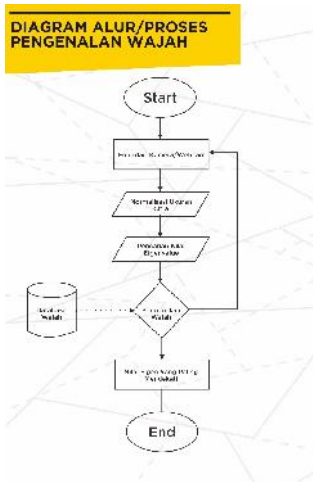
Keterangan Gambar 1:

1. Mencari literatur atau sumber pustaka yang membahas algoritma *Eigenface* dan ragam macam kegunaan serta penerapannya.
2. Melakukan perbandingan terhadap metode yang digunakan untuk menentukan kecocokan suatu wajah yang hendak dicocokkan.
3. Mengambil kesimpulan terhadap metode pencocokan terhadap suatu wajah yang hendak dicocokkan.

**2. Pembahasan**

**2.1 Diagram Alur Pengelan Wajah**

Proses atau alur tahapan pengenalan wajah seperti gambar berikut ini.



**Gambar X.** Diagram Alur Pengenalan Wajah

**2.2 Perhitungan Eigenface**

Perhitungan dengan menggunakan algoritma *Eigenface* dilakukan dengan beberapa tahapan seperti yang telah dijelaskan di atas. Langkah pertama yaitu menyiapkan kumpulan data dengan membuat suatu himpunan matriks yang ada di database, kemudian nilai tengah di cari, lalu setelah nilai tengah diperoleh maka akan dicari matriks Kovarian dari kumpulan matriks tersebut, menghitung nilai *eigen* dan vektor *eigen*. Dan diakhiri dengan identifikasi melalui kecocokan berdasarkan vektor-vektor tersebut.

**2.2.1 Penyusunan FlatVector Matriks**

Langkah pertama adalah menyusun himpunan S matriks yang terdiri dari seluruh *training image* (  $I_1, I_2, I_3, \dots, I_M$ ). *Training image* sendiri adalah gambar yang telah sebelumnya ditambahkan ke dalam *database* yang kemudian akan menjadi tolak ukur dalam pencocokan.

*Training Image 1*



**Gambar X.** Citra Wajah 1

Memiliki nilai matriks

$$C1 = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 2 & 4 \end{bmatrix}$$

*Training Image 2*



**Gambar X.** Citra Wajah 1

Memiliki nilai matriks

$$C1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 2 & 2 \end{bmatrix}$$

**2.2.2 Hitung Nilai Tengah atau Mean ( )**

Dari sekumpulan himpunan matriks yang telah diperoleh kemudian akan ditentukan nilai tengah atau mean ( ) dari kumpulan matriks tersebut. Yaitu dengan cara menjumlahkan matrik wajah 1, wajah 2, hingga wajah ke-n sesuai jumlah *training image* yang ada dalam *database*. Dalam hal ini hanya terdapat dua *training image* saja.

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

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n = \frac{1}{2} \begin{bmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 2 & 4 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 2 & 2 \end{bmatrix}$$

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

**2.2.3 Hitung selisih antara Matrik Training Image dengan Nilai Tengah (Mean / )**

Setelah nilai tengah atau mean ( ) diperoleh, kemudian selanjutnya mencari selisih ( ) antara matriks training image ( ) dengan nilai tengah ( ), dengan mengurangi training image ( ) dengan nilai tengah ( ).

$$\Phi_1 = \Gamma_1 - \Psi =$$

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

$$\Phi_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Phi_2 = \Gamma_2 - \Psi =$$

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

$$\Phi_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

**2.2.4 Hitung Nilai Matriks Kovarian**

Matriks Kovarian (C) dapat digunakan untuk memperoleh nilai eigen (eigenvalue / ) dan eigenvector (v).

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

$$L = A^T A$$

Hitung nilai matriks Kovarian (C)

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$L = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \rightarrow \text{Hasil Mat. Kovarian}$$

**2.2.5 Hitung Nilai Eigenvalue dan Eigenvector**

Setelah memperoleh hasil perhitungan matriks Kovarian maka selanjutnya akan mencari nilai eigen (eigenvalue / ) dan eigenvector (v) dari matriks Kovarian (C).

$$C x v_i = \lambda_i x v_i$$

Cari nilai eigen (eigenvalue) dan eigenvector (v):

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

$$L x v = I x v$$

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

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

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

Maka eigenvalue yang dihasilkan adalah:

$$\lambda_1 = 3, \lambda_2 = 1, \lambda_3 = 3$$

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

Eigenvector (v) diperoleh dengan melakukan substitusi nilai eigen ( ) dalam persamaan (I - L)v = 0. Eigenvector dari masing-masing nilai eigen dapat berdasarkan masing-masing kolom eigenvalue dan kemudian dihimpun kembali ke dalam satu matriks.

a. Untuk  $\lambda_1 = 3$ , maka :

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

Diperoleh eigenvector  $v_1$  adalah  $\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$

b. Untuk  $\lambda_2 = 1$ , maka :

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

Diperoleh *eigenvector*  $v_1$  adalah  $\begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix}$

b. Untuk  $n = 3$ , maka :

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

Diperoleh *eigenvector*  $v_3$  adalah  $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

Setelah memperoleh ketiga *eigenvector* tersebut maka matriks yang dihasilkan dari ketiganya adalah :

$$L = \begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

### 2.2.6 Hitung Nilai Eigenface

Setelah nilai L ditemukan, maka selanjutnya adalah menentukan nilai *eigenface* ( $\mu$ ) dengan rumus :

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

Cari nilai *Eigenface* ( $\mu$ ) :

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

$$\mu_i = v \chi \Phi_1$$

$$\mu_1 = \begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mu_1 = \begin{bmatrix} 0 & -1 & 0 \\ -2 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mu_2 = v \chi \Phi_2$$

$$\mu_2 = \begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\mu_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

### 2.2.7 Proses Identifikasi Wajah

Proses ini merupakan tahap untuk mengidentifikasi dan pencocokan dengan wajah yang sebelumnya telah

dimasukkan ke dalam *database*. Untuk mengenali wajah baru yang masuk atau yang lazim disebut dengan (*test face*), langkah yang dilakukan hampir sama yaitu dengan mencari nilai *eigenface* dari wajah yang baru (*test face*).

$$\mu_{new} = v \chi (\Gamma_{new} - \Psi)$$

$$\Omega = [\mu_1, \mu_2, \mu_3, \dots, \mu_M]$$

Tahapan pertama yaitu mencari selisih ( ) antara *test face* dengan nilai tengah ( ).



Gambar X. Citra Wajah Test Face

Memiliki nilai matriks

$$\text{Matriks} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 0 & 2 & 4 \end{bmatrix}$$

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

$$\Phi_{baru} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Selanjutnya, setelah selisih diperoleh maka nilai *eigenface* dapat diperoleh dengan.

$$\mu_{baru} = v \chi \Phi_{baru}$$

$$\mu_{baru} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

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

Setelah nilai *eigenface* yang baru (*test face*) ditemukan maka selanjutnya adalah mencari rentang selisih terkecil antara nilai *eigenface* wajah yang ada dalam *database* secara keseluruhan (*training image*) dengan nilai *eigenface* wajah yang ingin dicocokkan (*test face*) dengan menggunakan *euclidean distance*.

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

$$\varepsilon_1 = \|\Omega - \Omega_{test}\|$$

$$\varepsilon_2 = \|\Omega - \Omega_{\text{test}}\|$$

a.  $\varepsilon_1 = \|\Omega - \Omega_{\text{train}}\|$

$$\varepsilon_1 = \left\| \begin{bmatrix} 0 & -1 & 0 \\ -2 & -2 & 0 \\ 0 & 0 & 0 \\ -2 & -2 & -1 \\ 0 & 0 & 0 \end{bmatrix} \right\|$$

$$\varepsilon_1 = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\varepsilon_1 = \sqrt{(0)^2 + (1)^2 + (1)^2 + (0)^2 + (0)^2 + (1)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2}$$

$$\varepsilon_1 = \sqrt{3}$$

$$\varepsilon_1 = 1,73$$

b.  $\varepsilon_2 = \|\Omega - \Omega_k\|$

$$\varepsilon_2 = \left\| \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ -2 & -2 & -1 \\ 0 & 0 & 0 \end{bmatrix} \right\|$$

$$\varepsilon_2 = \begin{bmatrix} 0 & 1 & 1 \\ 2 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\varepsilon_2 = \sqrt{(0)^2 + (1)^2 + (1)^2 + (2)^2 + (1)^2 + (1)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2}$$

$$\varepsilon_2 = \sqrt{8}$$

$$\varepsilon_2 = 2,82$$

Dalam hasil pencocokannya, diperoleh rentang selisih jarak *eigenface* citra wajah satu (1,73) lebih kecil daripada *eigenface* citra wajah (2,82) dengan tolak ukur yaitu wajah yang ingin dicocokkan (*test face*) yang mengindikasikan bahwa *test face* lebih cocok atau lebih mirip dengan *training image* satu dibandingkan dengan *training image* dua.

### 3. Kesimpulan

Kesimpulan dari penelitian ini adalah hasil pengenalan 2 image wajah dengan menggunakan metode eigence yang selanjutnya dengan menggunakan euclidean distance, maka hasil perbandingan kecocokan wajah yang diperoleh, dengan pertama memiliki rentang kecocokan 1,73 sedangkan image kedua sebesar 2,82

### Daftar Pustaka

- [1] Al Fatta, Hanif. 2006. Sistem Presensi Karyawan Berbasis Pengenalan Wajah Dengan Algoritma *Eigenface*. Yogyakarta : STMIK AMIKOM.
- [2] Brigida. 2012. Pengenalan Pola. <http://informatika.web.id/komponen-sistem-pengenalan-pola.htm>.
- [3] Denny, M. 2012. Pengenalan *Computer Vision* dengan EmguCV di C#.Net.
- [4] Indra.2012. "Sistem Pengenalan Wajah Dengan Metode *Eigenface* Untuk Absensi Pada PT. Florindo Lestari", Seminar Nasional Teknologi Informasi dan Komunikasi Terapan 2012, ISBN: 979-26-0255-0, Semarang,23 Juni 2012,.
- [5] Rahim, M.A & dkk. 2013. "Perancangan Aplikasi Pengenalan Wajah Menggunakan Metode *Viol-Jones*". Seminar Nasional "Science, Engineering dan Teknologi" Medan : STMIK Budi Dharma, 2012.
- [6] Suprianto, D & dkk. 2013." Sistem Pengenalan Wajah Secara Real-Time, dengan *Adaboost, Eigenface PCA & MySQL*". Malang. Universitas Brawijaya Malang. Jurnal *EECCIS* vol 7, No 2 Desember 2013
- [7] Turk, M ad Pentland, A. 1991. *Face Recognition Using Eigenfaces*.[http://www.cs.ucsb.edu/~mturk/Papers/mturk\\_CVPR91.pdf](http://www.cs.ucsb.edu/~mturk/Papers/mturk_CVPR91.pdf)

### Biodata Penulis

**Norhikmah**, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Sistem Informasi STMIK AMIKOM Yogyakarta, lulus tahun 2012. Memperoleh gelar Magister Komputer (M.Kom). Program Pasca Sarjana Magister Teknik Informatika STMIK AMIKOM Yogyakarta, lulus tahun 2014. Saat ini menjadi Dosen di Fakultas ilmu komputer jurusan sistem informasi UNIVERSITAS AMIKOM Yogyakarta

**Aditya Dharma Satria S**, Seorang mahasiswa semester 5 fakultas ilmu komputer jurusan sistem informasi di Universitas AMIKOM

