

## BAB 2

### LANDASAN TEORI

Bab ini menjelaskan landasan teori dari metode yang digunakan dalam proses pengenalan wajah, yaitu terdiri atas: metode Eigenface, dan metode Jarak Euclidean. Metode Eigenface digunakan untuk melakukan ekstraksi ciri citra wajah, dan untuk mengukur tingkat kemiripan antar citra wajah yang sudah diekstrak tersebut digunakan metode Jarak Euclidean.

#### 2.1. Eigenface

Eigenface adalah metode yang digunakan untuk melakukan ekstraksi ciri citra wajah. Dalam bidang pengenalan wajah terdapat beberapa pendekatan untuk mengekstraksi ciri citra wajah, diantaranya adalah dengan mengidentifikasi ciri-ciri spesifik dari wajah seperti mata, alis, hidung, dan mulut. Metode lainnya adalah dengan menggunakan rasio jarak dari elemen-elemen wajah. Dalam perkembangannya, metode-metode tersebut tidak memberikan hasil yang akurat, dan pendekatan yang terbukti mampu memberikan hasil terbaik dalam melakukan ekstraksi ciri wajah adalah pemrosesan citra wajah secara keseluruhan [3]. Pendekatan ini berusaha untuk memperoleh semua variasi informasi dalam sekumpulan citra wajah dan tidak tergantung terhadap beberapa penilaian terhadap hubungan di antara ciri-ciri wajah. Beberapa metode telah diajukan untuk melakukan ekstraksi ciri wajah. Diantaranya adalah Eigenface, Fisherface, Neural Network, dan Elastic Matching. Eigenface merupakan metode yang paling sering digunakan dan terbukti berhasil dalam melakukan ekstraksi ciri wajah [3, 5]

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 dengan mengubah cara penghitungan matriks kovarian [3].

### 2.1.1. Metode Eigenface

Eigenfaces menggunakan *unsupervised statistical technique* yang cukup populer, yaitu *Principal Component Analysis (PCA)*. Metode PCA atau yang biasa dikenal dengan metode *Karhunen-Loven*, berusaha melakukan proyeksi dari ruang citra dengan dimensi yang lebih tinggi ke ruang ciri dengan dimensi yang lebih rendah. Reduksi dimensi dari ruang citra ke ruang ciri dilakukan untuk meningkatkan efisiensi dalam proses komputasi dan mengurangi *storage* yang diperlukan [3, 5, 6]. Selain itu reduksi dimensi dari vektor citra dengan dimensi yang lebih tinggi ke vektor ciri dengan dimensi yang lebih rendah akan memaksimalkan jarak antara semua kelas wajah [5].

### 2.1.2. Algoritma Eigenface

Algoritma Eigenface adalah sebagai berikut:

#### 1. Membuat matriks basis data citra wajah **I**

Sebuah citra wajah berukuran **w x h** (lebar x tinggi) adalah matriks piksel yang elemennya berisi nilai tertentu yang merepresentasikan intensitas warna pada piksel tersebut. Matriks piksel tersebut dapat direpresentasikan dalam vektor berdimensi **n** yang disebut juga vektor citra wajah **x<sub>i</sub>**, dimana **n** adalah **w x h**, dan **i** adalah index citra wajah. Matriks basis data citra wajah **I** adalah matriks berukuran **N x n** yang setiap barisnya berisi vektor-vektor citra wajah **x<sub>i</sub>**, dimana **N** adalah jumlah citra wajah. Representasi matriks **I** dapat dilihat pada Gambar 2.1 di bawah ini.

$$\mathbf{I} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \dots \\ \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ x_{31} & x_{32} & x_{33} & \dots & x_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ x_{N1} & x_{N2} & x_{N3} & \dots & x_{Nn} \end{bmatrix}$$

Gambar 2.1 : Matriks Basis Data Citra Wajah

2. Membuat matriks rata-rata  $\mu$

Rumus penghitungan matriks rata-rata adalah sebagai berikut:

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (1.1)$$

Dengan menggunakan rumus tersebut akan didapatkan  $\mu = [\mu_1 \ \mu_2 \ \mu_3 \ \mu_4 \ \dots \ \mu_n]$ .

3. Membuat matriks selisih  $A$

Matriks selisih  $A$  dibuat dengan mengurangi setiap elemen matriks  $I$  pada kolom ke- $i$  dengan  $\mu_i$ . Representasi matriks  $A$  dapat dilihat pada Gambar 2.2

$$A = \begin{pmatrix} X_{11}-\mu_1 & X_{12}-\mu_2 & X_{13}-\mu_3 & \dots & X_{1n}-\mu_n \\ X_{21}-\mu_1 & X_{22}-\mu_2 & X_{23}-\mu_3 & \dots & X_{2n}-\mu_n \\ X_{31}-\mu_1 & X_{32}-\mu_2 & X_{33}-\mu_3 & \dots & X_{3n}-\mu_n \\ \dots & \dots & \dots & \dots & \dots \\ X_{N1}-\mu_1 & X_{N2}-\mu_2 & X_{N3}-\mu_3 & \dots & X_{Nn}-\mu_n \end{pmatrix}$$

Gambar 2.2 : Matriks Selisih

4. Membuat matriks kovarian

Rumus penghitungan matriks kovarian atau *total scatter* matriks  $S_T$  adalah sebagai berikut:

$$S_T = A^T A \quad (1.2)$$

$S_T$  adalah matriks berdimensi  $n \times n$ .

5. Mengitung vektor eigen dan nilai eigen dari matriks kovarian

Rumus untuk menghitung vektor eigen dan nilai eigen dari matriks kovarian adalah sebagai berikut:

$$S_T v = \lambda v \quad (1.3)$$

Pada rumus diatas  $\mathbf{v}$  adalah vektor eigen, dan  $\lambda$  adalah nilai eigen.

#### 6. Memilih principal component

Pada proses 5 akan dihasilkan  $\mathbf{n}$  vektor eigen dan  $\mathbf{n}$  nilai eigen. Dari  $\mathbf{n}$  vektor eigen diambil  $\mathbf{m}$  vektor eigen  $\mathbf{v}$  yang memiliki nilai eigen terbesar. Karena vektor *eigen* ini memiliki dimensi yang sama dengan citra awal, maka vektor *eigen* ini juga biasa disebut *eigenpicture* atau Eigenface [3, 5].

#### 7. Melakukan ekstraksi citra wajah

Untuk melakukan proses ekstraksi citra wajah dilakukan perkalian antara matriks  $\mathbf{A}$  dan Eigenfaces. Representasi hasil ekstraksi basis data citra wajah atau  $\mathbf{ER}$  dapat dilihat pada Gambar 2.3.

$$\mathbf{ER} = \mathbf{Av} = \begin{pmatrix} X_{11} & X_{12} & X_{13} & \dots & X_{1n} \\ X_{21} & X_{22} & X_{23} & \dots & X_{2n} \\ X_{31} & X_{32} & X_{33} & \dots & X_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ X_{N1} & X_{N2} & X_{N3} & \dots & X_{Nn} \end{pmatrix} \begin{pmatrix} V_{11} & V_{12} & V_{13} & \dots & V_{1m} \\ V_{21} & V_{22} & V_{23} & \dots & V_{2m} \\ V_{31} & V_{32} & V_{33} & \dots & V_{3m} \\ \dots & \dots & \dots & \dots & \dots \\ V_{n1} & V_{n2} & V_{n3} & \dots & V_{nm} \end{pmatrix}$$

Gambar 2.3 : Matriks Eigenface

## 2.2. Jarak Euclidean

Berdasarkan hasil ekstraksi citra wajah menggunakan metode Eigenface, setiap citra wajah dinyatakan dalam vektor berdimensi  $\mathbf{m}$  yang disebut juga vektor citra wajah. Vektor citra wajah tersebut dapat dilihat sebagai sebuah titik dalam ruang berdimensi  $\mathbf{m}$ . Untuk mengukur tingkat kemiripan antara 2 citra wajah dapat digunakan jarak antara kedua vektor citra wajahnya. Semakin kecil jarak antara 2 vektor citra wajah, berarti kedua citra wajah tersebut memiliki tingkat kemiripan yang tinggi. Untuk mengukur jarak antara 2 vektor citra wajah, dapat digunakan metode Jarak Euclidean.

Dalam matematika Jarak Euclidean menyatakan jarak antara dua titik dalam suatu ruang. Secara umum, yang dimaksud jarak antara dua titik adalah garis terpendek diantara semua garis yang menghubungkan kedua titik tersebut. Dalam ruang

*euclidean* berdimensi  $m$ ,  $R^m$ , jarak antara titik  $x$  dan  $y$  dapat dirumuskan sebagai berikut:

$$D = \sqrt{\sum_{i=0}^m |x_i - y_i|^2} \quad (1.4)$$

Pemilihan Jarak Euclidean dikarenakan metode ini cocok digunakan menghitung jarak antara titik-titik piksel dari dua citra yang berbeda. Kombinasi Jarak Euclidean dengan PCA akan memaksimalkan jarak antar vektor citra wajah.

