

## BAB 2

### LANDASAN TEORI

Bab ini akan menjelaskan berbagai landasan teori yang digunakan oleh penulis dalam penelitian ini dan menguraikan hasil studi literatur yang telah dilakukan penulis. Bab ini terbagi ke dalam beberapa bagian yang masing-masing akan menjelaskan mengenai konsep-konsep inti yang mendasari penelitian ini.

#### 2.1. Tinjauan Umum Metode Pengenalan Wajah

Disebabkan oleh sifatnya yang dapat berubah-ubah, dan struktur tiga dimensinya yang kompleks, Gross (2005) menggolongkan proses pengenalan wajah menggunakan komputer ke dalam jenis persoalan yang tidak sederhana. Menurut Zhao dkk (2003), permasalahan pengenalan wajah dapat diformulasikan sebagai pernyataan umum berikut: “jika diberikan citra tunggal yang tidak bergerak (*still image*) atau rangkaian citra bergerak (*video*) dari sebuah situasi, identifikasi atau verifikasi satu atau lebih orang dari situasi tersebut, menggunakan basis data yang terdiri dari kumpulan wajah.”

Terkait dengan hal ini, tampilan suatu wajah dipengaruhi oleh banyak jumlah faktor, dan proses pengenalan wajah menjadi permasalahan dimana cakupan variabilitas dari pengenalan citra wajah menjadi sesuatu hal yang kompleks. Moghadam, Wahid dan Pentland (1998) menyebutkan adanya dua kelas variabilitas antara citra-citra wajah yang berbeda, yaitu variabilitas *extra-personal* dan variabilitas *intra-personal*. Variabilitas *extra-personal* timbul karena perbedaan identitas wajah, sedangkan variabilitas *intra-personal* adalah variasi yang muncul pada identitas wajah yang sama yang disebabkan oleh kondisi internal misalnya ekspresi wajah, umur, tata rambut dan elemen wajah lainnya.

Menurut Ming-Hsuan, Kriegman dan Ahuja (2002), faktor-faktor yang mempengaruhi sebuah sistem pengenalan wajah dapat digolongkan sebagai berikut:

- Posisi wajah. Pengenalan sebuah citra wajah tidak hanya dilakukan pada posisi wajah yang mengarah kedepan saja, tetapi juga pada posisi wajah yang lain.
- Fitur struktural wajah. Ciri-ciri wajah misalnya janggut, kumis dan kacamata bisa mempengaruhi tingkat variabilitas sebuah wajah.
- Ekspresi wajah. Penampilan sebuah wajah dipengaruhi secara langsung oleh ekspresi wajahnya.
- Keberadaan benda penghalang. Adakalanya ada benda-benda yang menghalangi wajah seseorang.
- Orientasi citra. Citra wajah dapat bervariasi menurut perbedaan rotasi poros kamera.
- Kondisi pencitraan. Salah satu contohnya adalah pencahayaan.

## 2.2. Metode Eigenface

Konsep Eigenface pertama kali dicetuskan oleh Turk dan Pentland pada tahun 1991. Sebelum memasuki penjelasan lebih rinci mengenai konsep ini, ada baiknya kita mengkaji prinsip dasar dan gambaran umum metode eigenface.

Metode Eigenface memperlakukan permasalahan pengenalan wajah sebagai pengenalan berdimensi dua, dengan alasan bahwa wajah pada umumnya tampil pada posisi vertical dan oleh karenanya dapat dideskripsikan oleh sekumpulan tampilan karakteristik dua dimensi. Citra-citra diproyeksikan ke dalam suatu "ruang wajah" yang paling baik mengkodekan variasi diantara citra-citra tersebut. Ruang wajah inilah yang oleh Turk dan Pentland (1991) dijuluki sebagai "eigenface", yaitu eigenvectors dari kumpulan wajah terkait.

Prinsip dasar dari metode eigenface adalah bagaimana caranya untuk mengekstrak informasi yang relevan dari sebuah citra wajah, mengubahnya ke dalam satu set kode yang paling efisien, dan membandingkan kode wajah ini dengan basis data berisi beragam wajah yang telah dikodekan secara serupa.

Turk dan Pentland menyebutkan bahwa secara matematis konsep eigenface

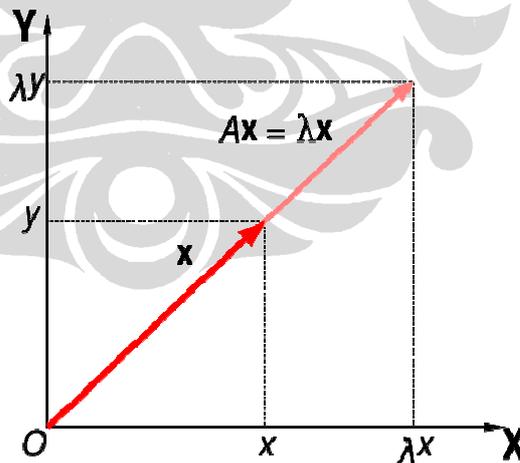
bertujuan untuk mencari *principal components* dari suatu distribusi wajah-wajah, yaitu Eigenvector dari matriks kovarian (*covariance matrix*) dari sekumpulan citra wajah. Dalam pembahasan ini, *principal component* dari sebuah matriks adalah vektor-vektor Eigen yang memiliki nilai Eigen terbesar.

Lebih lanjutnya, Eigenvector ini dapat dilihat sebagai sebuah kumpulan fitur yang secara bersama-sama akan merepresentasikan karakter dari variasi diantara citra-citra wajah. Dari gambaran konsep ini dapat dilihat bahwa yang perlu diuraikan dalam pembahasan berikutnya adalah konsep Eigenvector dan matriks kovarian.

### 2.2.1. Eigenvector dan Eigenvalue

Vektor Eigen (Eigenvector) dari sebuah transformasi linear didefinisikan sebagai suatu vektor non-nol dimana, ketika transformasi tersebut diaplikasikan kepadanya, vektor ini kemungkinan dapat berubah panjangnya, tetapi tidak mungkin berubah arahnya.

Untuk lebih jelasnya, diberikan gambar di bawah ini:



Gambar 0.1. Vektor Eigen

Pada gambar diatas,  $A$  merupakan sebuah transformasi linear. Vektor non-nol  $x$  didefinisikan sebagai Eigenvector dari transformasi  $A$  karena ia memenuhi persamaan  $Ax = \lambda x$  untuk suatu nilai skalar  $\lambda$ . Pada kasus ini, skalar  $\lambda$  merupakan

nilai Eigen (Eigenvalue) dari  $A$  yang bersesuaian dengan Eigenvector  $x$ .

### 2.2.2. Matriks Kovarian

Matriks kovarian (*covariance matrix*) adalah sebuah matriks yang terdiri dari kovarian-kovarian (*covariances*) antara elemen-elemen sebuah vektor. Untuk memahami definisi ini, kita perlu memahami definisi kovarian terlebih dahulu.

Kovarian merupakan sebuah ukuran akan bagaimana dua variabel berubah secara bersama-sama. Jika dua buah variabel cenderung bervariasi dengan cara serupa (misalkan, jika salah satu diantaranya berada diatas nilai ekspektasinya / *expected value* dan variabel yang lainnya juga cenderung berada diatas nilai ekspektasi dirinya), maka kovarian diantara kedua variabel tersebut bernilai positif. Sebaliknya, jika salah satu variabel itu berada diatas nilai ekspektasinya namun variabel yang lain justru berada dibawah nilai ekspektasi dirinya sendiri, maka nilai kovarian diantara keduanya akan menjadi negatif.

Dalam dunia statistik, nilai ekspektasi (*expected value*) dari sebuah variabel acak adalah integral dari variabel acak yang bersesuaian dengan pengukuran probabilitasnya. Jika variabel acak ini bersifat diskrit (*discrete*), maka nilai ekspektasi ini setara dengan penjumlahan probabilitas / kemungkinan dari seluruh nilai yang mungkin. Sebagai contoh, nilai ekspektasi dari pengocokan dadu bersisi enam adalah

$$E(X) = \frac{1 + 2 + 3 + 4 + 5 + 6}{6} = 3.5$$

(2.1)

Kembali pada pembahasan matriks kovarian, diberikan contoh di bawah ini:

Jika elemen-elemen dalam vektor

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \tag{2.2}$$

adalah variabel acak, masing-masing dengan varian yang nilainya terhingga

(finite), maka matriks kovarian  $\Sigma$  adalah matriks yang mana elemen  $(i, j)$ nya adalah kovarian:

$$\Sigma_{ij} = \text{cov}(X_i, X_j) = E[(X_i - \mu_i)(X_j - \mu_j)] \quad (2.3)$$

dimana

$$\mu_i = E(X_i) \quad (2.4)$$

adalah nilai ekspektasi dari elemen ke- $i$  dalam vektor  $X$ .

Dengan kata lain, matriks kovarian yang kita dapatkan adalah sebagai berikut:

$$\Sigma = \begin{bmatrix} E[(X_1 - \mu_1)(X_1 - \mu_1)] & E[(X_1 - \mu_1)(X_2 - \mu_2)] & \cdots & E[(X_1 - \mu_1)(X_n - \mu_n)] \\ E[(X_2 - \mu_2)(X_1 - \mu_1)] & E[(X_2 - \mu_2)(X_2 - \mu_2)] & \cdots & E[(X_2 - \mu_2)(X_n - \mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ E[(X_n - \mu_n)(X_1 - \mu_1)] & E[(X_n - \mu_n)(X_2 - \mu_2)] & \cdots & E[(X_n - \mu_n)(X_n - \mu_n)] \end{bmatrix} \quad (2.5)$$

### 2.2.3. Implementasi Eigenface

Langkah-langkah yang digunakan dalam metode Eigenface adalah sebagai berikut:

- Persiapan kumpulan citra latihan (*training set*). Setiap citra diubah ke dalam satu vektor, yaitu dengan cara menggabungkan baris-baris pixel dari citra aslinya. Citra *greyscale* dengan  $r$  buah baris dan  $c$  buah kolom misalnya, akan direpresentasikan sebagai vektor dengan  $r \times c$  buah elemen. Seluruh citra di kumpulan ini kemudian disimpan dalam satu matriks  $T$ , dimana setiap elemen barisnya mewakili sebuah citra.
- Penghitungan dan pengurangan citra tengah. Setiap citra asli dalam  $T$  harus dikurangi dengan citra tengah  $\mathbf{a}$ .
- Penghitungan vektor Eigen dan nilai Eigen dari matriks kovarian  $\mathbf{S}$ . Setiap vektor Eigen memiliki dimensi yang sama dengan citra aslinya, sehingga ia dapat dianggap sebagai sebuah citra. Vektor-vektor Eigen dari matriks kovarian inilah yang disebut sebagai Eigenfaces.

- Pemilihan *principal components*. Matriks kovarian berukuran  $D \times D$  akan menghasilkan vektor eigen sebanyak  $D$  buah. Yang harus kita lakukan adalah menyimpan vektor-vektor Eigen dengan nilai Eigen bersesuaian yang terbesar.

### 2.3. Metode Sparse Representation

Berkebalikan dengan metode Eigenface, metode Sparse Representation merupakan konsep yang masih sangat baru dikenal dalam bidang pengenalan wajah. Sebelumnya, konsep Sparse Representation ramai dibahas oleh komunitas pemrosesan sinyal, terutama berkisar mengenai penemuan fenomena bahwa jika representasi optimal telah cukup *sparse*, ia dapat secara efisien dihitung menggunakan optimisasi *convex* (Donoho, 2006).

Perkembangan studi ini diantaranya dipelopori oleh penemuan Candes, Romberg, dan Tao (2004), yaitu tentang bagaimana sinyal yang memiliki data frekuensi yang jauh dari lengkap ternyata dapat direkonstruksi secara tepat. Menindaklanjuti inilah Donoho kemudian menyatakan bahwa untuk nyaris setiap persamaan linear dari sistem *underdetermined* yang besar, solusi L1-norm minimalnya ternyata sekaligus merupakan solusi yang paling bersifat *sparse*.

#### 2.3.1. Pengenalan Wajah sebagai Representasi *Sparse* yang Linear

Metode pengenalan wajah yang digunakan dalam penelitian ini memanfaatkan penggunaan citra-citra wajah yang tersedia di dalam basis data dari kelas-kelas yang telah ditentukan untuk menemukan kelas dimana sebuah citra wajah yang akan dikenali seharusnya berada.

Wright, Yang, Ganesh, Sastry dan Yi (2008) berargumen bahwa dalam sebuah proses pengenalan wajah, citra wajah yang akan dikenali, yaitu citra uji, dapat direpresentasikan dalam metode Sparse Representation sebagai kombinasi linear dari citra-citra wajah yang tersedia di dalam basis data (citra latih).

Setiap citra wajah yang tersedia dalam ukuran berdimensi  $w \times h$  (dimana  $w$  adalah

ukuran lebar citra dan  $h$  adalah ukuran panjang atau tinggi citra) dapat dilihat sebagai sebuah titik dalam ruang  $\mathbb{R}^m$  dimana  $m = w \times h$ . Citra-citra dari satu wajah yang sama, namun dengan variasi *intra-personal* berbeda, membentuk sebuah subruang berdimensi rendah yang kurang lebih linear dalam ruang  $\mathbb{R}^m$ , yang disebut subruang wajah, sehingga  $n_i$  buah citra latih dari kelas ke- $i$  akan disusun sebagai kolom-kolom dari matriks  $A_i$  seperti berikut:

$$A_i = [v_{i,1}, v_{i,2}, \dots, v_{i,n_i}] \in \mathbb{R}^{m \times n_i} \quad (2.6)$$

Jika diberikan kumpulan citra latih yang memadai dari suatu kelas ke- $i$  (matriks  $A_i$ ), maka suatu citra uji  $y \in \mathbb{R}^m$  dari kelas yang sama akan terletak pada jangkauan linear citra-citra latih yang memiliki asosiasi dengan kelas  $i$  tersebut:

$$y = \alpha_{i,1}v_{i,1} + \alpha_{i,2}v_{i,2} + \dots + \alpha_{i,n_i}v_{i,n_i}, \quad (2.7)$$

untuk suatu bilangan skalar  $\alpha_{i,j} \in \mathbb{R}, j = 1, 2, \dots, n_i$ .

Karena status keanggotaan  $i$  dari citra uji diatas masih belum diketahui, maka harus didefinisikan sebuah matriks baru,  $A$ , sebagai gabungan dari seluruh  $n$  buah citra latih dari seluruh kelas-kelas yang berjumlah  $k$ :

$$A \doteq [A_1, A_2, \dots, A_k] = [v_{1,1}, v_{1,2}, \dots, v_{k,n_k}]. \quad (2.8)$$

Dengan ini, representasi linear dari  $y$  dapat ditulis dalam ruang lingkup seluruh citra latih sebagai:

$$x_0 = [0, \dots, 0, \alpha_{i,1}, \alpha_{i,2}, \dots, \alpha_{i,n_i}, 0, \dots, 0]^T \in \mathbb{R}^n \quad (2.9)$$

dimana  $y = Ax_0 \in \mathbb{R}^m$ , adalah sebuah vektor koefisien dimana seluruh nilai elemennya adalah nol kecuali yang berasosiasi dengan kelas ke- $i$ .

Dapat dilihat bahwa vektor  $x_0$  disini menjadi kunci untuk dapat mengetahui

identitas citra uji  $y$ . Salah satu kelebihan metode ini adalah tidak adanya kenaikan *order of growth* dalam komputasi yang bersangkutan. Dengan kata lain, kompleksitas komputasi tetap linear terhadap ukuran citra latih.

Tampak jika  $m > n$ , maka sistem persamaan  $y = Ax$  bersifat *overdetermined* sehingga solusi  $x_0$  merupakan solusi yang unik. Namun bagaimanapun, dalam permasalahan pengenalan wajah yang umum ditemui, sistem persamaan  $y = Ax$  biasanya bersifat *underdetermined*, dan oleh karenanya tidak memiliki solusi yang unik.

Kesulitan diatas dapat diatasi dengan memilih solusi normalisasi-L2 yang minimum:

$$(\ell^2) : \quad \hat{x}_2 = \arg \min \|x\|_2 \quad \text{untuk } Ax = y \quad (2.10)$$

Namun walaupun permasalahan optimisasi ini dapat diselesaikan mudah melalui pseudoinvers dari matriks  $A$ , solusi yang didapat,  $x_2$ , belum memberikan cukup informasi yang diperlukan untuk mengenali citra uji  $y$ .

Sebaran nilai-nilai pada  $x_2$  pada umumnya bersifat padat, dengan nilai-nilai non-nol yang besar, yang berasosiasi dengan berbagai citra latih dari banyak kelas yang berbeda.

Sebagai jalan keluarnya, pertama-tama kita harus menyadari bahwa sebuah citra uji  $y$  yang *valid* seharusnya dapat direpresentasikan secara memadai dengan hanya menggunakan sekumpulan citra latih dari satu kelas yang sama. Representasi ini sudah sewajarnya bersifat *sparse* jika jumlah kelas yang ada,  $k$ , cukup besar. Semakin bersifat *sparse* solusi  $x_0$  yang diperoleh, semakin mudah untuk menentukan identitas *test sample* (citra wajah yang ingin dikenali) secara akurat. Oleh karena ini dapat dilihat betapa perlunya kita mencari solusi yang bersifat paling *sparse* untuk persamaan  $y = Ax$ , yang dengan demikian memecahkan persamaan optimisasi dibawah ini:

$$(\ell^0) : \quad \hat{x}_0 = \arg \min \|x\|_0 \quad \text{untuk } Ax = y \quad (2.11)$$

dimana  $\|\cdot\|_0$  mewakili norm-L0, yang memberikan jumlah nilai-nilai non-nol dalam suatu vektor. Akan tetapi, permasalahan dalam mencari solusi paling *sparse* dari sebuah sistem persamaan linear yang bersifat *underdetermined* merupakan persamaan yang bersifat NP-hard, yang bahkan untuk dilakukan aproksimasi atasnya pun sulit. Jalan keluar untuk ini akan dibahas dalam pembahasan berikut.

### 2.3.2. Solusi *Sparse* melalui Minimisasi-L1

Wright, Yang, Ganesh, Sastry dan Yi (2008) menyebutkan bahwa penemuan-penemuan terbaru di bidang representasi *sparse* dan *compressed sensing* menemukan bahwa jika solusi  $x_0$  yang dicari seperti tersebut diatas bersifat cukup *sparse*, maka solusi dari permasalahan minimisasi-L0 tersebut adalah sama dengan solusi yang dapat diperoleh dari permasalahan minimisasi-L1 dibawah ini:

$$(\ell^1) : \quad \hat{x}_1 = \arg \min \|x\|_1 \quad \text{untuk } Ax = y \quad (2.12)$$

Chen, Donoho dan Saunders (2001) menyatakan bahwa persoalan diatas dapat dipecahkan dalam jangka waktu *polynomial* dengan menggunakan metode-metode pemrograman linear yang standar.

### 2.3.3. Algoritma Lasso

Dalam dunia statistik, metode pemecahan permasalahan diatas menyerupai sebuah algoritma yang dikenal dengan nama algoritma Lasso. Lasso merupakan metode yang cukup baru diperkenalkan dalam estimasi model linear. Lasso meminimisasi *residual sum of squares* (RSS) yang rentan terhadap penjumlahan dari nilai absolut koefisien-koefisien yang nilainya kurang dari suatu konstanta (*constant*).

RSS sendiri didefinisikan sebagai perbedaan antara data dan model estimasi yang kita gunakan. Semakin kecil nilai perbedaan ini, akan semakin baik estimasi tersebut. Rumus umum nilai ini adalah:

$$RSS = \sum_{i=1}^n (y_i - f(x_i))^2. \quad (2.13)$$

Untuk model regresi standar  $y_i = a + bx_i + \varepsilon_i$  dimana a dan b merupakan koefisien dan  $\varepsilon$  merupakan nilai *error*, nilai RSS menjadi nilai penjumlahan atas pangkat dua dari estimasi  $\varepsilon_i$ , yaitu:

$$RSS = \sum_{i=1}^n (y_i - (a + bx_i))^2. \quad (2.14)$$

#### 2.3.4. Klasifikasi melalui Koefisien *Sparse*

Klasifikasi disini diperlukan untuk mengatasi *error* yang disebabkan oleh adanya kemungkinan *noise* dan permasalahan permodelan dalam sistem pengenalan wajah di dunia nyata, termasuk didalamnya variabilitas *extra-personal* dan *intra-personal* citra wajah.

Idealnya, jika diketahui sebuah citra uji  $y$  dari salah satu kelas-kelas dalam basis data, kita akan dapat memperoleh representation *sparse*-nya,  $x_1$ , dengan nilai-nilai non-nol dalam  $x_1$  berasosiasi dengan kolom-kolom A dari obyek tunggal dari kelas  $i$ , dan dengan demikian solusi yang tepat akan diperoleh.

Tetapi disebabkan kondisi non-ideal yang telah disebutkan diatas, kita memerlukan *classifier* untuk menentukan klasifikasi, untuk menentukan citra manakah dalam basis data yang berasosiasi secara paling kuat dengan citra wajah yang ingin dikenali.

Sebetulnya terdapat banyak cara untuk merancang *classifier* dalam persoalan ini, tetapi diketahui bahwa salah satu cara terbaik dan terefisien adalah dengan

mengklasifikasi  $y$  berdasarkan sebaik apa suatu koefisien berasosiasi dengan seluruh training samples dari setiap obyek wajah yang menghasilkan  $y$ .

Proses klasifikasi ini berlangsung sebagai berikut: untuk setiap kelas  $i$ , dibangun fungsi karakteristik  $\delta_i : \mathbb{R}^n \rightarrow \mathbb{R}^n$  yang memilih koefisien-koefisien yang berasosiasi dengan kelas ke- $i$ . Untuk  $x \in \mathbb{R}^n$ ,  $\delta_i(x) \in \mathbb{R}^n$  adalah sebuah vektor baru dimana nilai-nilai non-nol yang ada hanyalah nilai-nilai di  $x$  yang berasosiasi dengan kelas  $i$ . Hanya dengan memanfaatkan koefisien yang berasosiasi dengan kelas ke- $i$ , kita dapat melakukan aproksimasi pada citra uji yang diberikan sebagai  $y_i = A\delta_i(x_1)$ .

Selanjutnya kita akan dapat mengklasifikasi  $y$  berdasarkan aproksimasi diatas, yaitu dengan cara memasangkannya dengan kelas yang meminimumkan selisih antara  $y$  dan  $y_i$ .

