

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Pengolahan Citra**

Pengolahan citra adalah suatu sistem dimana proses yang dilakukan dengan masukan (*input*) berupa citra (*image*) dan hasilnya (*output*) juga citra (*image*). Pengolahan citra digunakan untuk memanipulasi gambar yang ditangkap oleh kamera menjadi gambar yang siap digunakan pada proses selanjutnya[6]. Pada pengenalan wajah diperlukan proses pengolahan citra yang merupakan suatu metode untuk melakukan beberapa operasi pada gambar, untuk mendapatkan gambar yang disempurnakan atau untuk mengekstrak beberapa informasi yang berguna dari gambar tersebut.

Citra adalah representasi dua dimensi untuk bentuk-bentuk fisik nyata tiga dimensi. Citra dalam perwujudan dapat bermacam-macam, mulai dari gambar perwujudan nya dapat bermacam-macam, mulai dari gambar putih pada sebuah foto (yang tidak bergerak) sampai pada gambar warna yang bergerak pada televisi[7]. Proses transformasi dari bentuk dua dimensi untuk menghasilkan citra akan dipengaruhi oleh bermacam macam faktor yang mengakibatkan penampilan citra suatu benda tidak sama persis dengan bentuk fisik nyatanya. Faktor-faktor tersebut merupakan efek degradasi atau penurunan kualitas yang dapat berupa rentang kontras benda yang terlalu sempit atau terlalu lebar, distorsi geometrik kekaburan (*blur*), kekaburan akibat objek citra yang bergerak 9 *motion blur*, *noise* atau gangguan yang disebabkan oleh interferensi pembuat citra, baik itu pembuat transduser, peralatan elektronik maupun peralatan optik.

#### **2.2 Pengenalan Wajah**

Pengenalan wajah adalah teknologi yang digunakan untuk mengidentifikasi identitas citra seseorang. Pengenalan wajah ini adalah salah satu aplikasi paling populer untuk analisis dan identifikasi citra. Sistem ini termasuk kategori biometrik untuk sistem keamanan dan privasi[8]. Pengenalan wajah dirancang untuk

mendeteksi dan mengenali orang saat citra mereka diambil dengan lensa kamera[9]. Teknologi pengenalan wajah dapat digunakan dalam berbagai aplikasi seperti otentikasi identitas, akses kontrol, dan pengawasan. Kegiatan dan minat penelitian dalam bidang pengenalan wajah telah meningkat secara signifikan dalam beberapa tahun terakhir[10].

Dengan teknologi yang sudah ada, pengenalan citra wajah diproses melalui sistem digital, citra-citra wajah tersebut diolah oleh sistem komputer dengan menggunakan teknologi kecerdasan buatan untuk memperoleh hasil dari segala bentuk perubahan yang terjadi pada wajah. Hal ini memungkinkan kepolisian dapat mengenali pola wajah seorang penjahat walaupun seorang penjahat tersebut mengenakan kacamata, memotong rambut, janggut, bahkan perubahan wajah karena bertambahnya umur[11].

### 2.3 *Principal Component Analysis (PCA)*

*Principal component analysis* (PCA) digunakan untuk mengekstrak fitur dari citra wajah. Secara sederhana PCA adalah transformasi linear untuk menentukan sistem koordinat yang baru dari sebuah dataset[12]. Teknik PCA ini mereduksi informasi data yang besar dari sebuah citra wajah tanpa menghilangkan informasi yang ada pada sebuah citra wajah tersebut.

Misal citra wajah  $I(x,y)$  berupa array  $N = w \times h$  nilai identitas dua dimensi atau vektor dimensi  $N^2$ . Sebuah tipikal gambar berukuran  $92 \times 112$  menggambarkan vektor dimensi 10.304, atau setara sebuah titik dalam ruang 10.304 dimensi. Fungsi utama dari PCA ini adalah menemukan nilai vektor yang terbaik untuk distribusi gambar wajah dalam seluruh ruang dimensi gambar. Berikut langkah-langkah untuk mengekstraksi fitur PCA:

1. Mendapatkan satu set gambar  $M$  dengan ukuran  $N = w \times h$ , dengan  $w$  adalah lebar dan  $h$  adalah tinggi citra dapat direpresentasikan dengan  $I_1, I_2, I_3, \dots, I_M$ , sehingga vektor berukuran  $N^2$ . Citra wajah dapat dituliskan sebagai  $\Gamma_i = \{\Gamma_1, \Gamma_2, \dots, \Gamma_M\}$ , dimana setiap  $\Gamma_i$  adalah vektor dari dimensi  $N$  dan jumlah citra wajah dalam citra latih disimbolkan oleh  $M$ .

2. Representasikan setiap citra  $I_i$  menjadi vector  $\Gamma_i$  sebagai berikut:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} \xrightarrow{\text{concatenate}} \begin{bmatrix} a_{11} \\ \vdots \\ a_{1N} \\ \vdots \\ a_{2N} \\ \vdots \\ a_{NN} \end{bmatrix} = \Gamma_i \quad (2.1)$$

3. Rata-rata set pelatihan ditentukan oleh

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i \quad (2.2)$$

4. Mencari rata-rata dari beberapa wajah yang berbeda dengan vektor  $\{\Phi_1, \Phi_2, \Phi_3, \dots, \Phi_M\}$ .

$$\Phi_i = \Gamma_i - \Psi \quad (2.3)$$

5. Mencari matriks kovarian dengan persamaan sebagai berikut:

$$C = A \times A^T, \text{ dimana } A = [\Phi_1, \Phi_2, \Phi_3, \dots, \Phi_M] \quad (2.4)$$

Dalam kasus ini, kovarian matrik  $C$  mempresentasikan hubungan antara 2 matrik dan varian yang terdapat dalam dataset. Sedangkan  $A^T$  adalah transpose dari matriks  $A$ , dimana  $A$  adalah selisih dari masing-masing citra latih dengan rata-rata dari citra wajah (*average face*). Matriks kovarian ini adalah matriks yang dapat menghasilkan vektor eigen dan nilai eigen yang kita butuhkan.

6. Dikarenakan matriks  $w \times h$  dari citra wajah dianggap sebagai vektor dimensi  $N$ , menunjukan bahwa matriks  $C$  berdimensi  $N^2$ . Untuk menghindari perhitungan terhadap matrik kovarian yang besar dengan menggunakan persamaan sebagai berikut:

$$C \times u_i = \lambda_i \times u_i \quad (2.5)$$

Dimana  $u_i$  adalah vektor eigen, dan  $\lambda_i$  adalah nilai eigen dari citra wajah yang bersangkutan.

7. Mendapatkan vektor eigen dan nilai eigen dari matrik kovarian ( $C$ ). Kita misalkan  $L = A^T \times A$ . Matrik  $L$  adalah matrik dengan dimensi  $M \times M$  dan  $M$  lebih kecil dari  $N$ , sehingga lebih mudah untuk menghitung vektor eigen dan nilai eigen dari matrik tersebut yang dipecahkan melalui persamaan berikut:

$$\begin{aligned}
L \times V_i &= d_i \times V_i \\
A^T \times A \times V_i &= d_i \times V_i \\
A \times A^T \times A \times V_i &= d_i \times A \times V_i \\
C \times A \times V_i &= d_i \times A \times V_i
\end{aligned}$$

Berdasarkan persamaan diatas, maka  $u_i = A \times V_i$  dan  $\lambda_i = d_i$  adalah vektor eigen dan nilai eigen dari matrik kovarian, sehingga akan didapatkan vektor eigen dan nilai eigen sebanyak  $M$  dari matrik  $L$ .

8. Selanjutnya pilih  $K$  eigenvektor terbaik, pemilihan secara heuristik. sebelum melakukan perhitungan bobot,  $\Phi_i$  direpresentasikan dengan kombinasi linear dari eigenvektor  $U_i$  sebagai berikut:

$$\Phi_i = \sum_{j=1}^K w_j u_j, \text{ dimana } u_j \text{ adalah eigenface.} \quad (2.6)$$

Sehingga nilai bobot dapat dihitung dengan rumus:

$$w_j = u_j^T \Phi_i \quad (2.7)$$

Selanjutnya citra training yang sudah dinormalisasi, direpresentasikan menjadi sebuah vektor berikut:

$$\Omega = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_k \end{bmatrix} \quad (2.8)$$

Dimana  $i = 1, 2, 3, \dots, M$ . Hasil akhirnya, vektor yang telah dihitung sesuai dengan setiap gambar pada *training set* akan disimpan sebagai *template*. Setelah vektor fitur (bobot vektor) diketahui, lalu tahap selanjutnya adalah tahap pengenalan dengan menggunakan klasifikasi.

## 2.4 Klasifikasi

Klasifikasi pada penelitian ini merupakan proses pemisahan hasil ekstraksi ciri untuk memisahkan dua kelas yaitu wajah dan non-wajah. Jenis-jenis *classifier* terbagi dalam tiga jenis utama yaitu:

1. *Classifier* berdasarkan keputusan Bayes

*Classifier* ini mencoba melakukan klasifikasi  $M$  kelas ( $w_1, w_2, \dots, w_M$ ) dan pola yang tidak diketahui yang dipresentasikan dengan vektor  $x$  dengan membentuk probabilitas kondisional sebanyak  $M$  ( $P(w_i, x), i = 1, 2, \dots, M$ ). nilai tersebut mewakili probabilitas bahwa pola yang tidak diketahui tersebut tergolong kedalam kelas  $w_i$  tertentu. *Classifier* mencari nilai kemungkinan terbesar tersebut untuk melakukan klasifikasi.

## 2. *Linear Classifier*

*Classifier* mencoba melakukan klasifikasi dengan menggunakan persamaan *linear* sederhana. Keunggulan *classifier* ini adalah simplisitasnya dan metode komputasi yang digunakan.

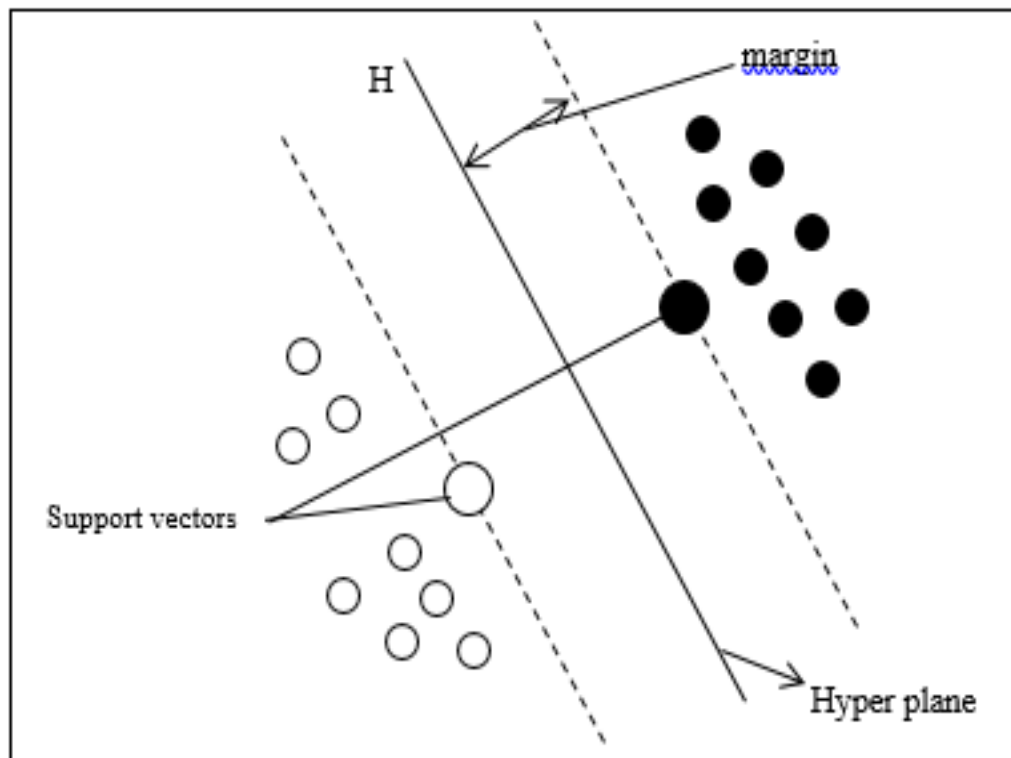
## 3. *Non linear Classifier*

*Classifier* ini merupakan pengembangan dari *linear classifier*. Untuk kasus-kasus dimana kelas-kelas tidak dapat dipisahkan secara *linear*, *classifier* ini dirancang secara optimal dengan persyaratan tertentu, misalnya dengan meminimalkan kuadrat *error* yang dihasilkan.

## 2.5 *Support Vector Machine (SVM)*

*Support Vector Machine (SVM)* yaitu metode klasifikasi yang diperkenalkan pertama kali oleh Vapnik pada tahun 1998. Pada dasarnya, metode ini bekerja dengan cara mendefinisikan batas antara dua kelas dengan jarak maksimal dari data yang terdekat. Algoritma ini diterapkan pada ruang fitur dimensi tinggi. Hal ini meningkatkan kesulitan untuk menghitung produk skalar dari ruang fitur yang ada. Adapun fungsi kernel digunakan untuk menghitung produk skalar ini. Dengan menggunakan fungsi kernel tidak memerlukan perhitungan ruang fitur secara eksplisit[13].

Metode SVM awalnya dikembangkan sebagai pengklasifikasi linear. Dengan memanfaatkan metode kernel ini juga diterapkan untuk pemetaan data *non-linear*. Cara pemisahan data dengan metode SVM ditunjukkan pada contoh yang disederhanakan pada Gambar 2.1.



Gambar 2. 1 Pemisahan Data dengan Metode SVM

Data  $p$ -dimensi dipisahkan oleh SVM menggunakan bidang yang berdimensi  $p-1$  (*hyperplane*) sedemikian rupa sehingga memaksimalkan margin dari pengumpulan data, margin didefinisikan sebagai jarak minimal sampel ke *hyperplane*.

Proses pembelajaran pada SVM bertujuan untuk mendapatkan hipotesis berupa bidang pemisah terbaik yang tidak hanya meminimalkan *empirical risk* yaitu rata-rata *error* pada suatu data pelatihan, tetapi juga memiliki generalisasi yang baik. Generalisasi adalah kemampuan sebuah hipotesis untuk mengklasifikasikan data yang tidak terdapat dalam data pelatihan dengan benar. Untuk menjamin generalisasi ini, SVM bekerja berdasarkan prinsip *Structural Risk Minimization* (SRM). Berikut merupakan algoritma SVM:

1. Ide fitur-fitur dari data dipetakan kedalam ruang dimensi lebih tinggi menggunakan fungsi kernel (*linear, polynomial, radial basis function, sigmoid*).

2. Mencari *hyperplane* terbaik yang memisahkan data yang telah dipetakan dalam ruang dimensi tinggi.
3. *Hyperplane* terbaik dapat diperoleh dengan memaksimalkan jarak *hyperplane* dengan titik data terdekat (*Support Vector*).

### 2.5.1 Fungsi Kernel Pada SVM

Dengan menggunakan fungsi kernel, maka dapat menghasilkan fitur-fitur baru yang akan digunakan untuk mengklasifikasi. Fitur-fitur dari data dipetakan kedalam ruang dimensi tinggi menggunakan fungsi kernel (*linear, polynomial, sigmoid, atau radial basis function*).

Tabel 2. 1 Fungsi Kernel Berdasarkan Jenisnya

Jenis Kernel	Fungsi
Linear	$K(x_i, x_j) = x_i^T x_j$
Polynomial	$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \quad \gamma > 0$
Sigmoid	$K(x_i, x_j) = \tanh(x_i^T x_j + r)$
Radial Basis Function	$K(x_i, x_j) = \exp(-\gamma \ x_i^T - x_j\ ^2), \gamma > 0$

## 2.6 Python

Python merupakan sebuah Bahasa pemrograman yang bersifat *open source*. Python dibuat oleh Guido Van Rossum pertama kali di Centrum Wiskunde & Informatica (CWI) di Belanda pada awal tahun 1990-an. Python dapat dijalankan pada berbagai sistem operasi. Kelebihan lain dari pemrograman Python adalah bentuk program yang sederhana sehingga mudah untuk dipelajari [14].

Pada penelitian ini menggunakan Bahasa pemrograman Python dan juga dilengkapi dengan beberapa *library* yang salah satunya adalah *library* OpenCV. Yang mana OpenCV merupakan sebuah *library* yang bersifat *open source* yang dikembangkan oleh Intel dan fokus untuk menyederhanakan pemrograman terkait dengan pengolahan citra digital. OpenCV memiliki banyak fitur terkait visi komputer (*computer vision*) antara lain: pengenalan wajah, deteksi wajah, Kalman

filtering, dan berbagai jenis metode AI (*Artificial Intellegence*). OpenCV dapat berjalan di berbagai Bahasa pemrograman seperti, C, C++, Java, Python, dan juga mendukung berbagai platform sistem operasi seperti, Windows, Linux, Mac OS, IOS, dan Android[15].

## 2.7 Data Set Wajah Yang Digunakan

### 2.7.1 Wajah AT&T

Dataset wajah AT&T berisi citra wajah yang diambil antara April 1992 dan 1994. Ada 10 data wajah yang berbeda dari masing-masing 40 subjek berbeda. Semua citra gambar diambil dengan latar belakang gelap dengan posisi subjek tegak. Dataset tersebut berformat PGM, dengan ukuran setiap citra adalah  $92 \times 112$  piksel, dengan 256 tingkat keabuan per piksel[16]. Dibawah ini adalah contoh citra dataset dari AT&T.



Gambar 2. 2 Dataset AT&T[16]



Tujuan dari penggunaan penggunaan dataset AT&T yaitu sebagai percobaan pegujian pada sistem pengenalan, sebelum sistem diuji dengan menggunakan dataset yang dibuat secara mandiri. Selain itu, dataset AT&T juga digunakan sebagai pembanding hasil tingkat akurasi pengenalan dengan dataset lainnya.

### **2.7.2 Data Wajah Georgia Tech**

Data wajah Georgia Tech berisi gambar 50 orang orang yang diambil dalam dua atau tiga sesi antara 01/06/99 dan 15/11/99 di Pusat Pemrosesan Sinyal dan Gambar di Institut Teknologi Georgia[17].

Semua orang dalam database diwakili oleh 15 gambar JPEG berwarna dengan *background* berantakan yang diambil pada resolusi 640×480 piksel. Ukuran rata-rata wajah dalam gambar ini adalah 150×150 piksel. Gambar menunjukkan wajah depan dan/atau miring dengan ekspresi wajah, kondisi pencahayaan, dan skala yang berbeda. Setiap gambar diberi label secara manual untuk menentukan posisi wajah dalam gambar.



Gambar 2. 3 Dataset Georgia Tech

### 2.7.3 Data Wajah Yale

Basis data berisi 5760 gambar sumber cahaya tunggal dari 10 subjek yang masing-masing terlihat di bawah 576 kondisi tampilan ( $9 \text{ pose} \times 64 \text{ kondisi iluminasi}$ ). Untuk setiap subjek dalam pose tertentu, gambar dengan pencahayaan sekitar (latar belakang) juga ditangkap. Oleh karena itu, jumlah total gambar sebenarnya adalah  $5760 + 90 = 5850$ . Ukuran total database terkompresi adalah sekitar 1GB.

Gambar dalam database diambil menggunakan rig penerangan. Rig ini dilengkapi dengan 64 strobo yang dikendalikan komputer. 64 gambar subjek dalam pose tertentu diperoleh pada kecepatan bingkai kamera ( $30 \text{ frame/detik}$ ) dalam waktu sekitar 2 detik, jadi hanya ada sedikit perubahan pada pose kepala dan ekspresi wajah untuk 64 gambar (+1 ambient) tersebut[18].



Gambar 2. 4 Dataset Yale

#### 2.7.4 Data Wajah mandiri

Dataset mandiri merupakan data wajah yang diambil secara mandiri oleh peneliti dengan menggunakan *webcam* beresolusi  $640 \times 480$  piksel. Terdapat 10 data wajah yang berbeda dari masing 40 subjek berbeda. Semua citra gambar diambil dengan posisi subjek tegak dan berbeda ekspresi dengan latar belakang dan pencahayaan yang berbeda serta berwarna. Dataset ini dibuat sebagai ciri perwakilan dari wajah lokal (orang Indonesia) untuk mendukung penelitian dibidang pengenalan wajah. Dataset tersebut berformat JPG dengan ukuran setiap citra adalah  $92 \times 112$  piksel. Dibawah ini adalah contoh citra dari dataset Mandiri.



Gambar 2. 5 Dataset Mandiri

## 2.8 Confusion Matrix

Pada dasarnya *confusion matrix* memberikan informasi tentang perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi yang sebenarnya. *Confusion matrix* dalam bentuk tabel menggambarkan kinerja model klasifikasi pada serangkaian data uji dengan nilai nyata yang diketahui[19]. Berikut merupakan gambar *confusion matrix* dengan 4 kombinasi nilai prediksi dan nilai aktual yang berbeda:

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	<p><b>TP</b> (True Positive)</p>	<p><b>FP</b> (False Positive) <i>Type I Error</i></p>
	0 (Negative)	<p><b>FN</b> (False Negative) <i>Type II Error</i></p>	<p><b>TN</b> (True Negative)</p>

Gambar 2. 6 *Confusion Matrix*

Terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada *confusion matrix*, yaitu sebagai berikut:

1. *True Positive* (TP) yaitu, kondisi dimana data positif yang telah diprediksi dengan hasil benar.
2. *False Positive* (FP) yaitu, kondisi dimana data negatif diprediksi sebagai data positif.
3. *True Negative* (TN) yaitu, kondisi dimana data negatif yang diprediksi benar.
4. *False Negative* (FN) yaitu, kondisi dimana data positif tetapi diprediksi sebagai data negatif.

Dari istilah-istilah tersebut maka didapatkanlah rumus *confusion matrix* untuk melakukan perhitungan mendapatkan akurasi, presisi, dan recall sebagai berikut:

$$\text{akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.9)$$

$$\text{presisi} = \frac{TP}{TP + FP} \quad (2.10)$$

$$\text{recall} = \frac{TP}{TP + FN} \quad (2.11)$$