

## **BAB II**

### **DASAR TEORI**

#### **2.1 Pengolahan Citra**

Pengolahan Citra atau *Image Processing* merupakan sebuah proses untuk memperbaiki kualitas citra pada gambar agar dapat mudah diinterpretasi oleh manusia dan komputer, juga dapat dikatakan proses memanipulasi dan memodifikasi gambar dengan berbagai macam teknik dan algoritma tertentu sehingga menghasilkan keluaran yang lebih baik atau mengubahnya dengan keluaran berbeda. Sedangkan, pada sebuah citra dapat direpresentasikan sebagai susunan beberapa piksel berkolom dan berbaris yang memiliki nilai atau informasi sehingga piksel dapat dihitung secara matematis [7].

Pada pengolahan citra terbagi kedalam beberapa kategori pemrosesannya, yaitu kategori rendah, menengah, dan tinggi. Kategori rendah melibatkan operasi-operasi sederhana seperti prapengolahan citra untuk mengurangi derau atau *noise*, mengatur kontras, dan ketajaman citra. Pada kategori ini memiliki input dan output berupa citra. Sementara, pada pengolahan kategori menengah melibatkan beberapa operasi seperti segmentasi dan klasifikasi citra. Terakhir, pada kategori tinggi melibatkan proses pengenalan dan pendeskripsian citra, contohnya citra uji akan mencocokkan dengan citra latih pada fase pengenalannya.

#### **2.2 Pengenalan Wajah**

Pengenalan wajah merupakan salah satu teknologi biometrik yang mengaplikasikan wajah sebagai identitasnya, pada proses pengenalan terdapat beberapa tahapan dan metode yang akan digunakan pada sistemnya. Dalam penerapannya sendiri pengenalan wajah menggunakan sebuah perangkat kamera atau *webcam* untuk menangkap wajah seseorang kemudian membandingkan wajah dengan basis data yang telah tersimpan [6].

Secara umum, terdapat empat tahapan dalam sistem pengenalan wajah yaitu tahap akuisisi citra wajah, prapemrosesan, fitur ekstraksi, dan klasifikasi pada citra wajah. Dari keempat tahapan tersebut tahap yang paling penting dalam sistem

pengenalan wajah yaitu fitur ekstraksi dan klasifikasi. Kedua tahap tersebut terdapat banyak metode yang dapat digunakan untuk membangun sistem pengenalan wajah, juga masing-masing memiliki kelebihan dan kekurangan dari berbagai aspek. Salah satunya tahap fitur ekstraksi yang memiliki banyak metode dengan kelebihan dan kekurangan, ada yang mengutamakan waktu komputasi yang cepat dengan mengorbankan tingkat akurasi, juga mengorbankan waktu komputasi untuk mencapai tingkat akurasi yang tinggi [8].

Adapun metode fitur ekstraksi yang saat ini diteliti menggunakan *Principal Component Analysis* (PCA) dan klasifikasinya menggunakan *Naïve Bayes*, kedua metode tersebut tentunya memiliki fungsi dan karakteristik kegunaan yang berbeda.

### 2.3 Citra Grayscale

Citra dengan skala keabuan memiliki nilai minimum (bernilai = 0) dan nilai maksimum. Banyaknya kemungkinan nilai minimum dan maksimum bergantung pada jumlah bit yang digunakan yang umumnya menggunakan 8-bit. Misalkan dengan skala keabuan 4 bit, maka jumlah kemungkinan nilai  $2^4 = 16$ , dan nilai maksimumnya  $2^4 - 1 = 15$ , sedangkan dengan skala keabuan 8-bit, jumlah kemungkinan nilai  $2^8 = 256$ , dan nilai maksimum  $2^8 - 1 = 255$  [9].

Secara digital *grayscale image* dapat direpresentasikan sebagai kumpulan piksel dengan warna merah, biru, dan hijau (*RGB-Layer*), lalu ketiga warna tersebut dijumlahkan kemudian dibagi tiga sehingga didapat nilai rata-rata. Nilai rata-rata inilah yang nantinya akan digunakan untuk memberikan warna pada piksel gambar sehingga warna menjadi *grayscale*. Secara digital, *grayscale image* dapat direpresentasikan sebagai array dua dimensi dengan tiap elemen dalam *array* menunjukkan intensitas (*greylevel*) dari *image* pada posisi koordinat yang bersesuaian. Citra yang direpresentasikan dalam 8-bit diatur intensitas cahayanya dengan nilai 0 menunjukkan nilai tergelap dan 255 dengan intensitas cahaya paling terang [10]. Berikut merupakan persamaan untuk mengkonversi citra RGB ke *grayscale*, sebagai berikut:

$$Grayscale = 0.299R + 0.587G + 0.114B \quad (2.1)$$

## 2.4 Principal Component Analysis (PCA)

*Principal Component Analysis* merupakan salah satu teknik multivariat yang paling populer, biasanya digunakan pada pengenalan pola dan pemrosesan sinyal. PCA juga merupakan salah satu cara paling mudah untuk mereduksi dimensi dengan melakukan transformasi ruang dimensi sebelumnya yang tinggi menjadi ruang dimensi rendah, namun dari hasil reduksi tersebut masih terdapat sisa informasi sebelumnya, sehingga metode PCA dapat dikatakan kurang optimal dalam pemisahan antar kelas dengan ini mempengaruhi tingkat akurasi. Adapun beberapa algoritma turunan dengan berdasarkan PCA seperti, *Eigenface*, *Probabilistic Eigenface*, dan *Two Dimensional PCA (2D-PCA)*. Pada penelitian ini akan menggunakan algoritma *Eigenface* sebagai algoritmanya [11].

*Eigenface* berasal dari kata “*Eigen*” berarti karakteristik dan “*wart*” berarti nilai, “*Eigenwert*” sendiri merupakan istilah kata yang berasal dari Bahasa Jerman. Metode *Eigenface* merupakan hasil dari transformasi citra-citra wajah menjadi set fitur karakteristik wajah dalam bentuk *eigenvector* dari matriks kovarian citra-citra wajah tersebut [8]. Untuk menghasilkan *eigenface* sekumpulan citra digital dari wajah manusia diambil dengan kondisi pencahayaan yang sama kemudian dinormalisasikan dan diproses pada resolusi yang sama (misal  $w \times h$ ), kemudian citra tersebut diperlakukan sebagai vektor dimensi  $w \times h$  yang dimana komponennya diambil dari nilai piksel citra.

Adapun perhitungan *Eigenface* secara matematis sebagai berikut [2]:

- a) Menyiapkan dan mengubah M jumlah citra di basis data dengan ukuran  $w \times h$   $112 \times 92$  piksel kedalam bentuk himpunan Y yang terdiri dari seluruh citra latihan  $(I_1, I_2, \dots, I_M)$ .

$$Y = (I_1, I_2, \dots, I_M). \quad (2.2)$$

- b) Mengambil nilai tengah atau *mean* ( $\psi$ )

$$\psi = \frac{1}{M} \sum_{N=1}^M \Gamma_i. \quad (2.3)$$

- c) Mencari selisih ( $\phi$ ) antara citra latih ( $\Gamma_1$ ) dengan nilai tengah ( $\psi$ )

$$\phi = \Gamma_1 - \Psi. \quad (2.4)$$

Hasil dari nilai *mean* digunakan untuk mencari seisih dari citra latih dan nilai rata-rata. Apabila hasil yang diperoleh kurang dari nol, maka dianggap nol.

- d) Menghitung nilai matriks kovarian (C)

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T. \\ L = A^T A \quad L = \Phi_m^T \Phi_n \quad (2.5)$$

Matriks kovarian diperoleh dari hasil  $\Phi_1, \Phi_2, \Phi_3$  dikalikan dengan hasil dari mengubah urutan (*transpose*)  $\Phi_1, \Phi_2, \Phi_3$ .

- e) Menghitung *eigenvalue* ( $\lambda$ ), *eigenvector* ( $v$ ) dan matriks kovarian (C).

$$C \times v_i = \lambda_i \times v_i. \quad (2.6)$$

- f) Setelah *eigenvector* ( $v$ ) diperoleh, maka *eigenface* ( $\mu$ ) dapat dicari.

$$\mu_i = \sum_{k=1}^M v_i \times \Phi_k, 1 = 1, \dots M. \quad (2.7)$$

Nilai  $\mu$  diperoleh dengan mengalikan nilai  $v$  dan  $\Phi$ .

- g) Normalisasi *eigenface* dengan menggunakan persamaan (2.8), dan simpan *eigenface* hasil normalisasi kedalam matriks  $\mu$ .

$$\mu_{ni} = \frac{\mu_i}{\mu_i, \mu_k^T} \quad (2.8)$$

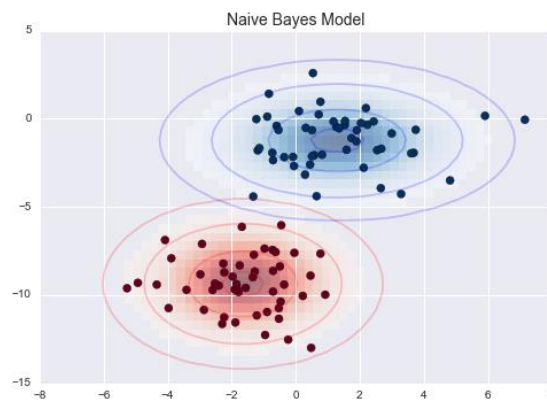
- h) Hitung bobot dengan mengalikan matriks selisih  $\Phi_k$  berukuran  $M \times wh$  matriks *transpose* dari  $\mu_{ni}$  yang berukuran  $wh \times M$ , menggunakan persamaan (2.9).

$$\omega = \Phi_k \times \mu_{nk}^T \quad (2.9)$$

Nilai  $\mu$  diperoleh dengan mengalikan nilai  $v$  dan  $\Phi$ .

## 2.5 Naïve Bayes Classifier (NBC)

Metode *Naïve Bayes* merupakan salah satu metode klasifikasi populer yang sering digunakan pada *Machine Learning* dengan probabilitas dan statistik sebagai perhitungannya. Metode ini pertama kali dikemukakan oleh Thomas Bayes untuk memprediksi probabilitas di masa depan berdasarkan pengalaman sebelumnya [12].



Gambar 2.1 Klasifikasi *Naive Bayes*

Fitur utama dari *Naïve Bayes Classifier* (NBC) Persamaan dari metode *Naïve Bayes* adalah asumsi yang sangat kuat atau naif dari independensi untuk setiap parameter, kondisi, atau peristiwa [13]. Klasifikasi ini dapat dilatih dengan sangat efektif dalam pembelajaran yang diawasi dan dapat juga digunakan pada situasi dunia nyata yang kompleks. Kelebihan dengan menggunakan *Naïve Bayes* ialah hanya sedikit menggunakan data pelatihan yang penting untuk karakterisasi dan klasifikasi. Dapat dilihat dengan persamaan 2.11.

$$P(C|x) = \frac{P(x|C) P(c)}{P(x)} \quad (2.10)$$

Dapat dijelaskan bahwa  $x$  adalah kelas yang belum diketahui,  $c$  adalah hipotesis data dengan suatu kelas spesifik,  $P(c|x)$  adalah probabilitas hipotesis berdasarkan kondisi,  $P(c)$  adalah hipotesis, probabilitas hipotesis,  $P(x|c)$  adalah probabilitas pada kondisi hipotesis, dan  $P(x)$  adalah probabilitas  $c$ . Pada *Naïve Bayes*

*Classifier* (NBC) terdapat beberapa macam *classifier* yang dapat digunakan berdasarkan kegunaan dan fungsinya, berikut penjelasannya.

1. *Gaussian Naïve Bayes Classifier*

Penggunaan *classifier* jenis ini digunakan pada data yang bersifat kontinu, pendekatan secara umumnya dengan mendistribusikan nilai-nilai sesuai dengan distribusi *gaussian*. Lalu, probabilitas bersyarat dapat dihitung menggunakan rata-rata dan perbedaan dari nilai-nilai [14]. Oleh karena itu untuk memperkirakan probabilitas kumpulan data kontinu, dengan perhitungannya sebagai berikut [3]:

$$P(x = x|C = c) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.11)$$

Keterangan:

$x$  = variabel  
 $c$  = kelas  
 $\mu$  = rata-rata  
 $\sigma$  = standar deviasi

2. *Multinomial Naïve Bayes Classifier*

Model *Multinomial Naïve Bayes* dirancang dan dikembangkan dari algoritma *Bayes* berguna untuk pengklasifikasian teks atau dokumen. Model ini memperhitungkan jumlah frekuensi dari setiap kata yang muncul pada dokumen tertentu. Berbeda dengan *Gaussian Naïve Bayes* pada pemilihan datanya, *Multinomial Naïve Bayes* cocok digunakan pada data yang diskrit sedangkan *Gaussian Naïve Bayes* cocok pada data kontinu [15]. Dengan perhitungannya sebagai berikut [16]:

$$P(c|\text{term dokumen } d) = P(c) \times P(t_1|c) \times \dots \times P(t_n|c) \quad (2.12)$$

Keterangan:

$P(c)$  = Probabilitas *prior* dari kelas  $c$   
 $t_n$  = Kata dokumen  $d$  ke- $n$   
 $P(c|\text{term dokumen } d)$  = Probabilitas suatu dokumen termasuk kelas  $c$

$P(t_n|c)$  = Probabilitas kata ke- $n$  dengan diketahui kelas  $c$

Mencari probabilitas *prior* dari kelas  $c$  dihitung dengan rumus sebagai berikut:

$$P(c) = \frac{N_c}{N} \quad (2.13)$$

Keterangan:

$N_c$  = Banyak kelas  $c$  pada semua dokumen

$N$  = Banyak seluruh dokumen

### 3. *Bernoulli Naïve Bayes Classifier*

Model algoritma *Bernoulli Naïve Bayes Classifier* ini biasanya digunakan untuk data yang diasumsikan mengambil nilai “Benar” atau “Salah” apabila muncul dalam dokumen. Model ini terlihat mirip seperti multinomial dalam proses klasifikasi, algoritma ini juga dapat dikatakan pendekatan yang populer untuk tugas klasifikasi teks tetapi berbeda dengan pendekatan multinomial dalam hal ini multinomial menggunakan perhitungan secara jumlah frekuensi yang keluar, sedangkan pendekatan Bernoulli hanya merancang apakah suatu istilah “Benar” atau “Tidak” dalam dokumen yang sedang dipertimbangkan [17]. Dengan perhitungannya sebagai berikut [18]:

$$P(p|n) \propto P(p) \prod_{1 \leq k \leq nd} P(t_k | p)(1 - P(t_{k'} | p)) \quad (2.14)$$

Pada penjelasan rumus diatas, dimana  $P(t_k | p)$  ialah representasi probabilitas bersyarat dari istilah  $t_k$  yang muncul dalam sebuah artikel berita atau dokumen polaritas  $p$  dan  $P(t_{k'} | p)$  representasi probabilitas tidak bersyarat dari istilah  $t_{k'}$  yang muncul dalam artikel berita atau dokumen. Kedua probabilitas bersyarat ini dapat diberikan sebagai berikut:

$$P(t_k | p) = \frac{\text{count}(t_k | p) + 1}{\text{count}(Np) + 2} \quad (2.15)$$

$$P(t_{k'} | p) = \frac{\text{count}(t_{k'} | p) + 1}{\text{count}(Np) + 2} \quad (2.16)$$

Dimana nilai  $\text{count}(t_k | p)$  yang artinya nilai perhitungan dalam setiap kemunculan suatu istilah dalam artikel berita berpolaritas  $p$  diaman untuk

berita tertentu, nilai yang kemunculannya dapat “1” atau “0” dan *count* ( $Np$ ) berarti jumlah artikel berita yang memiliki polaritas sebagai  $p$ .  $P(p)$  dapat di representasikan dengan perhitungan sebagai berikut:

$$P(p) = \frac{\text{Jumlah artikel berita polaritas } p}{\text{Total jumlah artikel berita}} \quad (2.17)$$

## 2.6 Python

Bahasa Python dikembangkan sekitar akhir tahun 1980-an, lalu dimulai implementasi pada akhir tahun 1989 oleh *Guido van Rossum* di Pusat Penelitian Wiskunde dan Informatika (*National Research Institute for Mathematics and Computer Science*) di Belanda. Python merupakan penerus Bahasa pemrograman ABC, yang terinspirasi dari SETL. Pada Februari 1991 *Van Rossum* menerbitkan kode Python ke *alt sources newsgroup*. Nama Python sendiri terinspirasi dari acara komedi televisi “Monty Python’s Flying Circus,” karena *Van Rossum* penggemar berat acara tersebut [19].

## 2.7 OpenCV

OpenCv merupakan sebuah *library* yang bersifat *open source* atau dapat digunakan secara gratis, OpenCv dikembangkan oleh intel untuk digunakan pada pengolahan citra dan *computer vision*. OpenCv dibuat menggunakan bahasa pemrograman C dan C++ serta dapat dijalankan pada sistem operasi Linux, Windows dan MacOS dan bahasa pemrograman Python, C, C++, dan Java. *Library* ini didesain untuk aplikasi *real-time*, juga memiliki fungsi untuk gambar/video.

## 2.8 Pengujian Kinerja Algoritma

Suatu algoritma dapat diukur efisiensinya berdasarkan seberapa baik akurasi dan jumlah waktu komputasi yang diperlukan untuk mengeksekusinya. Algoritma yang efisien ialah algoritma dengan hasil akurasi yang baik dengan waktu komputasi yang singkat. Berikut penjelasan mengenai perhitungan akurasi dan waktu komputasi sebagai berikut [20]:



### 1. Akurasi

Akurasi suatu ukuran ketepatan sistem dalam mengenali masukan yang akan diberikan sehingga menghasilkan keluaran yang benar. Secara sistematis dapat dituliskan sebagai berikut.

$$A = \frac{B}{n} \times 100\% \quad (2.18)$$

Keterangan:

$A$ : Akurasi

$B$ : Jumlah data benar

$n$ : Jumlah data keseluruhan

### 2. Waktu Komputasi

Waktu komputasi adalah waktu yang dibutuhkan oleh sistem untuk melakukan suatu proses. Pada sistem ini, waktu komputasi dihitung dengan menggunakan waktu selesai dikurangi dengan waktu mulai sehingga akan diperoleh waktu komputasi sistem.

$$W_k = W_s - W_m \quad (2.19)$$

Keterangan:

$W_k$ : Waktu Komputasi (s)

$W_s$ : Waktu Selesai (s)

$W_m$ : Waktu Mulai (s)

## 2.9 *Confusion Matrix*

*Confusion Matrix* merupakan tabel yang menyatakan klasifikasi jumlah data uji yang benar dan jumlah data uji yang salah. Berikut contoh *Confusion Matrix* yang ditunjukkan pada Tabel 2.1 [20].

Tabel 2.1 Model *Confusion Matrix*

	Nilai Aktual	
Nilai Prediksi	Positive	Negative
Positive	TP (True Positive)	FP (False Positive)
Negative	FP (False Negative)	TN (True Negative)

Keterangan [21]:

TP (*True Positive*) = Jumlah data positif yang terklasifikasi dengan benar oleh sistem.

TN (*True Negative*) = Jumlah data negatif yang terklasifikasi dengan benar oleh sistem.

FP (*False Positive*) = Jumlah data positif namun terklasifikasi salah oleh sistem.

FN (*False Negative*) = Jumlah data negatif namun terklasifikasi salah oleh sistem.

Rumus *Confusion Matrix* untuk menghitung *accuracy*, *precision*, dan *recall* seperti berikut.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.20)$$

$$precision = \frac{TP}{TP+FP} \quad (2.21)$$

$$recall = \frac{TP}{TP+FN} \quad (2.22)$$

## 2.10 Dataset Wajah

Dataset wajah yang digunakan pada penelitian ini yaitu dengan menggunakan dataset AT&T (ORL), dataset Georgia Tech, dataset Yale B Face, dan dataset Mandiri. Pada penjelasan dibawah ini mengenai dataset yang digunakan ialah sebagai berikut.

### 2.10.1 Dataset AT&T

Dataset AT&T merupakan sekumpulan basis data gambar yang di ambil pada April 1992 dan April 1994 di lab. Basis data ini digunakan dalam konteks proyek pengenalan wajah yang dilakukan dengan bekerjasama Bersama *Speech, Vision and Robotics Group* dari Departemen Teknik Univesitas Cambridge.

Pada *library* ini berisi 400 gambar pelatihan dari 40 kelas dan 200 sampel dari 40 subjek yang sama dengan format PGM berukuran 112 x 92 piksel, dengan 256 *grey level per-pixel*. Pengambilan gambar ini diambil dengan periode berbeda, pencahayaan berbeda, dan ekspresi muka berbeda seperti mata terbuka atau tertutup, senyum atau tidak senyum, dan menggunakan kacamata atau tidak [22]. Berikut sampel dataset gambar wajah AT&T berikut ini [23].



Gambar 2.2 Dataset AT&T.

### 2.10.2 Dataset Georgia Tech

Dataset *Georgia Tech* merupakan kumpulan basis data gambar wajah berisi 50 orang yang diambil sekitar dua atau tiga sesi dengan jangka waktu antara 06 Januari 1999 sampai dengan 15 November 1999 berlokasi di Pusat Sinyal dan Pengolahan Citra Institut Teknologi Georgia. Semua orang akan diwakili oleh 15 gambar wajah berformat JPEG dengan latarbelakang yang tidak beraturan diambil dalam resolusi 640 x 480 piksel. Dengan ukuran rata-rata seluruhnya ialah sekitar

150 x 150 piksel, lalu disimpan kedalam 50 direktori dengan nama  $s_1, s_2, \dots, s_{50}$ . Dalam setiap direktori terdapat 14 gambar wajah dengan penamaan 01.jpg, 02.jpg, ..., 14.jpg sesuai dengan orang dalam basis data [24]. Berikut sampel dataset gambar wajah *Georgia Tech* berikut ini [25].



Gambar 2.3 Dataset Georgia Tech.

### 2.10.3 Dataset Yale B Face

Dataset *Yale B Face* sebuah kumpulan basis data wajah yang dihasilkan oleh Universitas California San Diego di Laboratorium *Computer Vision*, dengan berisikan 5850 gambar wajah, tetapi pada penelitian ini hanya menggunakan 900 gambar wajah yang memiliki 65 variasi iluminasi dengan berukuran 192 x 168 piksel berformat PGM. Berikut sampel dataset gambar wajah *Yale B Face* berikut ini [26].



Gambar 2.4 Dataset Yale B Face

#### 2.10.4 Dataset Mandiri

Dataset mandiri merupakan kumpulan basis data gambar yang dihasilkan dari pengambilan gambar secara langsung oleh peneliti, dataset ini berukuran 112 x 92 piksel dengan warna asli RGB (*Red*, *Green*, dan *Blue*) pada keseluruhan gambar sebelum di *greyscale*. Adapun, dataset ini memiliki citra wajah yang didalamnya terdapat oklusi, pencahayaan dan latar belakang berbeda. Dataset mandiri dapat dilihat pada gambar 2.7 dibawah ini.



Gambar 2.5 Dataset Mandiri.