

## BAB II

### TINJAUAN PUSTAKA

#### A. Penelitian Terdahulu

1. Norhikmah & Aditya Dharma Satria S (2018) telah mengembangkan implementasi metode algoritma *eigenface* dan *euclidean distance* dalam sistem pengenalan wajah. dalam sistem ini digunakan variabel 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.
2. Muhammad Rizki Muliawan, Beni Irawan & Yulrio Brianorman (2016) telah mengembangkan implementasi pengenalan wajah dengan metode *eigenface* pada sistem absensi. Pemrosesan pengenalan wajah pada sistem absensi dengan menggunakan metode *eigenface* pada *opencv* dapat dilakukan dengan cara input data pengguna dan data wajah beserta *password* dari masing-masing pengguna. Selanjutnya scan untuk mengetahui apakah wajah sudah sesuai dengan *database*, lalu masukkan *password*, maka proses absensi berhasil. Pemrosesan pengenalan wajah dengan menggunakan metode *eigenface* pada *opencv* ini dikatakan sensitif, karena bergantung pada intensitas cahaya, jarak, dan sudut pandang wajah. Telah berhasil diterapkan algoritma *eigenface* untuk mendeteksi wajah dengan bantuan *opencv*, sehingga dapat diimplementasikan ke dalam sistem.

Pada tabel dibawah ini merupakan penelitian terdahulu yang telah dilakukan oleh beberapa penulis mengenai algoritma *eigenface* diantaranya sebagai berikut:

**Tabel 2.1 Penelitian Terdahulu**

No.	Nama	Judul	Metode	Hasil
1.	(Effendi.,dkk 2018)	<i>Real Time Face Recognition using Eigenface and Viola-Jones Face Detector</i>	<i>Eigenface</i>	Dari testings, ditemukan bahwa aplikasi ini dapat mengenali gambar wajah dengan akurasi setinggi 90% dan serendah 6,67%. Solusi ini dapat digunakan sebagai alternatif untuk kehadiran real time dengan intensitas cahaya 170 Lux, webcam resolusi 320 x 240 pixel, dan subjek berdiri 1 meter sementara tidak mengenakan kacamata. Waktu pengenalan rata-rata adalah 0,18125 ms.
2.	(Norhikmah, 2018)	Metode Algoritma <i>Eigenface</i> Dan <i>Euclidean Distance</i> Dalam Sistem Pengenalan Wajah	Metode Algoritma <i>Eigenface</i> Dan <i>Euclidean Distance</i>	Variabel hasil pengenalan 2 image wajah dengan menggunakan metode <i>eigence</i> yang selanjutnya dengan menggunakan <i>euclidean distance</i> , maka hasil perbandingan kecocokan wajah yang diperoleh, dengan pertama memiliki rentang kecocokan 1,73 sedangkan image kedua sebesar 2,82.
3.	(Muliawan, Irawan, & Brianorman, 2015)	Implementasi Pengenalan Wajah Dengan Metode <i>Eigenface</i> Pada Sistem Absensi.	<i>Eigenface</i> pada <i>opencv</i>	Pemrosesan pengenalan wajah dengan menggunakan metode <i>eigenface</i> pada <i>opencv</i> ini dikatakan sensitif, karena bergantung pada intensitas cahaya, jarak, dan sudut pandang wajah. Telah berhasil diterapkan algoritma <i>eigenface</i> untuk mendeteksi wajah dengan bantuan <i>opencv</i> , sehingga dapat diimplementasikan ke dalam sistem.
4.	(Utomo & Mulyono, 2019)	<i>Performance Analysis of Face Recognition using Eigenface Approach</i>	<i>eigenface</i>	Penelitian ini bertujuan untuk menguji kinerja PT algoritma <i>eigenface</i> untuk mengenali gambar wajah. Di dalam Penelitian tiga kali, pengujian didasarkan pada database publik gambar wajah yang berbeda. Berdasarkan semua percobaan yang ada telah dilakukan, dapat disimpulkan bahwa <i>eigenface</i>

				Algoritma mampu mengenali gambar wajah dengan benar di bawah kondisi normal.
5.	(Azis, Nasrun, Setianingsih, & Murti, 2018)	<i>Face Recognition In Night Day Using Method Eigenface</i>	<i>eigenface</i>	Dari dua gambar ini, juga dapat disimpulkan bahwa penambahan data latihan malam dapat meningkatkan akurasi eigenface, seperti yang terlihat dari nilai Max dengan semua akurasi di atas 60, tidak seperti tanpa data latihan malam hanya 60. Peningkatan gambar yang memiliki akurasi tertinggi dengan data latihan malam adalah LT, dan CL tanpa data latihan malam.
6.	(Husein, 2018)	<i>Mobile Face Recognition Application Using Eigen Face Approaches for Android</i>	<i>eigenface</i>	sistem pengenalan wajah seluler manfaat dari portabilitas ponsel dan menawarkan antarmuka yang disederhanakan yang bermanfaat dari sistem yang mudah digunakan orang-orang. Juga pencahayaan harus di bawah kontrol, karena fakta bahwa dalam kasus di mana ada bayangan di wajah, itu akan membuat mendeteksi wajah prosedur yang sulit.

## **B. Landasan Teori**

### **1. Fintech**

*Financial technology (fintech)* merupakan inovasi layanan keuangan dengan menggunakan teknologi agar masyarakat dapat dengan mudah mengakses produk dan layanan keuangan serta melemahkan *barrier to entry* (BI, 2016). Perkembangan *fintech* ini mengubah pola model bisnis keuangan saat ini dimana melemahnya *barrier to entry* memberikan peran bagi *Fintech* dalam memunculkan perilaku *unregulated* yang menjalankan model bisnis layaknya perusahaan atau institusi *regulated* (Bank Indoensia, 2016). Perkembangan *fintech* di dunia sudah dimulai sejak tahun 1800-an dengan munculnya *telegraf* dan semakin berkembang pada tahun-tahun selanjutnya khususnya pada era digital saat ini.

Sementara itu, *fintech* di Indonesia mulai berkembang dari tahun 2006 dengan jumlah perusahaan pada awalnya hanya empat perusahaan saja dan berkembang menjadi 16 perusahaan di tahun 2007. Perkembangan signifikan terjadi pada tahun 2015 hingga 2016 dimana jumlah perusahaan yang menjalankan model bisnis *fintech* sekitar 165 perusahaan (Rusydia, 2019).

### **2. Fintech Syariah**

*Fintech Syariah* menurut Mukhlisin (2017) adalah kombinasi, inovasi yang ada dalam bidang keuangan dan teknologi yang memudahkan proses transaksi dan investasi berdasarkan nilai-nilai syariah. Ia berpendapat, walaupun *fintech* ini merupakan terobosan baru tetapi mengalami perkembangan yang pesat. Islam merupakan agama yang komprehensif sehingga dalam bidang keuangan ini harus memiliki aturan yang sesuai dengan prinsipnya sesuai syariah (Yarli, 2018).

### **3. Pengenalan Pola**

Pola adalah suatu bentuk dimana masing-masing pola memiliki ciri-cirinya. Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola yang lainnya. Ciri yang baik adalah ciri yang memiliki daya pembeda yang tinggi, sehingga pengelompokan pola berdasarkan ciri yang dimiliki dapat dilakukan dengan keakuratan yang tinggi (Rizki, Irawan, & Brianorman, 2015).

#### 4. Pengenalan Wajah

Pengenalan wajah adalah salah satu teknologi biometrik yang telah banyak diaplikasikan dalam sistem keamanan selain pengenalan retina mata, pengenalan sidik jari dan iris mata. Dalam aplikasinya sendiri pengenalan wajah menggunakan sebuah kamera untuk menangkap wajah seseorang kemudian dibandingkan dengan wajah yang sebelumnya telah disimpan di dalam database tertentu (Rizki et al., 2015).

Pengenalan wajah melibatkan banyak variabel, misalnya citra sumber, citra hasil pengolahan citra, citra hasil ekstraksi dan data profil seseorang. Dibutuhkan juga alat pengindra berupa sensor kamera dan metode untuk menentukan apakah citra yang ditangkap oleh *webcam* tergolong wajah manusia atau bukan, sekaligus untuk menentukan informasi profil yang sesuai dengan citra wajah yang dimaksud.

#### 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 diencode dan dibandingkan dengan hasil *decode* yang sebelumnya telah dilakukan (Norhikmah, 2018b).

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.

Algoritma pengenalan wajah dimulai dengan membuat matriks kolom dari wajah yang dimasukkan ke dalam database. Ratarata *vector* citra (*mean*) dari matriks kolom dihitung dengan cara membaginya dengan jumlah banyaknya citra yang disimpan didalam database. Algoritma *eigenface* secara keseluruhan cukup sederhana. Image matriks ( $\Gamma$ ) direpresentasikan ke dalam sebuah himpunan matriks ( $\Gamma_1, \Gamma_2, \dots, \Gamma_M$ ). Cari nilai rata-rata ( $\Psi$ ) dan gunakan untuk mengekstraksi *eigenvector* ( $v$ ) dan *eigenvalue* ( $\lambda$ ) dari himpunan matriks. Gunakan nilai *eigenvector* untuk mendapatkan nilai *eigenface* dari *image*. Apabila ada sebuah image baru atau test face ( $\Gamma_{new}$ ) yang ingin dikenali, proses yang sama juga diberlakukan untuk *image* ( $\Gamma_{new}$ ), untuk mengekstraksi *eigenvector* ( $v$ ) dan *eigenvalue* ( $\lambda$ ), kemudian cari nilai *eigen-face* dari *image test face* ( $\Gamma_{new}$ ). Setelah itu barulah *image* baru ( $\Gamma_{new}$ ) memasuki tahapan pengenalan dengan menggunakan metode *euclidean distance*. Tahap perhitungan selengkapnya dapat dilihat seperti berikut ini.

a. Tahapan Perhitungan *Eigenface*

- 1) Langkah pertama adalah menyiapkan data dengan membuat suatu himpunan  $S$  yang terdiri dari seluruh *training image*, ( $\Gamma_1, \Gamma_2, \dots, \Gamma_M$ )

$$S = (\Gamma_1, \Gamma_2, \dots, \Gamma_M)$$

- 2) Langkah kedua adalah ambil nilai tengah atau mean ( $\Psi$ )

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

- 3) Langkah ketiga kemudian cari selisih ( $\Phi$ ) antara nilai *training image* ( $\Gamma_i$ ) dengan nilai tengah ( $\Psi$ )

$$\phi_i = \Gamma_i - \Psi$$

- 4) Langkah keempat adalah menghitung nilai matriks kovarian ( $C$ )

$$C = \frac{1}{M} \sum_{n=1}^M \phi_n \phi_n^T = A A^T$$

$$L = A^T A \quad L = \Theta_m^T \Theta_n$$

- 5) Langkah kelima menghitung *eigenvalue* ( $\lambda$ ) dan *eigenvector* ( $v$ ) dari matriks kovarian ( $C$ )

$$C \times v_i = \lambda_i \times v_i$$

- 6) Langkah keenam, setelah *eigenvector* ( $v$ ) diperoleh, maka *eigenface* ( $\mu$ ) dapat dicari dengan:

$$\mu_i = \frac{1}{M} \sum_{k=1}^M v_i^k \phi_k$$

$$i=1, \dots, M$$

#### b. Tahapan Pengenalan wajah

1. Sebuah image wajah baru atau *test face* ( $\Gamma_{new}$ ) akan dicoba untuk dikenali, pertama terapkan cara pada tahapan pertama perhitungan *eigenface* untuk mendapatkan nilai eigen dari image tersebut.

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

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

- Gunakan metode euclidean distance untuk mencari jarak (distance) terpendek antara nilai eigen dari training image dalam database dengan nilai eigen dari image testface.

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

## 6. OpenCv

OpenCv adalah *library* yang didalam-nya terdapat berbagai fungsi yang diper-gunakan dalam *computer vision*. *OpenCv* ditulis dengan bahasa C dan C++ dan dapat berjalan di *Linux*, *Windows* dan *MacOS*.)

OpenCv ini adalah *library* yang sudah sangat terkenal pada pengolahan citra *computer vision*. OpenCv didesain untuk aplikasi *real-time*, memiliki fungsi-fungsi yang baik untuk *image/video*. *OpenCv* sendiri terdiri dari 5 *library*, yaitu :

- CV : untuk algoritma *image processing* dan *vision*.
- MLL : untuk *machine learning library*.
- HighGUI : untuk GUI, *Image* dan Video I/O.
- CvAux : Fungsi penolong *OpenCv*
- CXCORE : untuk struktur data, *support XML* dan fungsi-fungsi grafis.

## 7. Tensorflow

*TensorFlow* adalah *framework machine learning* yang bekerja dalam skala besar dan dalam *environment* yang *heterogeneous*. *TensorFlow* digunakan untuk melakukan eksperimen model *deep learning*, melatih model pada *dataset* yang berukuran besar, dan membuatnya layak diproduksi.

Selain itu *TensorFlow* juga mendukung *training* dan *inference* berskala besar dengan menggunakan ratusan server yang menggunakan

*Graphic Processing Unit* (GPU) untuk *training* secara efisien. Penelitian ini menggunakan *TensorFlow* sebagai *backend framework* untuk Keras.

#### 8. PCA (*Principal Component Analysis*)

*Principal Component Analysis* (PCA) adalah sebuah cara untuk mengidentifikasi pola pada data dan kemudian mengekspresikan data tersebut ke bentuk yang lain untuk menunjukkan perbedaan dan persamaan antar pola. Tujuan dari PCA adalah untuk mereduksi dimensi yang besar dari ruang data (*observed variables*) menjadi dimensi yang lebih kecil dari ruang fitur (*independent variables*), yang dibutuhkan untuk mendeskripsikan data lebih sederhana. Ruang fitur adalah ciri yang digunakan sebagai kriteria dalam pengklasifikasian.

*Eigenface* adalah salah satu algoritma pengenalan wajah yang berdasarkan pada PCA. Dalam fase pengenalan, *eigenface* mereduksi dimensi dari input gambar dengan memproyeksikannya ke dalam subruang yang ditemukan selama pelatihan. Memproyeksikan ke dalam subruang berarti menemukan titik terdekat dari gambar pelatihan. Subruang adalah dimensi terendah menggambarkan bahwa *eigenface* ditemukan selama fase pelatihan. Algoritma *Eigenface* secara keseluruhan cukup sederhana.

#### 9. Android

Android merupakan sistem operasi yang dikembangkan untuk perangkat *mobile* berbasis Linux. Pada awalnya sistem operasi ini dikembangkan oleh Android Inc. yang kemudian dibeli oleh *Google* pada tahun 2005.

Dalam usaha mengembangkan *Android*, pada tahun 2007 dibentuklah *Open Handset Alliance* (OHA), sebuah konsorsium dari beberapa perusahaan, yaitu *Texas Instruments*, *Broadcom Corporation*, *Google*, *HTC*, *Intel*, *LG*, *Marvell Technology Group*, *Motorola*, *Nvidia*, *Qualcomm*, *Samsung Electronics*, *Sprint Nextel*, dan *T-Mobile* dengan tujuan untuk mengembangkan standar terbuka untuk perangkat *mobile*. Pada tanggal 9 Desember 2008, diumumkan bahwa 14 orang anggota

baru akan bergabung dengan proyek *Android*, termasuk *PacketVideo*, *ARM Holdings*, *Atheros Communications*, *Asustek Computer Inc*, *Garmin Ltd*, *Softbank*, *Sony Ericsson*, *Toshiba Corp*, dan *Vodafone Group Plc*(Maiyana, 2018).

#### 10. **Android Studio**

Android studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi *Android* dan bersifat *open source* atau gratis. Peluncuran Android Studio ini diumumkan oleh *Google* pada 16 mei 2013 pada *event Google I/O Conference* untuk tahun 2013. Sejak saat itu, Android Studio menggantikan *Eclipse* sebagai IDE resmi untuk mengembangkan aplikasi Android.

Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan *Eclipse* disertai dengan ADT *plugin* (*Android Development Tools*). Android studio memiliki fitur:

- a. Proyek berbasis pada *Gradle Build*
- b. Refactory dan pembenahan *bug* yang cepat
- c. Tools baru yang bernama "*Lint*" diklaim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- d. Mendukung *Proguard* And *App-signing* untuk keamanan.
- e. Memiliki GUI aplikasi android lebih mudah.
- f. Didukung oleh *Google Cloud Platfrom* untuk setiap aplikasi yang dikembangkan (Juansyah, 2015).

#### 11. **Xampp**

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache, HTTP Server, MySQL *database*, dan penerjemah bahasa yang ditulis dengan Bahasa pemrograman PHP dan *Perl* (Syifani & Dores, 2018)

#### 12. **Sublime text**

*Sublime Text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai *platform operating system* dengan menggunakan teknologi Phyton API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah fleksibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages* (Syifani & Dores, 2018).

### 13. **MySql**

Menurut *Kustiyahningsih (2011:145)*, “MySQL adalah sebuah basis data yang mengandung satu atau jumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel”.

Menurut *Wahana Komputer (2010:21)*, MySQL adalah *database server open source* yang cukup populer keberadaannya. Dengan berbagai keunggulan yang dimiliki, membuat *software database* ini banyak digunakan oleh praktisi untuk membangun suatu project. Adanya fasilitas API (*Application Programming Interface*) yang dimiliki oleh MySQL, memungkinkan bermacam-macam aplikasi komputer yang ditulis dengan berbagai bahasa pemrograman dapat mengakses basis data MySQL.

Tipe data MySQL adalah data yang terdapat dalam sebuah tabel berupa *field –field* yang berisi nilai dari data tersebut. Nilai data dalam *field* memiliki tipe sendiri– sendiri”.(Firman et al., 2016)

### 14. **PHP**

*Programming Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan web dan dapat ditanamkan pada sebuah skrip HTML. (Firman dkk., 2016).