

## BAB II TINJAUAN PUSTAKA

### 2.1 Pengolahan Citra

Pengolahan citra adalah salah satu cabang ilmu informatika yang berfokus pada penggunaan teknik tertentu untuk mengubah citra menjadi citra lain. Istilah citra mengacu pada informasi yang dikomunikasikan secara visual dan merupakan representasi dari objek dunia nyata kedalam media dua dimensi, sedangkan istilah pengolahan citra mengacu pada tindakan mengolah informasi menjadi gambar dan membuat gambar sebagai keluaran.

Pengolahan citra digital melibatkan cara pandang visual yang dimiliki manusia, citra digital merupakan gambar yang ditangkap melalui perangkat digital, citra ditangkap melalui perangkat digital seperti kamera digital yang dapat merepresentasikannya kedalam dua dimensi melalui proses *image acquisition* atau akuisisi citra. Citra yang ditangkap oleh kamera seringkali mengalami beberapa gangguan, seperti kamera tidak fokus, adanya kabut yang menghalangi objek yang sedang ditangkap, muncul bintik-bintik akibat prosedur pengambilan yang tidak sempurna, lensa kamera yang kotor, objek yang buram karena kontras citra yang rendah, dll.

Oleh sebab itu, pengolahan citra digital sangat diperlukan untuk proses perbaikan kualitas citra. Setiap gangguan pada citra dinamakan *noise* atau derau yang artinya warnanya menjadi kabur sehingga sulit diinterpretasikan oleh manusia. Berdasarkan tujuan transformasinya, pengolahan citra terbagi menjadi *image enhancement* atau peningkatan kualitas citra dan *image restoration* atau pemulihan citra. *Image enhancement* cenderung memperhatikan perbaikan kualitas citra yang mengalami penurunan kualitas selama pembentukan citra atau justru memberi efek berlebih pada citra yang sudah ada, sementara *image restoration* adalah pada perbaikan citra yang mengalami kerusakan, baik selama proses digitalisasi maupun cacat akibat usia, jamur, goresan, pelabelan teks pada citra yang dilakukan baik sengaja maupun tidak sengaja. (Hasugian & Zufria, 2018)

## 2.2 Deteksi Wajah

Deteksi wajah merupakan fase pertama dalam semua prosedur pengenalan wajah yang berdampak signifikan pada kegunaan dan kinerja sistem pengenalan wajah. Terlepas dari memakai aksesoris, usia, atau ekspresi, deteksi wajah dapat mengenali dan menemukan wajah dalam sebuah gambar asalkan posisi wajah harus menghadap kamera, jika tidak menghadap kamera maka tidak dapat mendeteksi wajah tersebut. Pendeteksian wajah dapat dilakukan dengan memanfaatkan *library* yang bersifat *open source* yaitu *library OpenCV* yang awalnya hanya menggunakan bahasa pemrograman *C/C++*, namun sekarang telah dikembangkan juga ke *python, java dan matlab*. (Asmara et al., 2018)

## 2.3 Pengenalan Wajah

Pengenalan wajah atau *face recognition* adalah teknologi komputer yang memungkinkan untuk mengidentifikasi atau memverifikasi wajah seseorang melalui sebuah gambar digital, caranya adalah dengan mencocokkan tekstur lekuk wajah dengan data wajah yang tersimpan di *database*. Misalnya mencocokkan bentuk mata, bibir, hidung, dagu dan sebagainya. Dalam proses mengenali wajah, otak dan pikiran berusaha memahami dan mengartikan wajah yang ada di depannya, terutama wajah manusia. Proporsi dan ekspresi wajah manusia dinilai penting untuk identifikasi awal mengenai kecenderungan emosional, kualitas kesehatan, ataupun beberapa informasi sosial. (Suroso & Ermaya, 2018)

Pengenalan wajah menjadi salah hal penting karena wajah sebagai identitas utama yang dimiliki manusia. Pengenalan wajah dalam sistem dapat “dikenali” maupun “tidak dikenali” karena pada hakikatnya setiap manusia mempunyai struktur wajah yang berbeda-beda dan tidak ada satupun di dunia ini memiliki wajah yang sama, selain itu posisi wajah saat melakukan teknologi pengenalan wajah dengan *database* wajah yang ada tidak sesuai maka ketika mengukur tingkat akurasi bisa saja tidak relevan. Sistem pengenalan wajah dapat mencegah kecurangan oleh karyawan saat melakukan proses presensi karena *output* data dari *database* hanya mengenali sesuai data citra wajah yang ada dalam *database* tersebut.

Manfaat teknologi pengenalan wajah adalah dapat mengidentifikasi secara akurat bahkan ketika wajah seseorang sedang mengalami perubahan, seperti kumis atau janggut yang tumbuh. Meskipun demikian, tidak jarang menghadapi masalah dengan proses pengenalan wajah, seperti ketidakmampuan sistem untuk membedakan antara wajah yang kembar identik dan mirip dalam *data training*. Selain itu, terdapat tantangan dengan pengenalan wajah karena tidak cukup foto atau postur gambar untuk digunakan sebagai data *training*, yang membuat sistem kurang efektif dalam mengenali wajah. (Amalia, 2022)

#### 2.4 Sistem Presensi

Sistem presensi adalah sebuah kegiatan pengambilan data yang digunakan untuk mencatat jumlah kehadiran dalam suatu kegiatan. Presensi tersebut kemudian dapat digunakan untuk memberikan informasi untuk menilai keberhasilan proyek tertentu atau untuk mencari informasi dari mereka yang telah menjalankan proyek tersebut. Presensi tersebut kemudian dapat diproses untuk memberikan informasi untuk mengukur kesuksesan suatu kegiatan atau mencari informasi dari yang telah mengikuti kegiatan tersebut.

Berdasarkan hal tersebut, tujuan dari presensi adalah untuk mempermudah suatu instansi dalam mengetahui berapa banyak karyawan yang hadir maupun tidak hadir. Meskipun ada beberapa cara untuk menerapkan presensi, tetapi kebanyakan diantaranya adalah dengan menggunakan daftar presensi yang dicetak di kertas dan karyawan melakukan paraf untuk menandakan bahwa karyawan yang bersangkutan telah datang dan bekerja sesuai dengan tugas masing-masing. Maka penggunaan daftar presensi tersebut menyebabkan adanya ketidakvalidan sehingga karyawan tertentu melakukan penitipan presensi dan meminta rekan yang dikenal untuk mewakili paraf presensi dan tidak hadir untuk kerja sehingga merugikan instansi tersebut.

Pengumpulan data presensi secara manual memiliki berbagai kelemahan, antara lain salah *input* data, data yang hilang, data yang rusak, kecurangan karena datang terlambat tapi masih mengisi presensi, pendataan presensi yang lama

sehingga mengurangi efisiensi dan efektivitas dalam rekapitulasi data kehadiran. (Thiosdor et al., 2021)

## 2.5 Algoritma Eigenface

Algoritma *Eigenface* adalah salah satu algoritma pengenalan wajah yang berdasarkan pada *Principle Component Analysis* (PCA), dimana *eigenface* akan dibantu oleh PCA dalam pengambilan ciri-ciri khusus pada wajah manusia dengan tujuan untuk mengambil citra dari wajah itu sendiri yang pada nantinya citra itu akan membedakan wajah satu dengan wajah lainnya. Algoritma *Eigenface* dikembangkan di *Massachusetts Institute of Technology* (MIT) yang merupakan salah satu universitas teknologi terbaik didunia.

Algoritma *Eigenface* secara keseluruhan cukup sederhana, *training image* direpresentasikan dalam sebuah *eigenvector* atau gabungan *vector* dan digabung bersama-sama menjadi sebuah matriks tunggal. *Eigenvector* kemudian diekstraksi dan disimpan dalam *file temporary* atau *database*. *Training image* kemudian diproyeksikan dalam *feature space*, dinamai *face space* atau ruang *eigen* yang ditentukan oleh *eigenvector*.

*Eigen* berasal dari Bahasa Jerman yaitu asli atau karakteristik, *eigenvalue* atau nilai *eigen* dan *eigenvector* dapat lebih dipahami jika ditinjau dalam bentuk persoalan fisik. Jika sebuah matriks dikali dengan *vector*  $x$  dan memiliki hasil yang sama dengan sebuah skalar yang dikali dengan *vector* yang sama, maka nilai skalar tersebut disebut sebagai *eigenvalue* dan  $x$  disebut sebagai *eigenvector* yang berkoresponden dengan *eigenvalue*, maka dapat dikatakan terdapat sebuah matriks yang menggambarkan bentuk deformasi tersebut. (Suroso & Ermaya, 2018)

$$A\mathbf{x} = \lambda\mathbf{x} \dots \dots \dots (1)$$

Keterangan:

A: Matriks yang ditemukan *eigenvector* dan *eigenvalue*,

$\lambda$ : *Eigenvalue*,

$x$ : *Eigenvector*.

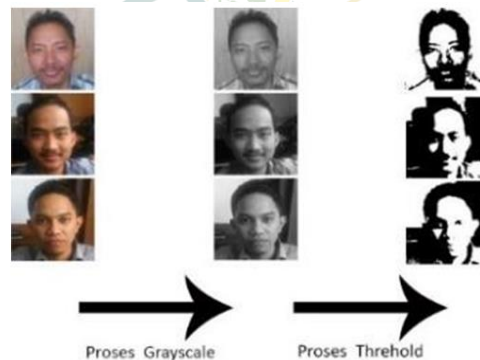
Algoritma *Eigenface* bisa dikatakan sebagai salah satu algoritma yang cukup

populer dan sering digunakan dalam menyelesaikan masalah pengenalan wajah dikarenakan implementasi formula yang digunakan algoritma *eigenface* lebih mudah dan waktu eksekusi yang lebih cepat dibandingkan algoritma pengenalan wajah lainnya.

Berikut adalah cara mencari nilai sederhana *eigenface*:

### 1. Proses *Image Test*

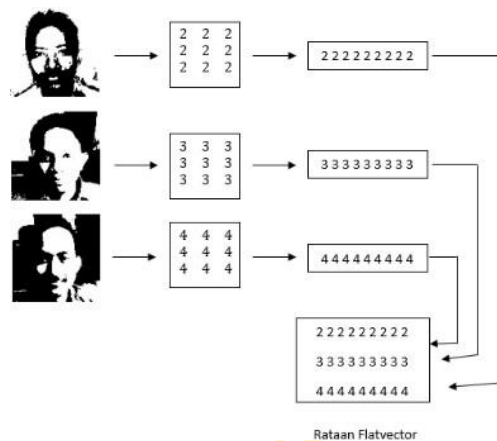
Langkah pertama setelah didapatkan citra wajah hasil tangkap gambar, citra wajah dirubah ke dari bentuk RGB ke dalam bentuk *grayscale*, setelah didapatkan citra keabuan ubah menjadi citra hitam putih dengan melakukan *treshold* agar kompleksitas citra lebih sederhana.



**Gambar 2. 1** Proses Penyederhanaan Citra Wajah

### 2. Penyusunan *Flatvector*

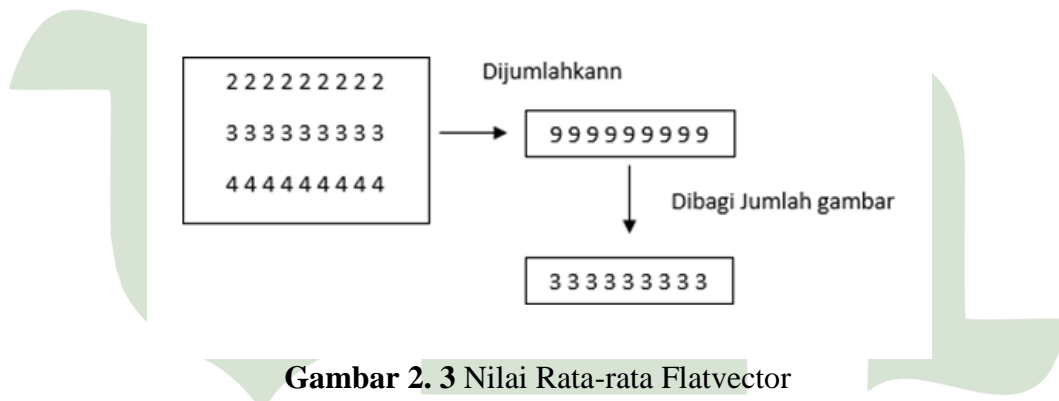
Selanjutnya adalah menyusun seluruh *training image* menjadi suatu matriks tunggal. Misalnya gambar yang disimpan berukuran  $H \times W$  piksel dan jumlahnya  $N$  buah, maka akan dimiliki *vector* ciri dengan dimensi  $N \times (W \times H)$ . Misalnya dalam *training image* terdapat 3 gambar dengan ukuran  $3 \times 4$  piksel maka kita akan mempunyai *eigenvector* ukuran  $3 \times 9$ . Ilustrasinya sebagai berikut:



**Gambar 2. 2** Penyusunan Flatvector

### 3. Perhitungan Rataan *Flatvector*

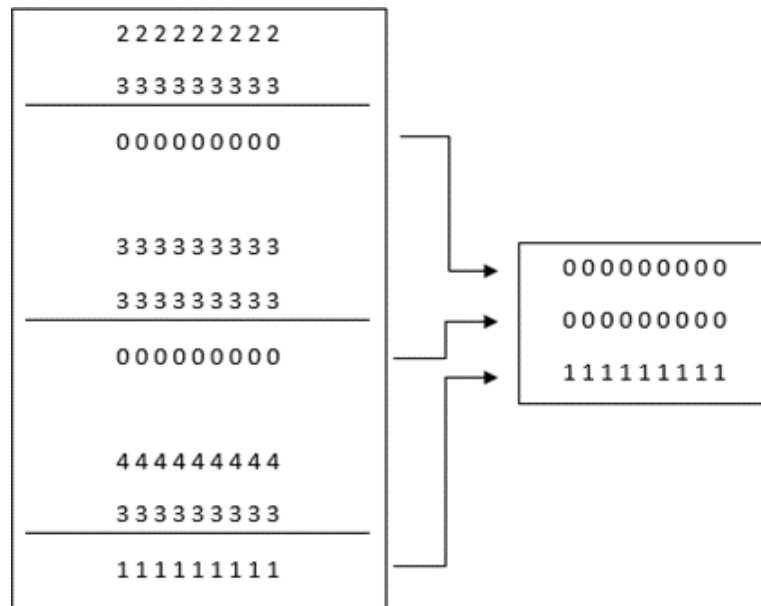
Dari *flatvector* yang diperoleh, jumlah seluruh barisannya sehingga diperoleh matriks berukuran  $1 \times (W \times H)$ . Setelah itu bagi matriks tersebut dengan jumlah gambar  $N$  untuk mendapatkan nilai rata-rata *flatvector*.



**Gambar 2. 3** Nilai Rata-rata Flatvector

### 4. Proyeksi Gambar Ke *Facespace* Untuk Mendapatkan Nilai *Eigenface*

Gambar diproyeksi ke *facespace* dengan mengkalikan dibasis *eigenface* proyeksi *vector* wajah akan dibandingkan dengan *vector* yang sesuai dengan menggunakan nilai rata-rata *vector* ciri, akan dihitung *eigenface* untuk matriks *vector* ciri dengan nilai rata-rata *vector* ciri, jika didapatkan nilai dibawah nol, ganti nilainya dengan nol. (Fauzi et al., 2020)



**Gambar 2. 4** Perhitungan Eigenface

## 2.6 OpenCV

*Open Source Computer Vision Library* (OpenCV) adalah suatu perangkat lunak tambahan yaitu sebagai komponen *library* atau pustaka untuk menyimpan sekumpulan data elektronik yang tersimpan dan dikendalikan oleh suatu perangkat komputer yang dapat digunakan dalam pemrograman-pemrograman yang berbasis *C/C++*, *python*, *java* dan *matlab* yang bisa digunakan di *windows*, *linux*, *Mac OS*, *iOS* dan *android* yang ditujukan untuk pengembangan *computer vision* dan *machine learning*. (Simanjuntak & Imelda, 2019)

Pada penelitian ini, *OpenCV* bersama dengan *Python* dimanfaatkan untuk mengolah gambar sesuai dengan tujuan masing-masing yang melibatkan kamera yang menangkap gambar lalu diolah di komputer. *OpenCV* akan digunakan dengan bantuan *pycharm* dan *python* agar kamera dapat menangkap bentuk wajah manusia sesuai yang dibutuhkan. *OpenCV* akan dimasukkan kedalam *pycharm* yang nantinya akan memerlukan *python* untuk menjalankan program web yang akan berjalan, tanpa adanya *OpenCV* program yang dibuat tidak akan berjalan.



**Gambar 2. 5** Logo OpenCV

## 2.7 Pycharm

*Pycharm* adalah salah satu bahasa pemrograman dari beberapa bahasa pemrograman seperti *python*, *java*, *C*, *C++* dan lain-lain. *Python* terkoneksi dengan *Pycharm* yang pastinya merupakan bahasa pemrograman yang *freeware* atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya, *pycharm* dapat menampung perangkat lunak *python* sebagai sarana untuk menjalankan program berbasis web. *Python* dikenal sebagai salah satu bahasa pemrograman tingkat tinggi, *python* terkenal pada kalangan programmer karena penggunaannya yang lebih sederhana dari bahasa pemrograman lainnya. Selain itu, *python* memiliki struktur sintaks yang rapi dan mudah dipahami oleh programmer.

Bahasa pemrograman ini menjadi umum digunakan untuk kalangan programmer di seluruh dunia dalam pembuatan perangkat lunaknya. Perangkat lunak yang dibuat bertujuan untuk mempermudah kehidupan manusia. (Wiratmaja et al., 2021)



**Gambar 2. 6** Logo Pycharm



## 2.8 Python

*Python* merupakan salah satu dari bahasa pemrograman yang umum digunakan oleh programmer atau pembuat program dalam membuat sebuah program. *Python* memiliki karakteristik sintaks yang tidak terlalu rumit. Oleh karena itu, *python* adalah salah satu bahasa pemrograman tingkat tinggi yang mudah untuk digunakan. Saat menggunakan bahasa pemrograman *python* untuk menulis kode program, ada beberapa aturan yang harus diikuti. Hal ini untuk mengantisipasi terjadinya kesalahan atau masalah pada program yang dibuat. Aturan sintaks *python* yang pertama adalah dalam penulisan *statement* atau perintah, perintah ini berupa sebuah intruksi atau kalimat pernyataan yang akan dieksekusi oleh komputer. Sebagai contoh dari pernyataan bahasa *python* adalah:

```
print ("Hello World")
```

Kemudian aturan yang kedua adalah aturan penulisan *string* atau data dalam bentuk kumpulan beberapa karakter atau huruf. Dalam bahasa pemrograman *python*, *String* umumnya ditulis sebagai karakter dalam suatu program dan di lampirkan dalam tanda kutip. Programmer dapat menggunakan tanda kutip tunggal, kutip ganda, atau kutip tiga. Misalnya:

```
kegiatan = "menonton bersama"
penulis = 'jojo moyes'
judul_film = '''aku setelah kamu'''
```

Lalu ada aturan penulisan untuk variabel dan tipe data *Python*. Variabel adalah tempat Simpan data yang dimasukkan ke dalam program. Sedangkan tipe data adalah tipe datanya disimpan dalam sebuah variabel. Variabel dapat berubah, artinya nilainya dapat berubah.

Dalam bahasa pemrograman *Python*, variabel dapat dibuat dalam format berikut:

```
"nama_variabel" (tanpa tanda petik)
```

```
Contoh: variabel_ku = "masih kosong"
```

```
Untuk menampilkan isi: print variabel_ku
```

Aturan dalam penulisan variabel adalah yaitu nama variabel boleh diawali dengan menggunakan huruf atau garis bawah. Karakter selanjutnya dapat berupa huruf, garis bawah, atau angka. Lalu karakter pada nama variabel bersifat *case* –

*sensitive* dan nama variabel tidak boleh menggunakan kata kunci *python* seperti *if*, *while*, *for*, dan sebagainya. Untuk menghapus variabel dapat menggunakan perintah *del* (nama variabel). Jenis-jenis tipe data yang terdapat pada *python* yaitu tipe data angka yang terbagi menjadi dua yaitu *Integer* (bilangan bulat) dan *Float* (bilangan pecahan), tipe data teks yang terbagi menjadi dua yaitu *Char* atau karakter/huruf dan *String*, dan tipe data *Boolean* yang merupakan tipe data yang hanya memiliki nilai benar atau salah (1 atau 0). (Wiratmaja et al., 2021)

## 2.9 Database MySQL

*Database* atau basis data adalah sebuah sistem yang diciptakan untuk mengorganisasi, menyimpan dan menarik data dengan mudah. *Database* terdiri dari kumpulan data yang terorganisir untuk 1 atau lebih penggunaan, dalam bentuk digital. *Database digital* dikelola menggunakan *Database Management System* (DBMS), yang menyimpan isi *database*, mengizinkan pembuatan dan pemeliharaan data pencarian dan akses yang lain. Beberapa basis data yang ada saat ini adalah : *MySQL*, *SQL Server*, *Ms.Access*, *Oracle*, dan *PostgreSq*.

Adapun beberapa fungsi dari *database* adalah mempermudah identifikasi data dengan cara pengelompokkan data, salah satu contoh dengan pembuatan beberapa tabel yang berbeda-beda, meminimalisir suatu data ganda, mempermudah penggunaan *user* dalam berbagai hal, misalnya pada saat memasukkan data baru, penyimpanan secara digital, menjadi alternatif lain terkait masalah penyimpanan ruang dalam suatu aplikasi. (Ramadhan & Mukhaiyar, 2020)

*MySQL* adalah sebuah perangkat lunak yang terdapat di dalam sistem manajemen *database*, *Database Management System* (SQL) atau yang biasa disebut DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia dan menjadi salah satu jenis *database server* yang paling terkenal. *MySQL* menggunakan bahasa *SQL* untuk mengakses *database*. Lisensi *MySQL* adalah *FOSS License Exception* dan *General Public License* (GPL) atau versi komersial. *MySQL* tersedia untuk beberapa *platform*, diantaranya adalah untuk versi *windows* dan versi *linux*. (Oktaviani et al., 2019)

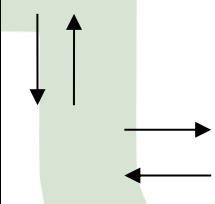


## 2.10 Flowchart

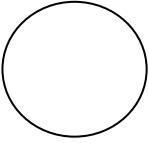
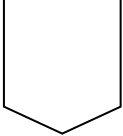
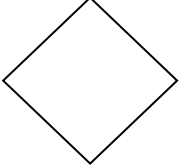

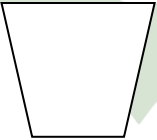


*Flowchart* atau diagram alir merupakan alat yang digunakan untuk menggambarkan sebuah algoritma, *flowchart* menggambarkan urutan logika dari suatu prosedur masalah, sehingga *flowchart* merupakan langkah-langkah penyelesaian masalah yang dituliskan dalam simbol-simbol tertentu.

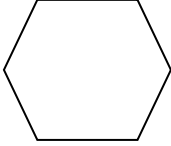

*Flowchart* merupakan alur pemikiran yang dituangkan kedalam bentuk gambar/symbol. Dengan menggunakan *flowchart*, maka seorang programmer dapat memberikan idenya secara tertulis sehingga dapat lebih mudah dipahami oleh pembaca, programmer lain, klien, atau tim kerja. (Mulyanto & Khasanah, 2018)

Adapun simbol-simbol diagram alir sebagai pedoman untuk langkah kerja dalam merancang aplikasi sebagai berikut:

**Tabel 2. 1** Simbol Flowchart

NO	Simbol Flowchart	Nama Flowchart	Fungsi
1.		<i>Flow / Connecting Line</i>	<i>Flow</i> berfungsi sebagai simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain.
2.		<i>Terminator</i>	<i>Terminator</i> berfungsi sebagai simbol yang menyatakan awal atau akhir suatu program.
3.		<i>Process</i>	<i>Process</i> berfungsi sebagai simbol yang menyatakan suatu proses pengolahan yang dilakukan oleh komputer.

4.		<i>On-Page / Connector Symbol</i>	<i>On-Page / Connector Symbol</i> berfungsi sebagai simbol keluar-masuk atau penyambungan proses dalam halaman yang sama.
5.		<i>Off-Page / Connector Symbol</i>	<i>Off-Page / Connector Symbol</i> berfungsi sebagai simbol keluar-masuk atau penyambungan proses dalam halaman yang berbeda.
6.		<i>Decision</i>	<i>Decision</i> berfungsi sebagai simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan yaitu ya atau tidak.
7.		<i>Input / Output</i>	<i>Input / Output</i> berfungsi sebagai simbol yang menyatakan proses <i>input</i> dan <i>output</i> suatu program.
8.		<i>Manual Operation</i>	<i>Manual Operation</i> berfungsi sebagai simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer.
9.		<i>Predefine Process</i>	<i>Predefine Process</i> berfungsi sebagai simbol untuk permulaan suatu <i>sub-program</i> .
10.		<i>Display</i>	<i>Display</i> berfungsi sebagai simbol yang menyatakan peralatan <i>output</i> yang digunakan.

11.		<i>Preparation</i>	<i>Preparation</i> berfungsi sebagai simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam <i>storage</i> .
12.		<i>Document</i>	<i>Document</i> berfungsi sebagai simbol yang menyatakan input berasal dari dokumen dalam bentuk fisik atau output yang dicetak.

*Flowchart* memiliki lima macam bentuk bagan alir yaitu sebagai berikut:

1. Diagram Alir Sistem (*System Flowchart*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem.
2. Diagram Alir Dokumen (*Document Flowchart*) merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Bagan ini disebut juga bagan alir formulir atau *paperwork flowchart*. Bagan alir ini menggunakan simbol-simbol yang sama dengan yang digunakan di dalam bagan alir sistem.
3. Diagram Alir Skematik (*Schematic Flowchart*) merupakan bagan alir yang mirip dengan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaannya adalah bagan alir skematik menggunakan simbol-simbol bagan alir sistem, juga menggunakan gambar-gambar komputer. Penggunaan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir. Penggunaan gambar-gambar ini memudahkan untuk dipahami, tetapi sulit dan lama menggambarinya.
4. Diagram Alir Program (*Program Flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir

program dibuat dari derivikasi bagan alir sistem. Bagan alir program dibuat dengan menggunakan simbol-simbol.

5. Diagram Alir Proses (*Process Flowchart*) merupakan bagan alir yang banyak digunakan diteknik industri. Bagan alir ini juga berguna bagi analis sistem untuk menggambarkan proses didalam prosedur.

Teknik pembuatan bagan alir dibagi menjadi dua bagian, yaitu:

1. *General Way*, Teknik pembuatan diagram alir dengan cara ini lazim digunakan dalam menyusun lagika suatu program, yang menggunakan pengulangan proses secara tidak langsung (*Non Direct Loop*).
2. *Interaction Way*, Teknik pembuatan diagram alir dengan cara ini biasanya digunakan untuk logika program yang cepat serta bentuk permasalahan yang kompleks, dimana proses secara langsung. (Oktavian et al., 2018)

### **2.11 PT. Buana Pilarjaya Mandiri**

Pada penelitian ini, objek yang menjadi informasi adalah mengenai presensi karyawan pada sebuah PT yang berbentuk digital, PT adalah kepanjangan dari Perseroan Terbatas yaitu suatu badan usaha yang menjalankan usaha yang dilindungi oleh hukum dengan modal yang terdiri dari modal saham yang bisa diperjualbelikan kepada pihak lain. PT tidak bisa didirikan oleh satu orang saja karena dibentuk berdasarkan kesepakatan bersama, maka bisa ditetapkan bahwa PT didirikan minimal oleh dua orang atau lebih yaitu diantaranya merupakan Direktur, Wakil Direkur dan Komisaris.

PT. Buana Pilarjaya Mandiri adalah badan usaha yang bergerak dibeberapa bidang seperti jasa pelaksana untuk konstruksi bangunan dan industri, jasa pelaksana untuk konstruksi gedung lainnya, jasa pelaksana untuk konstruksi saluran air, pelabuhan, dam, prasarana sumber daya air lainnya, jasa pelaksana untuk konstruksi jalan raya, jalan khusus, kereta api dan landas pacu bandara.

### **2.12 Penelitian Terdahulu**

Penelitian terdahulu sering disebut dengan pembanding bagi peneliti untuk mencari inspirasi baru pada penelitian berikutnya. Pada penelitian terdahulu, *eigenface* sudah banyak digunakan untuk mengembangkan aplikasi keamanan

dalam berbagai bidang. Berikut adalah penelitian terdahulu mengenai algoritma *eigenface*:

**Tabel 2. 2** Penelitian Terdahulu

NO	JUDUL PENELITIAN	TAHUN	ULASAN
1	Sistem Pengenalan Wajah Menggunakan Metode <i>Eigenface</i> Dengan <i>Visual Studio</i> (Sanusi et al., 2022)	2022	Dalam Jurnal ini, peneliti melakukan penelitian mengenai sistem pengenalan wajah menggunakan metode <i>eigenface</i> dengan <i>visual studio</i> , memakai 3 data citra untuk pengujian dan pengujian dilakukan dengan mengubah nilai <i>threshold</i> 100 sampai 1000 Berdasarkan hasil pengujian, sistem dapat mengenali gambar sesuai dengan label yang diberikan pada <i>database</i> dan tidak dapat mengenali gambar pada nama karena tidak sesuai dengan nama yang diberikan pada <i>database</i> , dengan memakai 3 data wajah dalam pengujian nilai <i>threshold</i> 300-1000, mendapatkan hasil akurasi sebesar 93,33% dalam mendeteksi pola wajah dengan benar.
2	Identifikasi Pengenalan Wajah Manusia Studi Kasus	2020	Dalam Jurnal ini, peneliti melakukan penelitian mengenai identifikasi pengenalan wajah

	<p>Pemakaian Aksesoris Topi Dengan Metode <i>Eigenface</i> (Fauzi et al., 2020)</p>		<p>manusia menggunakan aksesoris topi dengan metode <i>eigenface</i>, menggunakan data wajah manusia tanpa menggunakan aksesoris apapun sebagai acuan informasi sebuah datanya kemudian dilakukan penelitian menggunakan aksesoris topi untuk mengukur tingkat akurasi.</p> <p>Berdasarkan hasil pengujian, data yang diperoleh dapat dihasilkan untuk pengenalan wajah dengan menggunakan aksesoris topi untuk metode <i>eigenface</i> mencapai keakurasian 100%. Sedangkan, untuk pengenalan wajah dengan menggunakan aksesoris topi dengan wajah menghadap kesamping untuk metode <i>eigenface</i> mencapai keakurasian 62,5%. Ketika menghadap kesamping maka metode <i>eigenface</i> tidak dapat mendeteksi secara 100% wajah manusia, hal ini dikarenakan <i>database</i> yang ada hanya menghadap kedepan.</p>
3	<p>Perancangan Aplikasi <i>Scan Wajah Menggunakan Algoritma Eigenface</i></p>	2018	<p>Dalam Jurnal ini, peneliti melakukan penelitian mengenai perancangan aplikasi <i>scan wajah</i> menggunakan algoritma <i>eigenface</i></p>



	<p>Sebagai Alat Bantu Pengawas Ujian Agen Asuransi Pada PT. Adisarana Wanaartha Life Cabang The District Medan (Gulo et al., 2018)</p>	<p>sebagai alat bantu pengawas ujian, dengan menganalisa masalah tentang alat bantu <i>scan</i> wajah yang dirancang PT. Adisarana Wanaartha Life Cabang The District Medan dengan melakukan wawancara dan riset data untuk mengetahui apa kendala yang dihadapi.</p> <p>Berdasarkan hasil pengujian, pengenalan wajah ini dilakukan pada kondisi cahaya 500 - 1000 Lux, jarak 100 cm dan dengan kecepatan diam dan sedang, tingkat akurasi yang didapatkan dalam pengujian untuk 3 data wajah adalah 80%, 80% dan 90%, sehingga rata – rata keakuratan pendeteksian sample uji mencapai 83%. Telah berhasil diterapkan algoritma <i>eigenface</i> untuk mendeteksi wajah dengan bantuan <i>OpenCV</i>, sehingga dapat diimplementasikan ke dalam sistem dengan pencahayaan yang ideal yaitu 100-1500 Lux dan dengan jarak 100 cm serta dilakukan dalam keadaan diam dan menggunakan 50 sampel wajah agar dapat mengenali wajah dengan baik.</p>
--	--	--

4	<p>Pengembangan Identifikasi Wajah Menggunakan Fitur Citra Mata Menggunakan <i>Eigenface</i> (Siregar &amp; Sahputri, 2019)</p>	2018	<p>Dalam Jurnal ini, peneliti melakukan penelitian mengenai pengembangan identifikasi wajah menggunakan fitur citra mata menggunakan <i>eigenface</i>, Penelitian ini menggunakan pendekatan <i>skin detection</i> dengan model RGB untuk mengoptimalkan akurasi identifikasi wajah pada sistem yang diusulkan, citra wajah diambil menggunakan <i>webcam</i>, kemudian dilakukan akurasi wajah dan proses <i>skin detection</i> untuk optimalkan proses identifikasi fitur mata.</p> <p>Berdasarkan hasil pengujian, pendekatan <i>skin detection</i> model RGB terhadap identifikasi wajah berdasarkan ekstraksi fitur mata menggunakan <i>library open source OpenCV</i> pada sistem yang diusulkan memberikan pengaruh yang signifikan terhadap tingkat akurasi yakni sebesar 85.54% dibandingkan tanpa model RGB dengan tingkat akurasi 75.68%, namun jarak wajah terhadap <i>webcam</i> sangat berpengaruh</p>

			terhadap akurasi identifikasi sehingga perlu dilakukan penelitian lanjutan.
5	<p><i>Smart Home Security</i> menggunakan <i>Face Recognition</i> dengan Metode <i>Eigenface</i> Berbasis <i>Raspberry Pi</i> (Kurniawan &amp; Zulius, 2019)</p>	2019	<p>Dalam Jurnal ini, peneliti melakukan penelitian mengenai <i>smart home security</i> menggunakan <i>face recognition</i> dengan metode <i>eigenface</i> berbasis <i>raspberry pi</i>. Pada sistem pengenalan wajah ini dibagi menjadi dua tahapan, yaitu tahapan pengambilan gambar dan tahapan pengenalan. Didalam tahap pengambilan ini, gambar yang diambil akan dijadikan <i>database</i> lalu akan disimpan hasilnya. Setiap gambar berukuran (640x480) pixel. Setelah melalui proses itu, lalu gambar akan melalui tahap <i>eigenface</i>. Kemudian tahapan selanjutnya, merupakan tahapan pengenalan, kedua matriks dari gambar dibandingkan.</p> <p>Berdasarkan hasil penelitian, penggunaan metode <i>eigenface</i> dalam <i>smart home security</i> dengan <i>face recognition</i> menunjukkan hasil akurasi dengan rata-rata sebesar 72,5%. Dari hasil percobaan, jarak antara wajah yang akan dikenali dengan <i>webcam</i></p>

			<p>sangat berpengaruh terhadap proses pendeteksian wajah dengan jarak sejauh 25 cm dengan akurasi maksimum sebesar 90%. Sistem berhasil melakukan pengenalan meskipun posisinya berbeda-beda, karena yang digunakan adalah nilai dari <i>eigenface</i> tiap citra wajah yang dibandingkan. Penggunaan klas untuk pengelompokan data pemilik rumah dan bukan pemilik rumah sangat efektif digunakan dalam proses verifikasi antara pemilik atau pencuri. Tingkat keberhasilan pengenalan wajah dipengaruhi oleh deteksi wajah, pemrosesan awal dan penghitungan <i>eigenface</i> sebelumnya.</p>
6	<p>Perancangan Sistem Pengenalan Wajah Secara <i>Real Time</i> pada CCTV dengan Metode <i>Eigenface</i> (Jamhari et al., 2020)</p>	2020	<p>Dalam Jurnal ini, peneliti melakukan penelitian mengenai perancangan sistem pengenalan wajah secara <i>real time</i> pada <i>cctv</i> menggunakan algoritma <i>eigenface</i> dengan metode pembanding yang digunakan adalah LDA dan PCA. Berdasarkan hasil penelitian yang telah dilakukan, dari 9 sampel wajah orang, didapatkan bahwa algoritma <i>eigenface</i> dengan ekstraksi fitur PCA memiliki</p>

			<p>akurasi 98,06%, sedangkan <i>eigenface</i> dengan ekstraksi fitur LDA memiliki akurasi 97,73%. Hal ini terjadi karena PCA dapat memperkecil dimensi citra lebih sederhana dibandingkan LDA, membuat PCA memiliki keunggulan pada proses kecepatan mengambil keputusan untuk mengenali wajah di kondisi <i>real time</i>. Berdasarkan hasil akurasi tiap ekstraksi fitur, dapat disimpulkan bahwa PCA memiliki akurasi lebih baik dalam mengenali wajah manusia daripada LDA.</p>
--	--	--	---