



# Perbandingan Algoritma Fisherface dan Algoritma Local Binary Pattern Untuk Pengenalan Wajah

Nurul Amalia

Fakultas, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

Email: nurul01amalia@gmail.com

**Abstrak**—Sistem pengenalan wajah merupakan sistem yang digunakan untuk mendeteksi citra wajah, yang difungsikan untuk memberikan keakuratan pada sebuah sistem yang digunakan untuk kontrol akses bagi fasilitas yang membutuhkan sistem keamanan tersebut. Namun tidak jarang ditemukan masalah dalam proses pengenalan wajah, seperti kesulitan sistem untuk mengenali wajah jika ditemukan data training yang mirip antara satu dengan yang lain. Selain itu adanya kesulitan dalam proses pengenalan wajah dikarenakan kurangnya jumlah citra dan atau pose citra data training yang mengakibatkan sistem tidak optimal dalam mengenali wajah. Beberapa metode diajukan untuk menciptakan sistem pengenalan wajah yang handal, seperti Fisherface dan Local Binary Pattern. Kelebihan menggunakan metode Fisherface adalah saat waktu pemrosesan hanya memakan waktu yang relatif cepat. Ini disebabkan karena proses fisherfaces menggunakan proses berupa matriks. Selain metode fisherface, metode Local Binary Pattern juga digunakan dalam sistem pengenalan wajah. Metode ini juga dikenal cukup mudah dan efisien dalam penerapannya untuk mendapatkan karakteristik citra wajah. Dengan kelebihan dan kekurangan yang dimiliki masing-masing metode di atas, maka penulis akan melakukan uji perbandingan antara kedua metode tersebut untuk mengukur tingkat keakurasian dari masing-masing metode. Pada penelitian ini menggunakan dua algoritma yaitu algoritma Fisherface dan algoritma Local Binary Pattern. Dimana algoritma Fisherface merupakan turunan dari FLD yang dikombinasikan dengan (PCA). PCA bertugas mereduksi data masukan agar mempermudah dan mempercepat proses dan FLD bertugas untuk menghasilkan matriks sebaran untuk memudahkan klasifikasi dan pengenalan. Selain metode fisherface, algoritma Local Binary Pattern juga digunakan dalam sistem pengenalan wajah. LBP didefinisikan sebagai perbandingan nilai biner piksel pada pusat citra dengan 8 nilai piksel disekelilingnya dengan begitu dapat dicocokkan citra uji dengan citra acuan. Penelitian ini bermanfaat bagi para pengembangan sistem pengenalan wajah agar tepat dalam pemilihan metode, untuk meminimalisir terjadinya error dalam proses pengenalan wajah. Serta mendapat tingkat keakurasian yang tinggi dan kecepatan dalam pengenalan wajah.

**Kata Kunci:** Deteksi Wajah; Citra Pasfoto; Algoritma Fisherface; Algoritma; Local Binary Pattern

**Abstract**—Face recognition system is a system used to detect facial images, which is used to provide accuracy in a system used for access control for facilities that require such a security system. However, it is not uncommon to find problems in the face recognition process, such as the difficulty of the system to recognize faces if similar training data are found. In addition, there are difficulties in the face recognition process due to the lack of the number of images and/or poses of the training data images which results in the system not being optimal in recognizing faces. Several methods were proposed to create a reliable facial recognition system, such as Fisherface and Local Binary Pattern. The advantage of using the Fisherface method is that the processing time is relatively fast. This is because the fisherfaces process uses a matrix process. In addition to the fisherface method, the Local Binary Pattern method is also used in facial recognition systems. This method is also known to be quite easy and efficient in its application to obtain facial image characteristics. With the advantages and disadvantages of each of the above methods, the author will conduct a comparative test between the two methods to measure the level of accuracy of each method. In this study, two algorithms are used, namely the Fisherface algorithm and the Local Binary Pattern algorithm. Where the Fisherface algorithm is a derivative of FLD combined with (PCA). PCA is in charge of reducing input data to simplify and speed up the process and FLD is in charge of producing a distribution matrix to facilitate classification and recognition. In addition to the fisherface method, the Local Binary Pattern algorithm is also used in the face recognition system. LBP is defined as the comparison of the binary value of the pixel at the center of the image with the 8 values of the surrounding pixels so that the test image can be matched with the reference image. This research is useful for the development of facial recognition systems to be precise in the selection of methods, to minimize the occurrence of errors in the face recognition process. As well as getting a high level of accuracy and speed in facial recognition.

**Keywords:** Face Detection; Pasfoto Image; Fisherface Algorithm; Algorithm; Local Binary Pattern

## 1. PENDAHULUAN

Seiring dengan perkembangan teknologi informasi, kemajuan di bidang pengembangan aplikasi sedang mendapatkan perhatian penting bagi perkembangan teknologi informasi. Salah satu bidang yang sedang berkembang yaitu sistem pengenalan wajah. Pengenalan wajah merupakan sebuah sistem yang digunakan untuk mendeteksi citra wajah. Pengenalan wajah difungsikan untuk memberikan keakuratan pada sebuah sistem yang digunakan untuk kontrol akses bagi fasilitas yang membutuhkan sistem keamanan tersebut. Salah satu proses yang menggunakan teknologi pengenalan wajah ialah sistem absensi kehadiran pada suatu instansi atau perusahaan. Baik sekolah maupun perusahaan, saat ini sudah banyak yang menggunakan teknologi pengenalan wajah pada sistem kehadiran. Dengan menggunakan teknologi pengenalan wajah, baik mahasiswa maupun karyawan suatu perusahaan tidak lagi perlu melakukan absen secara manual, seperti mengisi data kehadiran secara tertulis ataupun tanda tangan pada buku kehadiran, dikarenakan hal ini cukup menyita waktu dan dinilai kurang efektif, karna seringkali mahasiswa menandatangani kolom kehadiran temannya yang tidak masuk.

Tak hanya pada sistem absensi, sistem pengenalan wajah biasanya juga digunakan sebagai pencarian orang hilang, pengawasan, identifikasi tersangka kejahatan, dan akses-akses menuju aset-aset berharga. Proses pengenalan wajah manusia dengan komputer dapat dilakukan dengan teknik pengolahan citra. Karna pendeteksian wajah (face detection) adalah suatu tahap awal yang sangat penting sebelum dilakukan proses pengenalan wajah (face recognition). Pendeteksian



wajah (face detection) dilakukan setelah dilakukannya input citra sebagai data training. Artinya wajah yang dapat dikenali hanya wajah yang telah dilakukan proses input citra sebelumnya. Pengenalan wajah manusia juga dapat di implementasikan ke dalam banyak aplikasi yang mencakup pengenalan wajah,

seperti pengawasan keamanan suatu area, pencarian database individu pada database citra digital dan sebagainya. Kelebihan sistem pengenalan wajah yaitu dapat mengenali secara tepat walaupun wajah tersebut mengalami perubahan seperti tumbuh kumis dan jenggot. Namun tidak jarang ditemukan masalah dalam proses pengenalan wajah, seperti kesulitan sistem untuk mengenali wajah jika ditemukan data training yang mirip antara satu dengan yang lain. Selain itu adanya kesulitan dalam proses pengenalan wajah dikarenakan kurangnya jumlah citra dan atau pose citra sebagai bahan data training yang mengakibatkan sistem tidak optimal dalam mengenali wajah. Beberapa metode diajukan untuk menciptakan sistem pengenalan wajah yang handal, seperti Fisherface dan Local binary pattern histogram.

Fisherface adalah salah satu metode yang digunakan dalam pembuatan sistem pengenalan wajah. Metode ini merupakan turunan dari Fisher's Linear Discriminant (FLD) yang dikombinasikan dengan Principal Component Analysis (PCA). PCA bertugas mereduksi data masukan agar mempermudah dan mempercepat proses dan FLD bertugas untuk menghasilkan matriks sebaran untuk memudahkan klasifikasi dan pengenalan. Selain metode fisherface, metode Local Binary Pattern (LBP) juga bisa digunakan dalam sistem pengenalan wajah. Metode ini juga dikenal cukup mudah dan efisien dalam penerapannya untuk mendapatkan karakteristik citra wajah. Local Binary Pattern (LBP) didefinisikan sebagai perbandingan nilai biner piksel pada pusat citra dengan 8 nilai piksel disekelilingnya dengan begitu dapat dicocokkan citra uji dengan citra acuan. Dengan kelebihan dan kekurangan yang dimiliki masing-masing metode di atas, maka penulis akan melakukan uji perbandingan antara kedua metode tersebut untuk mengukur tingkat keakurasian dari masing-masing metode. Agar kedepannya para pengembang sistem pengenalan wajah baik perusahaan ataupun instansi dapat lebih efektif dan tepat dalam memilih metode yang benar-benar sempurna dalam pengenalan wajah, sehingga peluang terjadinya error dalam proses pengenalan wajah sangat kecil. Serta mendapat tingkat keakurasian yang tinggi dan kecepatan dalam pengenalan wajah.

Pengenalan wajah merupakan sistem teknologi berbasis biometrik yang digunakan untuk mengidentifikasi seseorang dari suatu gambar digital ataupun video dengan cara menganalisa wajah mereka menggunakan algoritma tertentu. Beberapa penelitian sebelumnya yang juga membahas tentang metode Fisherface dan Local Binary Pattern. Pengenalan wajah menggunakan ekstraksi fitur Local Binary Pattern pada resolusi rendah 35 piksel, telah mencapai akurasi 90%. Metode Local Binary Pattern dikombinasikan dengan Principal Component Analysis untuk meningkatkan penelitian sebelumnya, berhasil mencapai akurasi 94% pada resolusi rendah [1]. Algoritma fisherfaces digunakan untuk pengenalan wajah. kelebihan menggunakan metode ini adalah waktu pemrosesan hanya memakan waktu yang relative singkat/cepat. Ini disebabkan karena proses fisherface menggunakan proses berupa matriks, sehingga sesuai untuk diterapkan kedalam sistem yang membutuhkan proses secara real-time Pengujian dilakukan dengan teknik Black Box. Dari pengujian dengan menggunakan 15 wajah sample menghasilkan akurasi sebesar 93,33% [2].

## **2. METODOLOGI PENELITIAN**

### **2.1 Citra Digital**

Citra Digital (digital image) adalah gambar pada bidang dwimatra (dua dimensi) yang dihasilkan dari gambar analog dua dimensi yang kontiniu menjadi gambar diskrit melalui proses digitasi. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Maksudnya, sebuah gambar dapat memberikan informasi yang lebih banyak daripada informasi yang disajikan dalam bentuk teks[1].

### **2.2 Pengenalan Wajah**

Wajah merupakan bagian depan dari kepala yang meliputi: daerah dari dahi hingga dagu, termasuk juga alis, mata, hidung, pipi, mulut, bibir, gigi dan kulit. Wajah digunakan untuk menunjukkan sikap seseorang terhadap ekspresi wajah, penampilan, serta identitas dari seseorang dan tidak ada satu pun wajah yang sama atau serupa mutlak, bahkan pada manusia kembar sekalipun. Wajah sebagai objek dua dimensi digambarkan dengan berbagai macam illuminasi, pose dan ekspresi wajah untuk diidentifikasi berdasarkan citra dua dimensi dari wajah tersebut. Oleh sebab itu dengan melihat wajah, manusia dapat mengenali atau mengidentifikasi seseorang dengan mudah[3]. Face Recognition merupakan pengembangan dari teknologi face detection dimana teknologi ini dapat menghasilkan/generate wajah dari hasil tangkapan kamera dan melakukan deteksi persamaan wajah dengan data wajah yang diketahui komputer, sehingga komputer dapat mengenali dan atau mengetahui keberadaan seseorang.

### **2.3 Algoritma Fisherface**

Dasar dari metode Fisherface ini adalah Fisher's Linear Discriminant (FLD). FLD ditemukan oleh Robert Fisher pada tahun 1936 untuk klasifikasi taksonomi dan menjadi salah satu teknik yang banyak digunakan dalam pengenalan pola (pattern recognition). FLD merupakan salah satu contoh metode class specific, karena metode ini berusaha untuk membentuk jarak (scatter) antar kelas dan intra kelas sehingga dapat menghasilkan klasifikasi yang lebih baik[3]

## 2.4 Algoritma Local Binary Pattern

Konsep dasar dari LBP yaitu menyimpulkan struktur lokal dari citra dengan membandingkan setiap piksel dengan piksel sekelilingnya. Cara perbandingan ini dilakukan dengan mengambil satu piksel tengah kemudian membandingkan nilainya dengan piksel sekelilingnya, jika nilai piksel yang mengelilinginya lebih besar atau sama dengan nilai pada piksel tengah maka piksel tersebut akan diberi nilai 1, sedangkan jika nilai dari piksel yang mengelilinginya kurang dari nilai piksel tengah maka diberi nilai 0, kemudian nilai dari setiap piksel pada citra akan menjadi angka biner. Setelah melakukan perbandingan nilai piksel keliling dengan nilai piksel tengah selanjutnya dilakukan penyusunan delapan nilai biner searah jarum jam dan merubah nilai biner tersebut kedalam nilai desimal untuk menggantikan nilai piksel tengah[4].

## 3. HASIL DAN PEMBAHASAN

Pada masa sekarang ini penggunaan kamera digital sebagai alat untuk mengambil citra telah menggantikan kamera analog karna kepraktisan maupun kemudahannya. Tetapi terkadang hasil citra yang didapatkan memiliki hasil visual yang kurang baik. Tampilan visual yang kurang baik tersebut disebabkan oleh dua faktor yaitu faktor luar dan faktor dalam. Faktor dalam ialah faktor yang disebabkan oleh kualitas kamera digital itu sendiri yang kurang baik, sehingga kualitas citra yang dihasilkan juga kurang baik. Sedangkan faktor luar adalah faktor yang disebabkan oleh adanya noise seperti kualitas pencahayaan pada citra digital yang tidak merata.

Pengenalan wajah merupakan salah satu bidang yang menarik untuk diteliti. Dan hingga saat ini masih belum ditemukan suatu metode yang benar-benar tepat dan sempurna dalam pengenalan wajah. Dikarnakan wajah manusia memiliki ciri yang kompleks. Sehingga untuk mengembangkan suatu perhitungan yang ideal merupakan hal yang sulit. Beberapa peneliti telah menggunakan metode yang berbeda-beda untuk pengenalan wajah, dan dari setiap metode-metode tersebut memiliki kelebihan dan juga kekurangan. Bahkan beberapa peneliti sengaja menggabungkan metode tersebut agar mendapatkan hasil yang lebih baik, tetapi masih saja belum dapat dikatakan sempurna. Dari sekian banyak pengembangan dan implementasi aplikasi pengenalan wajah, belum suatu aplikasi yang mengkombinasikan kecepatan, keakuratan, dan kemudahan terhadap variasi citra wajah.

Kesulitan-kesulitan yang biasa ditemui pada sistem pengenalan wajah biasanya dikarenakan karna variasi dalam satu wajah bisa sangat besar, namun variasi antara wajah yang berbeda sangat kecil. Deteksi wajah didasarkan pada identifikasi dan menentukan lokasi citra wajah manusia dalam gambar terlepas dari ukuran, posisi dan kondisi. Penelitian sistem pengenalan wajah manusia menggunakan metode Fisherface dan Local Binary Pattern. Sebelum proses pengenalan wajah berlangsung citra training melalui tahap pra-proses untuk dinormalisasi.

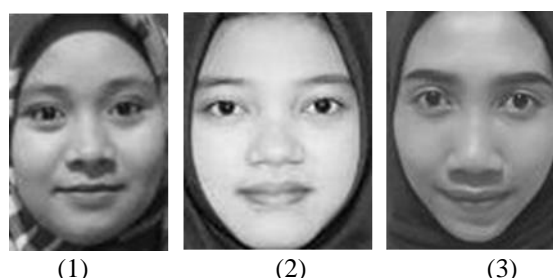
### 3.1 Penerapan Algoritma Fisherface

Berikut ini merupakan citra pasfoto untuk menganalisa sistem pengenalan wajah menggunakan metode *Fisherface*. Data yang digunakan dalam penelitian ini adalah citra wajah dengan jenis citra berwarna. Langkah pertama yang dilakukan pada penelitian ini adalah mengubah citra latih menjadi citra *grayscale* dan dibentuk menjadi matriks. Dalam penelitian ini, citra yang telah disiapkan dimasukkan ke komputer untuk diproses. Data *training* yang digunakan sebanyak 3 data *input* dengan ukuran 300 x 400 piksel. Gambar 1 di bawah ini merupakan citra *input* yang akan akan digunakan.



**Gambar 1.** Citra *Training* Pasfoto.jpg

Gambar 2 di bawah ini merupakan citra berwarna hasil dari proses deteksi wajah yang telah di ubah ke *grayscale* untuk mendapatkan nilai konversinya. Dapat dilihat seperti pada gambar 1 di bawah ini.



**Gambar 2.** Citra *Training* Hasil Konversi *Grayscale*



Kemudian setelah selesai di *grayscale* maka dapat diperoleh nilai konversinya. Berikut ini adalah reduksi dimensi pada citra *training* yang ditampilkan berukuran 5 x 5 piksel.

**Tabel 1.** Nilai Piksel 5 x 5 Citra *Training* (1)

94	78	74	72	93
63	46	44	51	63
65	52	50	64	65
67	63	61	68	67
68	74	71	64	68

**Tabel 2.** Nilai Piksel 5 x 5 Citra *Training* (2)

128	122	108	121	124
115	116	116	125	123
121	117	95	99	100
160	167	175	173	165
205	212	202	203	196

**Tabel 3.** Nilai Piksel 5 x 5 Citra *Training* (3)

72	67	68	83	104
73	159	42	47	44
55	99	41	55	54
46	53	40	46	55
70	55	56	55	422

Dari tabel di atas matrik piksel yang telah diambil dari 5 x 5 berdasarkan setiap unsur warna dapat dilihat dengan penyelesaian menggunakan Algoritma *Fisherfaces* pada gambar di bawah ini:

1. Mentransformasikan data *training* ke dalam vektor kolom (*face space*) Untuk melakukan proses transformasi data *training* ke dalam vektor kolom (*face space*) menggunakan Matlab hasilnya ditunjukkan pada tabel 4.

**Tabel 4.** Matriks Nilai *Face Space* (*T*)

<i>Face Space</i> ( <i>T</i> )		
Data <i>Training</i> 1	Data <i>Training</i> 2	Data <i>Training</i> 3
94	128	72
78	122	67
74	108	68
72	121	83
93	124	104
63	115	73
46	116	159
44	116	42
51	125	47
63	123	44
65	121	55
52	117	99
50	95	41
64	99	55
65	100	54
67	160	46
63	167	53
61	175	40
68	173	46
67	165	55
68	205	70
74	212	55
71	202	56
64	203	55
68	196	42

Setelah semua matriks citra data *training* diubah menjadi matriks 1D maka langkah selanjutnya adalah menghitung nilai *mean* dari matriks *T*

2. Menghitung Nilai *Mean*



Nilai *mean* merupakan jumlah baris dari semua data *training* dan dibagi dengan jumlah data *training*. Sehingga dapat ditulis dengan persamaan 6.

$$M = \text{sum\_tiapbaris} / \text{jumlah\_data\_training}$$

**Tabel 5.** Nilai *Mean*

Indeks	Nilai <i>M</i>	Indeks	Nilai <i>M</i>	Indeks	Nilai <i>M</i>
1	98	10	77	19	96
2	89	11	80	20	96
3	83	12	89	21	114
4	92	13	62	22	114
5	107	14	73	23	110
6	84	15	73	24	107
7	107	16	91	25	102
8	67	17	94		
9	74	18	92		

Setelah mendapatkan seluruh nilai *mean* dari data *training* maka langkah selanjutnya adalah menghitung matriks *A*.

3. Menghitung Matriks *A*

Setelah mendapatkan nilai rata-rata dari setiap baris maka langkah selanjutnya adalah mengurangi data *training* dengan nilai *mean*, dapat ditulis dengan persamaan 7.

$$A = T - M$$

**Tabel 6.** Matriks *A*

<i>Matriks A</i>		
Data Training 1	Data Training 2	Data Training 3
-4	30	-26
-11	33	-22
-9	25	-15
-20	29	-9
-14	17	-3
-21	31	-11
-61	9	52
-23	49	-25
-23	51	-27
-14	46	-33
-15	41	-25
-37	28	10
-12	33	-21
-9	26	-18
-8	27	-19
-24	68	-45
-31	73	-41
-31	83	-52
-28	77	-50
-29	69	-41
-46	91	-44
-40	98	-59
-39	92	-54
-43	96	-52
-34	94	-60

Setelah menghitung matriks *A* dengan mengurangi semua data *training* dengan rata-rata nilai setiap baris maka langkah selanjutnya adalah menghitung matriks kovarian dari matriks *A*.

4. Menghitung Matriks Kovarian

Matriks kovarian dari matriks *A* merupakan perkalian matriks *A* dengan matriks *A'*, dapat ditulis dengan persamaan 7.

**Tabel 7.** Matriks Kovarian (*L*)

<i>Matriks Kovarian (L)</i>		
Data Training 1	Data Training 2	Data Training 3
391876	-824442	431940
-824442	1734489	-908730
431940	-908730	476100



Perintah yang diketikkan pada Matlab *Command*

```
A = [ 391876 -824442 431940; -824442 1734489 -908730;
      431940 -908730 476100
```

Setelah mendapatkan nilai matriks kovarian (*L*) maka langkah selanjutnya adalah mencari nilai *Eigen Value (D)* dan *Eigen Vector (V)* dari matriks kovarian (*L*).

5. Menghitung *Eigen Value (D)* dan *Eigen Vector (V)*

*Eigen Value (D)* merupakan nilai karakteristik dari suatu matriks bujur sangkar yang lebih besar dari 0. Menggunakan aplikasi matlab untuk mendapatkan *Eigen Value (D)* dan *Eigen Vector (V)*, adapun perintah yang digunakan yaitu :

Perintah yang diketikkan pada Matlab *Command*

```
>> A = [ 391876 -824442 431940; -824442 1734489 -908730;
        431940 -908730 476100
```

Maka dari hasil aplikasi matlab tersebut diketahui nilai *Eigen Value (D)* dapat dilihat pada table 3.5 di bawah ini.

**Tabel 8 .Eigen Value (D)**

<i>Eigen Value (D)</i>		
Data Training 1	Data Training 2	Data Training 3
-0.0000	0	0
0	0,0000	0
0	0	2.6025

Setelah nilai *Eigen Value (D)* diperoleh, maka tahap selanjutnya yaitu menghitung *Eigen Vector (V)*. Nilai ini yang nantinya akan digunakan untuk menghitung nilai *Eigenface*. Untuk mencari nilai ini menggunakan fungsi khusus yang telah tersedia dalam Matlab.

**Tabel 9. Nilai Eigen Vector (V)**

<i>Eigen Vector (V)</i>		
V 1	V 2	V 3
-0.8820	0.2675	0.3880
-0.4636	-0.3443	-0.8164
-0.0848	-0.8999	0.4277

Setelah menemukan nilai *Eigen Vector (V)*, maka langkah selanjutnya adalah mengurutkan dan mengeliminasi *Eigen Value (D)* terkecil.

6. Mengurutkan dan mengeliminasi nilai *Eigen Value (D)* terkecil

Mengurutkan dan mengeliminasi *Eigen Value (D)* terkecil berfungsi untuk membuang nilai ciri yang tidak penting sehingga akan memudahkan proses selanjutnya karena dimensi data semakin kecil.

**Tabel 10. Eliminasi Eigen Value (D)**

<i>Eigen Value D</i>	
-0.8820	-0.8820
-0.4636	-0.4636
-0.0848	-0.0848

7. Menghitung nilai *Eigenface*

Setelah mengeliminasi nilai *Eigen Value (D)* maka langkah selanjutnya adalah menghitung nilai *Eigenface* dengan cara mengalikan matriks *A* dengan matriks *Eigen Value (D)*.

**Tabel 11. Nilai Eigenface**

<i>Eigenface</i>		
-0.0000	-0.0000	1.1241
0.0000	0.0000	-2.3649
-0.0000	-0.0000	1.2390

8. Memproyeksikan citra data *training* ke ruang *Eigen*

Setelah mendapatkan nilai *Eigenface* maka langkah selanjutnya adalah memproyeksikan citra data *training* ke ruang *Eigen* dengan cara mengalikan nilai *Eigenface* transpose dengan matriks *A*. Nilai proyeksi citra data *training* ke ruang *Eigen* ini akan digunakan untuk menghitung kelas *Within Scatter Matrix* dan digunakan untuk menghitung kelas *Between Scatter Matrix*. Seperti ditunjukkan pada Tabel 12

**Tabel 12 .Perkalian (A\*D)\*A**

<i>Hasil Matrix (A*D)*A</i>		
-0.0000	0.0000	-0.0000
-0.0000	0.0000	-0.0000
2.9255	-6.1547	3.2245

**Tabel 13.** Proyeksi Ruang *Eigen*

<i>Ruang Eigen</i>		
<i>Data Training 1</i>	<i>Data Training 2</i>	<i>Data Training 3</i>
895.43	-1883.83	986.97
611.41	-1286.31	673.92
348.68	-733.56	384.33

langkah selanjutnya yaitu mencari nilai rata-rata dari nilai proyeksi ruang *Eigen* dapat dilihat pada Tabel 14 berikut ini.

**Tabel 14** Nilai Rata-Rata Ruang *Eigen*

$M_{eigen}$
-0.47666
-0.32666
-0.18333

9. Proses Pengenalan Citra Wajah

Proses pengenalan adalah proses dimana wajah baru masuk dan akan dicocokkan dengan wajah yang ada di dalam *database* untuk mengetahui wajah mana yang cocok antara wajah yang baru masuk dengan wajah yang ada di dalam *database*. Untuk mengenali wajah baru yang masuk (*test face*), langkah yang dilakukan sama dengan data wajah yang ada di *database*, untuk mendapatkan nilai *fisherface* dari wajah baru.



(d)

**Gambar 3.** Data Uji Hasil Konversi *Grayscale*

Kemudian setelah di *grayscale* maka, langkah selanjutnya mencari nilai konversinya. Berikut ini adalah reduksi dimensi pada citra uji berukuran 5 x 5 piksel yang ditunjukkan pada Tabel 15.

**Tabel 15.** Matrik Piksel 5 x 5 dari citra uji (1)

113	114	113	112	114
104	106	104	102	103
93	95	94	94	96
77	79	79	80	84
66	68	68	70	74

Langkah selanjutnya adalah mereduksi citra 2D menjadi citra 1D yang disebut matriks augmentasi (*T*).

**Tabel 16.** Matriks Augmentasi (*T*)

Indeks	Nilai <i>M</i>	Indeks	Nilai <i>M</i>	Indeks	Nilai <i>M</i>
1	113	10	103	19	80
2	114	11	93	20	84
3	113	12	95	21	66
4	112	13	94	22	68
5	114	14	94	23	68
6	104	15	96	24	70
7	106	16	77	25	74
8	104	17	79		
9	102	18	79		

Setelah mereduksi citra uji menjadi augmentasi (*T*), maka langkah selanjutnya adalah menghitung selisih citra uji dengan nilai rata-rata baris. Cara menghitung mean dari citra uji adalah dengan mengurangkan nilai matriks augmentasi (*T*) data citra uji dengan nilai *mean* citra data *training*.  $Difference = citra\ data\ uji - M$

**Tabel 17.** Nilai Selisih Citra Data Uji

Indeks	Nilai <i>M</i>	Indeks	Nilai <i>M</i>	Indeks	Nilai <i>M</i>
1	15	10	26	19	-16
2	25	11	13	20	-12
3	30	12	6	21	-48
4	20	13	32	22	-46
5	7	14	21	23	-42
6	20	15	23	24	-37
7	-1	16	-14	25	-28
8	37	17	-15		
9	28	18	-13		

Pada Tabel 17 berisi nilai data uji yang *contered* (berpusat) yang akan dihitung selisihnya dengan nilai rata-rata baris. Selanjutnya mencari nilai *Principal Component Analysis* citra yang di uji menggunakan persamaan berikut:  $PCA\_Test = Eigenface' \times Difference$

**Tabel 18.** Nilai *PCA* Citra Data Uji

Nilai <i>PCA</i> Data Uji (d)
34.4871
-73.3119
38.409

Setelah nilai ciri data *training* dan citra uji diperoleh, maka langkah selanjutnya menghitung *Euclidean Distance* pada proses pengenalan wajah. Langkah selanjutnya yaitu proses akhir dari pengenalan wajah. Yaitu dengan menghitung *Similarity Distance* (kesamaan jarak) antara ciri dari citra *training* dengan ciri citra data uji. Perhitungan ini menggunakan *Euclidean Distance*.

**Tabel 19.** Perhitungan *Euclidean Distance* Data Uji

<i>Euclidean Distance</i> Data Uji		
Data <i>Training</i> 1	Data <i>Training</i> 2	Data <i>Training</i> 3
33.363	-70.947	37.17

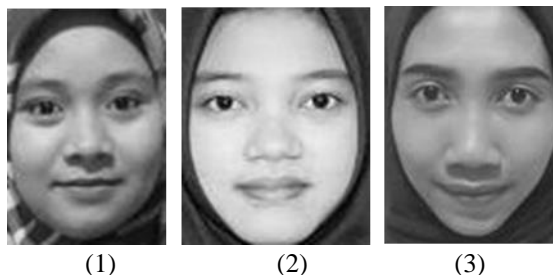
Pada Tabel 19 nilai *PCA Test* adalah nilai ciri *PCA* dari data uji, sedangkan nilai *PCA train* adalah nilai ciri *PCA* dari data *training*. Hasil perhitungan menyatakan bahwa nilai terkecil merupakan nilai yang diambil, maka *output* nilai data *itraining* 2 menjadi nilai *output* sistem pengenalan wajah ini.



**Gambar 4.** Citra Hasil Pengenalan

### 3.2 Penerapan Algoritma Local Binary Pattern

Pada penelitian tentang pengenalan wajah menggunakan algoritma *Local Binary Pattern*, data yang digunakan adalah data yang sama dengan penelitian pengenalan wajah menggunakan algoritma *Fisherface*. Gambar di bawah ini adalah citra berwarna hasil dari proses deteksi wajah yang telah di ubah ke *grayscale* untuk mendapatkan nilai konversinya.



**Gambar 5.** Tiga gambar input hasil konversi *grayscale*

Berikut ini adalah reduksi dimensi pada citra *training* yang berukuran 5 x 5 piksel. Untuk nilai konversinya ditampilkan pada Tabel 20 sampai 22 dibawah ini.





**Tabel 20.** Nilai Piksel 5 x 5 Citra *Training* (1)

94	78	74	72	93
63	46	44	51	63
65	52	50	64	65
67	63	61	68	67
68	74	71	64	68

**Tabel 21.** Nilai Piksel 5 x 5 Citra *Training* (2)

128	122	108	121	124
115	116	116	125	123
121	117	95	99	100
160	167	175	173	165
205	212	202	203	196

**Tabel 22.** Nilai Piksel 5 x 5 Citra *Training* (3)

72	67	68	83	104
73	159	42	47	44
55	99	41	55	54
46	53	40	46	55
70	55	56	55	42

1. Menghitung Nilai Biner

Kemudian setelah selesai di *grayscale* maka langkah selanjutnya adalah menghitung nilai biner. Untuk mendapatkan nilai binernya dapat dilakukan perhitungan dengan membagi gambar ke bagian-bagian yang lebih kecil atau sering disebut mask/kernel/template. kernel yang akan digunakan disini adalah 5x5 pixel. Dengan memiliki 9 pixel sebagai nilai acuan atau pusat pixel yang akan dibandingkan dengan 8 pixel sekelilingnya.

**Tabel 23.** nilai pixel citra 1

	y+1	y	y-1	width
x+1	46	44	51	
<u>x</u>	52	50	64	
x-1	63	61	68	
heigh				

2. Memberi Nilai Perbandingan

Selanjutnya adalah memberi nilai perbandingan.  $x > 0$  dan  $x < \text{lebar citra dikurang 1}$  dan  $y > 0$  dan  $y < \text{tinggi citra dikurang 1}$ .

**Tabel 24.** nilai biner citra 1

	y+1	y	y-1	width
x+1	0	0	1	
<u>x</u>	1		1	
x-1	1	1	1	
heigh				

Penyusunan nilai biner pada Tabel 24 dapat dilakukan dengan ketentuan searah jarum jam atau berlawanan jarum jam, sehingga nilai biner yang di dapat adalah = 1111100.

3. mengambil nilai pixel tengah dan pixel tetangga

Langkah selanjutnya adalah mengambil nilai pixel tengah  $ic$  dan pixel ketetangaan dari  $i7$  sampai  $i0$ . Untuk mendapatkan nilai  $ic$  yaitu dengan mengalikan nilai biner dengan bilangan eksponensial dari 2.

**Tabel 25.** hasil nilai tengah  $ic$

	y+1	y	y-1	width
x+1				
<u>x</u>		252		
x-1				
heigh				

Setelah mendapatkan nilai  $ic$  maka langkah selanjutnya adalah mengubah semua nilai warna pada pixel  $(x,y)$  dengan  $value$  pada variabel nilai.

**Tabel 26.** nilai *ic* citra 1

	y	y	y-1	width
x+1	46	44	51	
<u>x</u>	52	252	64	
x-1	63	61	68	
heigh				

4. penjumlahan nilai y dan memproses pixel selanjutnya

Selanjutnya melakukan penjumlahan nilai y dan memproses pixel selanjutnya. Setelah semua pixel di proses, maka akan terbentuk nilai citra hasil *LBP* seperti yang ditunjukkan pada Tabel 27 dibawah ini

**Tabel 27.** nilai hasil *LBP* citra training (1)

94	78	74	72	93
63	247	255	63	63
65	241	252	56	65
67	241	152	80	67
68	74	71	64	68

5. Proses Pengenalan Wajah

Setelah melakukan tahap *pre processing* menggunakan algoritma *local binary pattern* maka tahap selanjutnya adalah melakukan proses pengenalan wajah. Pada tahap pengenalan wajah menggunakan proses perhitungan *LBP* juga menggunakan data uji yang sama dengan yang sebelumnya.



**Gambar 6.** Citra uji yang telah di *grayscale*

Pada gambar 6 citra uji yang dilakukan telah di ubah menjadi citra *grayscale*. dan telah direduksi seperti pada tahap *Pre-processing* seperti yang ditunjukkan pada Tabel 28.

**Tabel 28.** Nilai matriks dari citra uji (1)

113	114	113	112	114
104	106	104	102	103
93	95	94	94	96
77	79	79	80	84
66	68	68	70	74

Langkah pertama yang dilakukan adalah menghitung histogram pada citra *training* dan citra uji. Untuk mendapatkan nilai histogram dari masing masing citra maka, dilakukan beberapa tahapan, tahap pertama yang harus dilakukan adalah menghitung frekuensi dan distribusi kumulatif dari nilai skala keabuan.

**Tabel 29.** frekuensi distribusi kumulatif citra training (1)

Skala Keabuan	Frek.	Distribusi Kumulatif	Skala Keabuan	Frek.	Distribusi Kumulatif
56	1	1	78	1	15 + 1 = 16
63	3	1 + 3 = 4	80	1	16 + 1 = 17
64	1	4 + 1 = 5	93	1	17 + 1 = 18
65	2	5 + 2 = 7	94	1	18 + 1 = 19
67	2	7 + 2 = 9	152	1	19 + 1 = 20
68	2	9 + 2 = 11	241	2	20 + 2 = 22
71	1	11 + 1 = 12	247	1	22 + 1 = 23
72	1	12 + 1 = 13	252	1	23 + 1 = 24
74	2	13 + 2 = 15	255	1	24 + 1 = 25

Tabel 29 merupakan daftar frekuensi dan perhitungan distribusi kumulatif dari nilai skala keabuan yang diperoleh dari citra training.



Tabel 30. frekuensi distribusi kumulatif citra uji

Skala Keabuan	Frek.	Distribusi Kumulatif	Skala Keabuan	Frek.	Distribusi Kumulatif
66	1	1	95	1	13 + 1 = 14
68	2	1 + 2 = 3	96	1	14 + 1 = 15
70	1	3 + 1 = 4	102	1	15 + 1 = 16
74	1	4 + 1 = 5	103	1	16 + 1 = 17
77	1	5 + 1 = 6	104	2	17 + 2 = 19
79	2	6 + 2 = 8	106	1	19 + 1 = 20
80	1	8 + 1 = 9	112	1	20 + 1 = 21
84	1	9 + 1 = 10	113	2	21 + 2 = 23
93	1	10 + 1 = 11	114	2	23 + 2 = 25
94	2	11 + 2 = 13			

Pada Tabel 30 diperoleh nilai frekuensi dan distribusi kumulatif pada citra uji, selanjutnya menghitung nilai keabuan dari hasil perhitungan distribusi kumulatif pada masing masing citra menggunakan rumus :

Dimana :  $K_o = \text{round} \left( \frac{C_i \cdot (2^k - 1)}{w \cdot h} \right)$

Ci = distribusi kumulatif dari nilai skala keabuan ke-I dari citra asli

Round = fungsi pembulatan ke bilangan yang terdekat

Ko = nilai keabuan hasil histogram

W = lebar citra

H = tinggi citra

Sehinga untuk citra training diperoleh skala keabuan 2

$K_o = \text{round} \left( \frac{1 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{255}{25} \right) = 10$

$K_o = \text{round} \left( \frac{4 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{1020}{25} \right) = 41$

$K_o = \text{round} \left( \frac{5 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{1275}{25} \right) = 51$

$K_o = \text{round} \left( \frac{7 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{1785}{25} \right) = 71$

$K_o = \text{round} \left( \frac{9 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{2295}{25} \right) = 92$

$K_o = \text{round} \left( \frac{11 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{2805}{25} \right) = 112$

$K_o = \text{round} \left( \frac{12 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{3060}{25} \right) = 122$

$K_o = \text{round} \left( \frac{13 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{3315}{25} \right) = 133$

$K_o = \text{round} \left( \frac{15 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{3825}{25} \right) = 153$

$K_o = \text{round} \left( \frac{16 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{4080}{25} \right) = 163$

$K_o = \text{round} \left( \frac{17 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{4335}{25} \right) = 173$

$K_o = \text{round} \left( \frac{18 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{4590}{25} \right) = 184$

$K_o = \text{round} \left( \frac{19 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{4845}{25} \right) = 194$

$K_o = \text{round} \left( \frac{20 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{5100}{25} \right) = 204$

$K_o = \text{round} \left( \frac{22 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{5610}{25} \right) = 224$

$K_o = \text{round} \left( \frac{23 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{5865}{25} \right) = 135$

$K_o = \text{round} \left( \frac{24 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{6120}{25} \right) = 245$

$K_o = \text{round} \left( \frac{25 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{6375}{25} \right) = 255$

Pada citra uji juga dilakukan cara yang sama untuk memperoleh skala keabuan 2.

$K_o = \text{round} \left( \frac{1 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{255}{25} \right) = 10$

$K_o = \text{round} \left( \frac{3 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{765}{25} \right) = 31$

$K_o = \text{round} \left( \frac{4 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{1020}{25} \right) = 41$

$K_o = \text{round} \left( \frac{5 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{1275}{25} \right) = 51$

$K_o = \text{round} \left( \frac{6 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{1530}{25} \right) = 61$



$$K_o = \text{round} \left( \frac{8 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{2040}{25} \right) = 82$$

$$K_o = \text{round} \left( \frac{9 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{2295}{25} \right) = 92$$

$$K_o = \text{round} \left( \frac{10 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{2550}{25} \right) = 102$$

$$K_o = \text{round} \left( \frac{11 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{2805}{25} \right) = 112$$

$$K_o = \text{round} \left( \frac{13 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{3315}{25} \right) = 133$$

$$K_o = \text{round} \left( \frac{14 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{3570}{25} \right) = 143$$

$$K_o = \text{round} \left( \frac{15 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{3825}{25} \right) = 153$$

$$K_o = \text{round} \left( \frac{16 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{4080}{25} \right) = 163$$

$$K_o = \text{round} \left( \frac{17 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{4335}{25} \right) = 173$$

$$K_o = \text{round} \left( \frac{19 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{4845}{25} \right) = 194$$

$$K_o = \text{round} \left( \frac{20 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{5100}{25} \right) = 204$$

$$K_o = \text{round} \left( \frac{21 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{5355}{25} \right) = 214$$

$$K_o = \text{round} \left( \frac{23 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{5865}{25} \right) = 235$$

$$K_o = \text{round} \left( \frac{25 \times (2^8 - 1)}{5 \times 5} \right) = \text{round} \left( \frac{6375}{25} \right) = 255$$

Hasil perhitungan pada seluruh skala nilai keabuan pada citra *training* dapat dilihat pada Tabel 31 dibawah.

**Tabel 31.** hasil skala keabuan citra *training*

Keabuan awal	Frek.	Keabuan hasil
56	1	10
63	3	41
64	1	51
65	2	71
67	2	92
68	2	112
71	1	122
72	1	133
74	2	153
78	1	163
80	1	173
93	1	184
94	1	194
152	1	204
241	2	224
247	1	235
252	1	245
255	1	255

**Tabel 32.** hasil skala keabuan citra uji

Keabuan awal	Frek.	Keabuan hasil
66	1	10
68	2	31
70	1	41
74	1	51
77	1	61
79	2	82
80	1	92
84	1	102
93	1	112
94	2	133
95	1	143
96	1	153



102	1	163
103	2	173
104	1	194
106	1	204
112	1	214
113	2	235
114	2	255

Tabel 32 adalah hasil skala keabuan dari citra uji untuk mendapatkan nilai histogram citra. Sedangkan matriks citra setelah proses *histogram* dapat dilihat pada Tabel 33, berikut hasil matriks citra *training* yang diperoleh dari proses perhitungan histogram.

**Tabel 33.** hasil skala keabuan citra training baru (1)

194	163	153	133	184
41	235	255	41	41
122	224	245	10	71
92	224	204	173	92
112	153	71	51	112

**Tabel 34.** Hasil skala keabuan citra uji baru (1)

235	255	235	255	255
194	204	194	102	173
112	143	133	133	153
61	82	82	92	163
10	31	31	41	51

Setelah melakukan perhitungan pada seluruh citra *training* dan citra uji, maka tahap akhir dari proses pengenalan wajah adalah menghitung jarak dari citra *training* dan citra uji. Perhitungan ini menggunakan *ecludian distance*.

$$\epsilon 1 = \|\Omega - \Omega_{new}\|$$

$$\begin{bmatrix} 194 & 163 & 153 & 133 & 184 \\ 41 & 235 & 255 & 41 & 41 \\ 122 & 124 & 145 & 10 & 71 \\ 92 & 224 & 204 & 173 & 92 \\ 112 & 153 & 71 & 51 & 112 \end{bmatrix} \begin{bmatrix} 235 & 255 & 235 & 255 & 255 \\ 194 & 204 & 194 & 102 & 173 \\ 112 & 143 & 133 & 133 & 153 \\ 61 & 82 & 82 & 92 & 163 \\ 10 & 31 & 31 & 41 & 51 \end{bmatrix}$$

$$\begin{aligned} &= \sqrt{(194)^2 + (163)^2 + (153)^2 + (133)^2 + (184)^2 + (41)^2 + (235)^2 + (255)^2 + (41)^2 + (41)^2 + (122)^2 + (124)^2 + (145)^2 + (10)^2 + (71)^2 + (92)^2 + (224)^2 + (204)^2 + (173)^2 + (92)^2 + (112)^2 + (153)^2 + (71)^2 + (51)^2 + (112)^2} \\ &= \sqrt{37636 + 26569 + 23409 + 17689 + 33856 + 1681 + 55225 + 65025 + 1681 + 1681 + 14884 + 15376 + 21025 + 100 + 5041 + 8464 + 50176 + 41616 + 29929 + 8464 + 12544 + 23409 + 5041 + 2601 + 12544} \\ &= 51,5666 \\ &= \sqrt{(235)^2 + (255)^2 + (235)^2 + (255)^2 + (255)^2 + (194)^2 + (204)^2 + (194)^2 + (102)^2 + (173)^2 + (112)^2 + (143)^2 + (133)^2 + (133)^2 + (153)^2 + (61)^2 + (82)^2 + (82)^2 + (92)^2 + (163)^2 + (10)^2 + (31)^2 + (31)^2 + (41)^2 + (51)^2} \\ &= \sqrt{55225 + 65025 + 55225 + 65025 + 65025 + 37636 + 41616 + 37636 + 10404 + 29929 + 12544 + 20449 + 17689 + 17689 + 23409 + 3721 + 6724 + 6724 + 8464 + 26569 + 100 + 961 + 961 + 1681 + 2601} \\ &= 61,3032 \\ &= 51,5666 - 61,3032 = -97,366 \end{aligned}$$

Pada perhitungan jarak antara histogram citra *training* dengan citra uji diperoleh hasil akhir seperti pada Tabel 35.







**Tabel 35.** jarak Histogram Citra

Citra Training	Citra Uji	Nilai
(1)	(1)	-97,366
(2)	(1)	-35,3740
(3)	(1)	19,331

### 3.4 Perbandingan Algoritma Fisherface dan Algoritma Local Binary Pattern

Perbandingan Algoritma *Fisherface* dan Algoritma *Local Binary Pattern* dilakukan untuk menganalisis kinerja dari algoritma berdasarkan hasil, nilai kemiripan dan tingkat kesulitan dari masing-masing metode. *Local Binary pattern* merupakan nilai yang optimal, karna memiliki selisih nilai yang minim (terkecil) dibanding algoritma *fisherface*.

**Tabel 36.** Hasil Data Uji

No	Gambar Training	Data Uji	Nilai Fisherface	Nilai LBP	Keterangan
1			33.363	-97.366	Tidak Ditemukan / Tidak mirip
2			-70.947	-35.3740	Tidak Ditemukan / Tidak mirip
3			37.17	19.331	Tidak Ditemukan / Tidak mirip

Dari hasil perhitungan jarak dengan *ecludian distance* bahwa nilai terkecil merupakan nilai yang diambil, maka algoritma *local binary pattern* merupakan algoritma dengan hasil *ecludian distance* dengan nilai terkecil yaitu 19.331 pada citra *training* 3. Sedangkan *fisherface* memiliki nilai terkecil yaitu 33.363 pada citra *training* 1.

**Tabel 37.** Hasil Rata-rata Data Uji

Operator	Citra uji	
	Nilai <i>ecludian distance</i> terkecil	Rata-rata
Fisherface	33.363	47,16%
Lbp	19.331	50.69%

Hasil akurasi yang tinggi bisa saja didapatkan, Tingkat kinerja *processor* berbeda-beda setiap dilakukan pengujian. Tingkat penggunaan *processor* sangat berpengaruh pada sistem ini dalam mengenali wajah manusia. Serta banyaknya data *training* juga sangat mempengaruhi keakuratan dalam pengenalan wajah.

## 4. KESIMPULAN

Berdasarkan hasil penelitian disimpulkan unsur pencahayaan atau kontras yang ada di sekitar objek mempengaruhi proses pendeteksian wajah dengan citra pasphoto. Pengenalan wajah dengan menggunakan Algoritma Local binary pattern pada penelitian ini memiliki nilai yang optimal dan tingkat akurasi lebih tinggi dibandingkan algoritma *Fisherface*. Hasil pengenalan wajah diambil dari nilai jarak euclidean terbesar. Nilai jarak euclidean yang semakin besar menandakan bahwa citra wajah yang hendak dikenali mirip dengan citra wajah yang terdapat di dalam database.

## REFERENCES

- [1] A. B. S and H. Maulana, "Pengenalan Citra Wajah Sebagai Identifier Menggunakan Metode Principal Component Analysis (PCA)," vol. 9, no. 2, pp. 166-175, 2016.
- [2] A. R. Putri, "Pengolahan Citra Dengan Menggunakan Web Cam Pada Kendaraan Bergerak Di Jalan Raya," JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform., vol. 1, no. 01, pp. 1-6, 2016, doi: 10.29100/jipi.v1i01.18.
- [3] D. Teknik, I. Universitas, M. Lhokseumawe, M. Teknik, and I. Universitas, "Pengenalan Bentuk Wajah Manusia Pada Citra Menggunakan Metode Fisherface," TECHSI J. Penelit. Tek. Inform., vol. 15, pp. 1-11, 2015.



- [4] L. W. Alexander, S. R. Sentinuwo, A. M. Sambul, T. Informatika, U. Sam, and R. Manado, "Implementasi Algoritma Pengenalan Wajah Untuk Mendeteksi Visual Hacking," *J. Tek. Inform. Univ. Sam Ratulangi*, vol. 11, no. 1, 2017, doi: 10.35793/jti.11.1.2017.16969.
- [5] R. Arlando Saragih, "Pengenalan Wajah Menggunakan Metode Fisherface," *J. Tek. Elektro*, vol. 7, no. 1, pp. 50-61, 2007, doi: 10.9744/jte.7.1.50-62.
- [6] R. Amat, J. Y. Sari, and I. P. Ningrum, "Implementasi Metode Local Binary Patterns Untuk Pengenalan Pola Huruf Hiragana Dan Katakana Pada Smartphone," *JUTI J. Ilm. Teknol. Inf.*, vol. 15, no. 2, p. 152, 2017, doi: 10.12962/j24068535.v15i2.a612.
- [7] Suendri, "Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan)," *J. Ilmu Komput. dan Inform.*, vol. 3, no. 1, pp. 1-9, 2018.
- [8] P. Pt and A. P. M. Rent, "No Title," vol. 2, no. 2, pp. 64-77, 2018.
- [9] A. Andrew, J. L. Buliali, and A. Y. Wijaya, "Deteksi Kecepatan Kendaraan Berjalan di Jalan Menggunakan OpenCV," *J. Tek. ITS*, vol. 6, no. 2, 2017, doi: 10.12962/j23373539.v6i2.23489.
- [10] B. Cahyono, "Penggunaan Software Matrix Laboratory (Matlab) Dalam Pembelajaran Aljabar Linier," *Phenom. J. Pendidik. MIPA*, vol. 3, no. 1, p. 45, 2016, doi: 10.21580/phen.2013.3.1.174.