BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi sistem

Implementasi merupakan tahap dimana sistem siap untuk dioperasikan. Hal ini dilakukan setelah penulisan kode program, uji sub sistem dan penggabungan masing-masing sub sistem. Pada tahap implementasi sistem ini, diharapkan sistem yang telah dirancang siap untuk dioperasikan pada keadaan yang sebenarnya, sehingga akan diketahui apakah sistem yang dibuat benar-benar dapat menghasilkan tujuan yang diharapkan.

Lingkungan implementasi sistem ada 2 yaitu, lingkungan perangkat keras dan lingkungan perangkat lunak. Kedua lingkungan implementasi ini dihrapkan dapat memenuhi persyaratan agar sistem yang dibangun dapat dijalankan dengan baik.

4.1.1 Implementasi Perangkat Keras

Sistem atau aplikasi dioperasikan dengan spesifikasi minimum perangkat keras yang digunakan mempunyai spesifikasi sebagai berikut :

Tabel 4.1Spesifikasi Minimum Perangkat Keras

Spesifikasi Minimum Perangkat Keras	
Processor	Intel Pentium 166 Mhz atau yang setara
RAM	256 MB
VGA	VRAM 8 MB
Harddisk	Ruang kosong 500 MB
Webcam	Dengan Parameter 1. *Resolution up to* 5 *megapixel*

	2. *Video Mode* 8 *bit or higher* 3. *Interface USB Port* 4. *Transmisison Rate* : 640 X 480 = 30 *frame/second*

4.1.2 Implementasi Perangkat Lunak

Sistem dioperasikan dengan spesifikasi minimum perangkat lunak yang digunakan mempunyai spesifikasi sebagai berikut :

Tabel 4.2 Spesifikasi Minimum Perangkat Lunak

Spesifikasi Minimum Perangkat Lunak	
Sistem Operasi	Microsoft Windows 2000 / XP / Vista / 7
Bahasa Pemrograman	Mathwork Matlab 7.1
Compiler	Microsoft C++ 6.0
Database	Mysql

4.1.3 Instalasi *Hardware*

Aplikasi perbandingan algoritma pengenalan wajah dirancang menggunakan media *input* yang berupa *webcam*. Pada pengembangan aplikasi ini *webcam* yang digunakan adalah *webcam* dengan tipe *Logitech QuickCam* HD 720p. *Webcam* dihubungkan dengan komputer melalui port USB.

Sebelum digunakan, *driver* untuk *webcam* harus diistalisasi karena nantinya aplikasi akan melakukan koneksi ke *driver webcam* untuk memfungsikannya sebagai media *input*. Jika *driver webcam* sudah diinstalisasi maka tinggal menghubungkan *webcam* ke port USB dan *webcam* siap digunakan sebagai media *input*.

4.2 Implementasi Basis Data

Pembuatan basis data dilakukan dengan mengunakan aplikasi pemrograman MySQL. Implementasi basis data dalam bahasa SQL adalah sebagai berikut :

1. Pembuatan *Database*

```
   CREATE DATABASE `db_foto` ;
```

2. Tabel foto

```
CREATE TABLE `foto`
(
  `id` bigint(20) NOT NULL auto_increment,
  `nama` varchar(255) NOT NULL,
  `foto` varchar(255) NOT NULL,
   PRIMARY KEY  (`id`)
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

3. Tabel *training*

```
CREATE TABLE `training`
(
  `id` bigint(20) NOT NULL auto_increment,
  `index_foto` text NOT NULL,
  `vektor` text NOT NULL,
  `metode` text NOT NULL,
   PRIMARY KEY  (`id`)
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.3 Implementasi Antar Muka

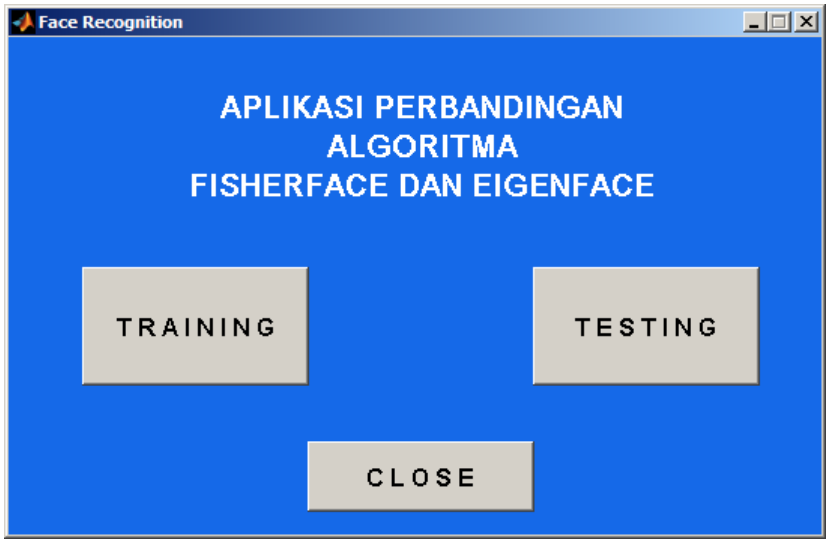
Tabel 4.3 Implementasi Antarmuka Aplikasi

No	Nama Form	Keterangan
1	Form Utama	Form utama dari Aplikasi
2	Form *Training*	Form untuk melakukan *training* terhadap citra pelatihan
3	*Form Testing*	Form untuk melakukan pengenalan atau deteksi.

Berikut adalah tampilan anta muka pengguna pada aplikasi perbandingan algoritma pengenalan wajah.

1. Form utama

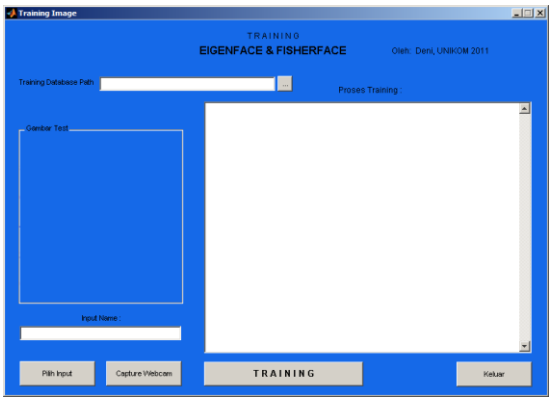
   Form ini terdapat tiga pilihan tombol yaitu tombol training, tombol testing dan tombol *close* untuk keluar dari aplikasi.



Gambar 4.1 Tampilan Form Utama

2. Form *training*

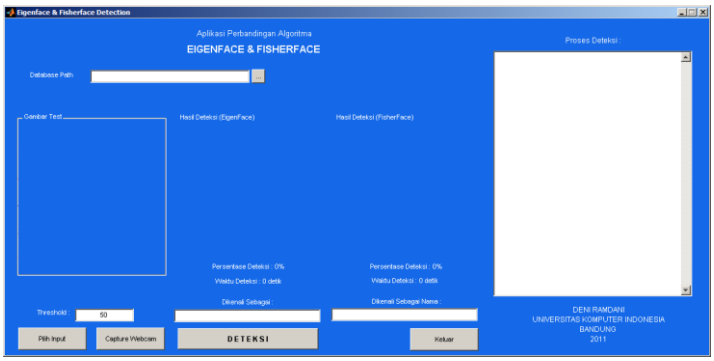
   Form ini digunakan untuk melakukan proses *training* terhadap citra pelatihan, sehingga akan didapatkan suatu pola yang digunakan untuk keperluan proses pengenalan dengan kedua metode. Form ini dilengkapi dengan menu, tombol, progress bar, jendela untuk menampilkan gambar dan jendela untuk menampilkan hasil proses *training.*



Gambar 4.2 Tampilan Form Training

3. Form *Testing*

   Form ini digunakan untuk melakukan pengujian pengenalan atau deteksi terhadap citra wajah yang akan dibandingkan dengan citra pelatihan yang sudah tersimpan.



Gambar 4.3 Tampilan Form Festing

4.4 Pengujian Sistem

Setelah melakukan tahapan implementasi, maka tahapan selanjutnya yaitu pengujian program yang telah dibuat. Sistem akan diuji dengan pengujian *alpha.*

Kedua algoritma akan diuji dengan menggunakan dokumen uji yang memiliki tiga buah citra wajah dari tiga orang sample dengan masing-masing memiliki lima pose yang berbeda. Pengukuran hasil pengujian adalah persentase akurasi dan waktu yang dibutuhkan masing-masing algoritma terhadap dokumen uji.

4.4.1 Pembentukan Data Awal

Untuk membentuk data awal yang akan dibuat sebagai *file* dalam komputer perlu dilakukan pengumpulan data yang kemudian akan dimasukkan kedalam sistem komputer. Data awal terdiri dari citra wajah yang di ambil dengan perangkat *webcam* dengan berbagai posisi.

4.4.1.1 Citra *Bitmap* sebagai Masukan Sistem Pengenalan Wajah

Sistem pengenalan wajah menerima masukan berupa citra bitmap dalam format *grey-level*. Kumpulan citra yang menjadi masukan pada proses pencarian matriks transformasi disebut *learning set*, dan N = jumlah citra *learning set*. Kumpulan citra yang menjadi masukan uji pengenalan disebut *test set*, dan T = jumlah citra *test set*. *Test set* dapat terdiri dari citra-citra pada learning set atau citra-citra lain.

4.4.1.2 Pemetaan Citra *Bitmap* menjadi Vektor Citra

Citra masukan dipetakan kedalam nilai-nilai bertipe *float*, yang merepresentasikan tingkat keabuan dari *pixel* pada posisi yang bersesuaian. Nilai

keabuan dari suatu citra kemudian disimpan dalam bentuk vektor baris, yang disebut vektor citra.

4.4.2 Kasus dan Hasil Pengujian *Alpha*

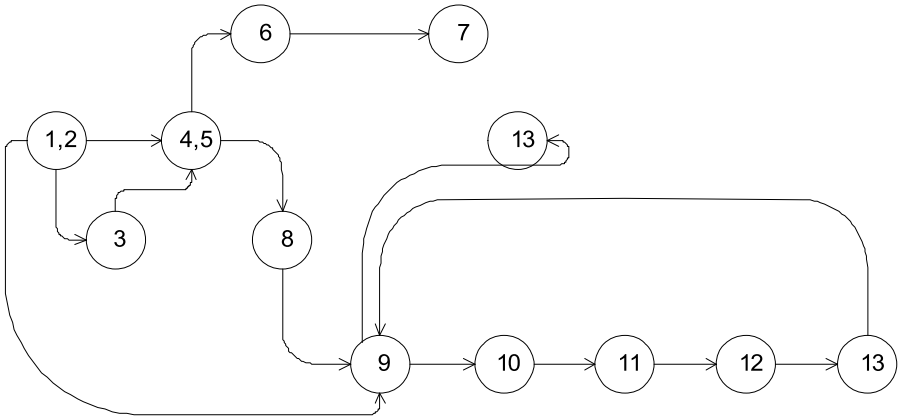
Pengujian *alpha* dilakukan dengan mengunakan uji coba *basis path* dengan menggunakan diagram alir dan menghitung kompleksitas siklomatis pada masing-masing proses. Kompleksitas siklomatis dihitung dengan menggunakan simpul predikat. Sehingga jumlah diagram alir harus sama dengan simpul predikat yang ditambah dengan satu.

4.4.2.1 Butir Pengujian

Pengujian yang dilakukan menggunakan pengujian *white box* yaitu dengan menguji setiap prosedur komputasi ataupun proses validasi yang harus dilewati untuk memastikan proses tersebut dapat berjalan secara efisien dan semua proses memiliki fungsi yang tepat guna dengan menggunakan sekumpulan dasar dari alur eksekusi.

A. Butir Pengujian Pendeteksi Kamera dan Pengambilan Citra

Penangkapan citra yang merupakan *input* yang diterima oleh sistem berupa citra wajah manusia dengan menggunakan perangkat keras *webcam* terlebih dahulu harus dilakukan pendeteksian kamera itu sendiri. Pendeteksian yang dilakukan dengan melakukan pendeteksian kamera 0 hingga 1 (hanya 2 kamera) jika masing-masing tidak terdeteksi maka sistem tidak dapat berjalan. Pengujian untuk prosedur pengambilan citra dan juga pendeteksian kamera dapat dilihat pada gambar 4.4.



Gambar 4.4 Diagram Alir Proses Pengambilan Citra

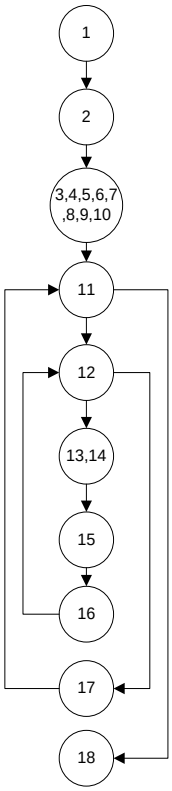
Jumlah dari *cyclomatix complexity* atau V(G) dari diagram alir pada gambar 4.4 adalah sebagai berikut :

Tabel 4.4 Perhitungan Jalur Independen dan Kompleksitas Siklomatis.

Jalur independen		Perhitungan siklomatis		Jumlah jalur grafik=siklomatis
No.	Jalur	Simpul predikat (P)	V(G)=P+1	
1	1-2-3-4-5-8-9-10-11-12-13-9-13	3	4	[√] diterima [ ] ditolak
2	1-2-4-5-8-9-10-11-12-13-9-13			
3	1-2-3-4-5-6-7			
4	1-2-4-5-6-7			

B. Butir Pengujian Proses Penskalaan

Setelah didapat input berupa citra dari hasil *capture* maka citra tersebut harus dirubah skala atau ukurannya untuk mempermudah dalam perhitungan. Pengujian untuk prosedur proses komputasi untuk penskalaan dapat dilihat pada gambar 4.5 dibawah ini :



Gambar 4.5 Diagram Alir Proses Penskalaan

Jumlah dari *cyclomatix complexity* atau V(G) dari diagram alir pada gambar 4.5 adalah sebagai berikut :

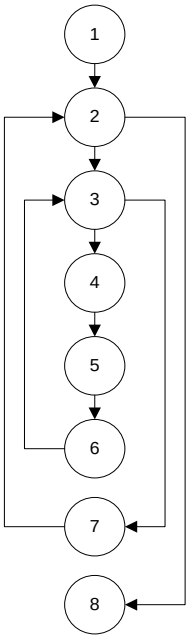
Tabel 4.5 Perhitungan jalur independen dan kompleksitas siklomatis.

Jalur independen		Perhitungan siklomatis		Jumlah jalur grafik=siklomatis
No.	Jalur	Simpul Predikat (P)	V(G)=P+1	
1	1-2-3,4,5,6,7,8,9,10-11-12-13,14-15-16-12-17-11-18	2	3	[√] diterima [ ] ditolak
2	1-2-3,4,5,6,7,8,9,10-11-18			
3	1-2-3,4,5,6,7,8,9,10-11-12-17-11-18			

C. Butir Pengujian Untuk Proses *Grayscale Image*

Setelah didapat input berupa citra dari hasil *capture* maka citra tersebut perlu dilakukan proses *grayscale* untuk menyederhanakan kompleksitas warna.

Pengujian untuk prosedur proses komputasi untuk *grayscale image* dapat dilihat pada gambar 4.6 dibawah ini :



Gambar 4.6 Diagram Alir Proses *Grayscale*

Jumlah dari *cyclomatix complexity* atau V(G) dari diagram alir pada gambar 4.6 adalah sebagai berikut :

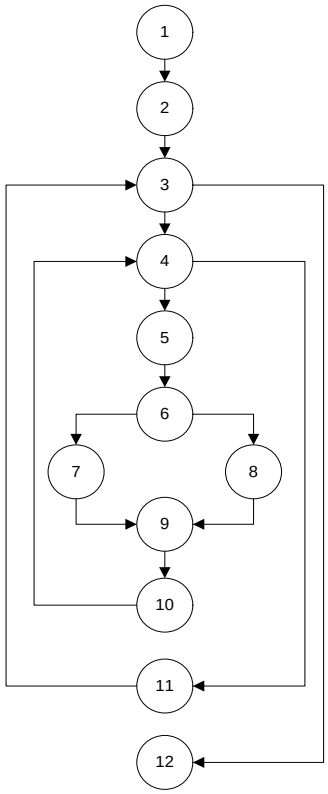
Tabel 4.6 Perhitungan Jalur Independen dan Kompleksitas Siklomatis.

Jalur independen		Perhitungan siklomatis		Jumlah jalur grafik=siklomatis
No.	Jalur	Simpul predikat (P)	V(G)=P+1	
1	1-2-3-4-5-6-3-7-2-8	2	3	[√] diterima [ ] ditolak
2	1-2-8			
3	1-2-3-7-2-8			

D. Butir Pengujian untuk Proses Binerisasi

Setelah didapat *image* yang ter-*grayscale* maka citra tersebut perlu dirubah ke dalam bentuk citra hitam putih untuk lebih menyedernakan kompleksitas

warna. Pengujian untuk prosedur proses komputasi untuk binerisasi dapat dilihat pada gambar 4.7 dibawah ini :



Gambar 4.7 Diagram Alir Proses Binerisasi

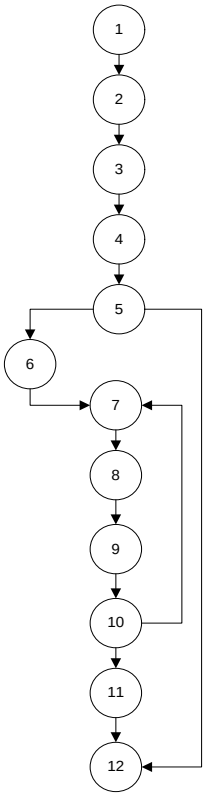
Jumlah dari *cyclomatix complexity* atau V(G) dari diagram alir pada gambar 4.7 adalah sebagai berikut :

Tabel 4.7 Perhitungan Jalur Independen dan Kompleksitas Siklomatis.

Jalur independen		Perhitungan siklomatis		Jumlah jalur grafik=siklomatis
No.	Jalur	Simpul predikat (P)	V(G)=P+1	
1	1-2-3-4-5-6-7-9-10-4-11-3-12	3	4	[√] diterima [ ] ditolak
2	1-2-3-12			
3	1-2-3-4-11-3-12			
4	1-2-3-4-5-6-8-9-10-4-11-3-12			

E. Perhitungan dengan metode *Eigenface*

Setelah didapat vektor ciri dari *image* selanjutnya adalah melakukan pengujian dengan metode *Eigenface*. Pengujian untuk prosedur proses deteksi dengan metode *eigenface* dapat dilihat pada gambar 4.8 dibawah ini :



Gambar 4.8 Diagram Alir Deteksi Dengan Metode *Eigenface*

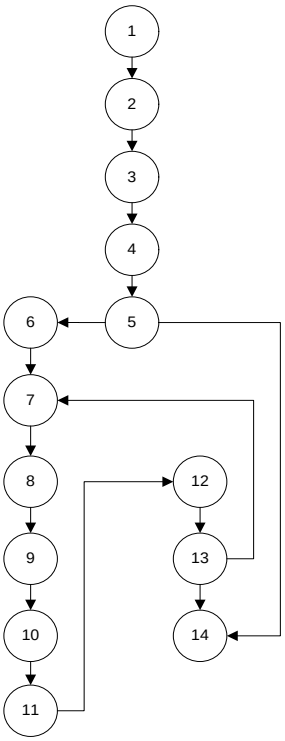
Jumlah dari *cyclomatix complexity* atau V(G) dari diagram alir pada gambar 4.8 adalah sebagai berikut :

Tabel 4.8 Perhitungan Jalur Independen dan Kompleksitas Siklomatis.

Jalur independen		Perhitungan siklomatis		Jumlah jalur grafik=siklomatis
No.	Jalur	Simpul predikat (P)	V(G)=P+1	
1	1-2-3-4-5-6-7-9-10-4-11-3-12	2	3	[√] diterima [ ] ditolak
2	1-2-5-12			
3	1-2-3-4-5-6-7-9-10-7-8-9-10-11-3-12			

F. Perhitungan dengan metode *Fisherface*

Setelah perhitungan dengan metode *eigenface* selesai selanjutnya adalah melakukan pengujian dengan metode *fisherface*. Pengujian untuk prosedur proses deteksi dengan metode *fisherface* dapat dilihat pada gambar 4.8 dibawah ini :



Gambar 4.9 Diagram Alir Deteksi Dengan Metode *Fisherface*

Jumlah dari *cyclomatix complexity* atau V(G) dari diagram alir pada gambar 4.8 adalah sebagai berikut :

Tabel 4.9 Perhitungan Jalur Independen dan Kompleksitas Siklomatis.

Jalur independen		Perhitungan siklomatis		Jumlah jalur grafik=siklomatis
No.	Jalur	Simpul predikat (P)	V(G)=P+1	
1	1-2-3-4-5-6-7-9-10-4-11-3-12-13-14	2	3	[√] diterima [ ] ditolak
2	1-2-5-14			
3	1-2-3-4-5-6-7-9-10-11-12-13-7-8-9-10-11-12-13-14			

4.4.2.2 Kesimpulan Hasil Pengujian *Alpha*

Berdasarkan hasil pengujian *Alpha* yang telah dilakukan dapat disimpulkan bahwa aplikasi sudah berjalan cukup maksimal, tetapi tidak menutup kemungkinan dapat terjadi kesalahan suatu saat, pada saat aplikasi digunakan, sehingga membutuhkan proses *maintenance* untuk lebih mengetahui kekurangan dari aplikasi.

4.4.3 Sampel Citra Uji Coba

Agar hasil pengujian yang dilakukan mendapatkan hasil maksimal, maka kondisi lingkungan seperti pencahayaan harus dibuat konstan hal ini dikarenakan ketika sebuah citra dirubah ke dalam citra hitam putih maka kompleksitas warna hanya dua. Jika citra wajah tersebut berada pada pencahayaan rendah maka yang ditangkap akan lebih banyak citra warna hitam. Hal ini menyebabkan hasil pengenalan terkadang menjadi lebih sulit. Serta ketika pengambilan citra uji posisi wajah tidak boleh berubah, karena akan menyebabkan hasil penangkapan citra akan menjadi kabur.

Sampel citra yang digunakan dalam uji coba terdiri dari dua macam sampel yaitu :

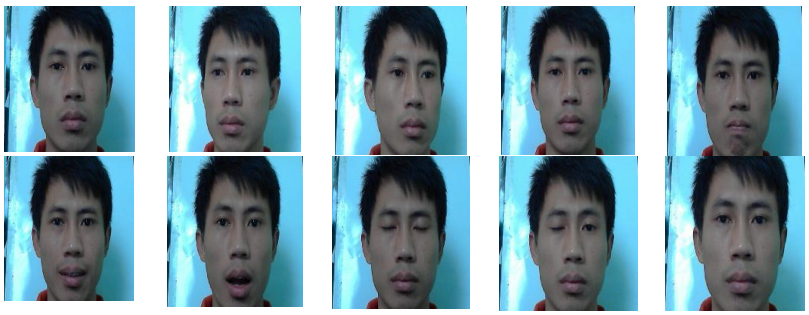
<u>Kelompok A</u>

Terdiri dari 10 kelas citra wajah. Masing-masing kelas terdiri dari 10 citra dari satu orang yang sama, dalam kondisi-kondisi berikut :

1. Tidak memakai kacamata
2. Menggunakan 3 variasi pencahayaan, yaitu pencahayaan dari arah depan, dari samping kanan dan samping kiri.

3. Menggunakan 5 ekpresi wajah, yaitu sedih, tidur, terkejut, tertawa dan berkedip.

Kelas citra yang digunakan untuk pengujian Kelompok A dapat dilihat pada gambar 4.10 dibawah ini :



Gambar 4.10 Kelas Uji Kelompok A

Kelompok B

Terdiri dari 10 kelas citra wajah. Masing-masing kelas terdiri dari 10 citra dari dua orang, dalam kondisi-kondisi berikut :

1. Tidak memakai kacamata
2. Menggunakan 3 variasi pencahayaan, yaitu pencahayaan dari arah depan, dari samping kanan dan samping kiri
3. Ekspresi standar atau tanpa ekspresi

Kelas citra yang digunakan untuk pengujian Kelompok B dapat dilihat pada gambar 4.11 dibawah ini :



Gambar 4.11 Kelas Uji Kelompok B

4.4.4 Rancangan Uji Coba

Uji coba yang dilakukan dalam tugas akhir ini terbagi menjadi beberapa kasus, yaitu :

a. Kasus I

Learning set : 10 citra Kelompok A, yaitu citra wajah tanpa kacamata

Test set : 5 citra yang sama dengan *learning set*

b. Kasus II

Learning set : 10 citra Kelompok A, sama dengan *learning set* Kasus I

Test set : 5 citra Kelompok A, yaitu citra wajah dalam 5 ekpresi → sedih, tidur, terkejut, senang dan mata berkedip

c. Kasus III

Learning set : seluruh citra Kelompok B

Test set : 5 citra yang sama dengan *learning set* citra Kelompok B

Asumsi : Untuk metode Fisherface, 1 kelas *training set* minimal terdiri dari 2 citra wajah. Hal ini mengingat persamaan (2.30), dimana terjadi pengurangan antara x_i dengan rata-rata kelas X_j, dimana $x_i \in X_j$. Apabila anggota kelas $X_j = 1$, maka hasil pengurangan adalah 0.

4.5 Keluaran Sistem Pengenalan Wajah

Analisa terhadap akurasi pengenalan dilakukan dengan melihat keluaran dari sistem pengenalan wajah.

4.5.1 Tingkat Kemiripan dengan *Eucledean Distance*

Untuk mengukur tingkat akurasi atau kemiripan digunakan perhitungan nilai jarak terpendek dengan *Eucledean Distance*. Hal ini dikarenakan jarak minimum citra uji akan dibandingkan dengan jarak citra pelatihan, jika didapatkan jarak pelatihan paling kecil atau nilai *eucledean distance* minimum, maka itulah citra yang akan dikenali ketika proses pengujian atau *testing.*

Sedangkan untuk mengukur persentase akurasi, didapatkan dari pembagian nilai citra pelatihan dibagi dengan nilai citra uji lalu dikalikan 100. Jika nilai *distance minimum* citra pelatihan lebih besar dari citra uji maka persentase akurasi akan bernilai nol.

Nilai dari perhitungan persentase akan digunakan sebagai persentase akurasi atau toleransi kesalahan dalam pengenalan. Ketika nilai citra uji lebih kecil dari ambang *tresshold* yang telah ditentukan maka perhitungan bernilai 0 maka citra tersebut tidak dikenali.

Untuk data yang akan digunakan untuk proses pengujian dan *training* adalah data citra wajah yang sama. Karena pada proses *training* citra yang digunakan hanya akan diambil pola dan nilai *distance* dari citra tersebut.

4.5.2 Uji Coba dan Analisa Sistem Pengenalan Wajah

Tabel-tabel hasil uji coba dengan *Eucledean Distance* menunjukkan akurasi pengenalan sistem untuk nilai *threshold* yang telah ditetapkan. Besar nilai *threshold* menentukan dimensi ruang citra yang terbentuk. Tabel-tabel hasil uji coba dengan kondisi pencahayaan tinggi dan pencahayaan rendah menunjukan perbandingan akurasi dan kecepatan deteksi terhadap *image test* dengan *image training set*.

4.5.2.1 Kasus I

Pengujian pada kasus I yaitu pengujian terhadap satu orang dengan nilai *thresshold* yang berbeda, tujuannya adalah untuk mendapatkan nilai *thresshold* yang paling sesuai untuk pengenalan citra dengan menggunakan *eucledean distance*.

Tabel 4.10 Hasil Uji Coba Pengenalan Kasus I- *Eucledean Distance*

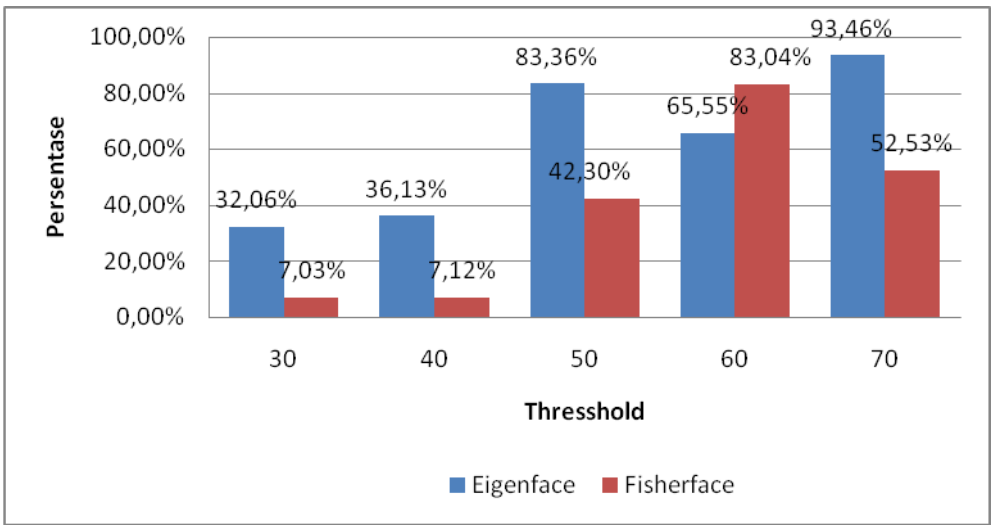
KASUS I							
T = 10	***Threshold***	**Eigenface**		**Ket.**	**Fisherface**		**Ket.**
N = 5		**Akurasi (%)**	**Waktu(s)**		**Akurasi (%)**	**Waktu (s)**	
	30	32,06%	0,639	Tidak	7,03%	0,687	Tidak
	40	36,13%	0,129	Tidak	7,12%	0,546	Tidak
	50	83,36%	0,639	Dikenali	42,30%	0,546	Tidak
	60	65,55%	0,624	Dikenali	83,04%	0,577	Dikenali
	70	93,46%	0,717	Dikenali	52,53%	0,577	Dikenali

Keterangan :

T : *Training Set*

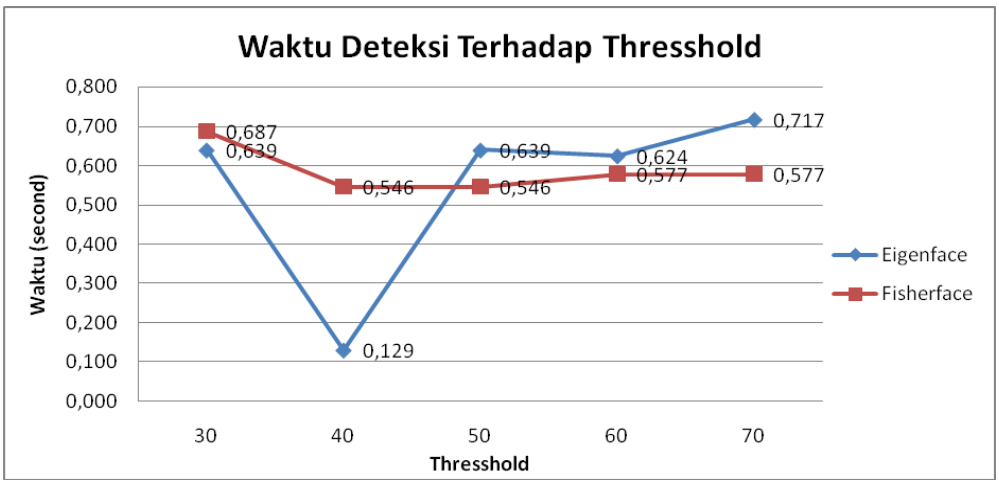
N : Sample Uji

Gambar 4.12 dibawah ini adalah grafik yang menunjukan persentase hasil pengenalan masing-masing algoritma dalam mengenali pola wajah pada setiap dokumen uji.



Gambar 4.12 Grafik Tingkat Akurasi Pengenalan Kasus I - Eucledean Distance

Gambar 4.13 dibawah ini adalah grafik yang menunjukan waktu yang dibutuhkan masing-masing algoritma dalam mengenali pola wajah pada masing-masing dokumen uji. Sumbu y adalah satuan waktu dalam detik.



Gambar 4.13 Grafik Waktu Pengenalan Kasus 1 - Eucledean Distance

Analisis Pengujian Kasus I :

Percobaan ini bertujuan untuk melihat akurasi pengenalan sistem terhadap nilai batas ambang *thresshold.* Citra wajah yang digunakan untuk pengujian adalah sama dengan citra pada pelatihan. Selain itu, akurasi pengenalan sistem terhadap data yang sudah dilatihkan, tampak pada percobaan ini.

Dari Tabel 4.10 tampak bahwa baik *Eigenface* maupun *Fisherface* memiliki pengenalan yang baik untuk data yang telah dilatihkan pada kondisi *thresshold* 60 dan 70. Pada percobaan ini metode *Eigenface* memiliki akurasi pengenalan yang lebih baik dibandingkan metode *Eigenface*.

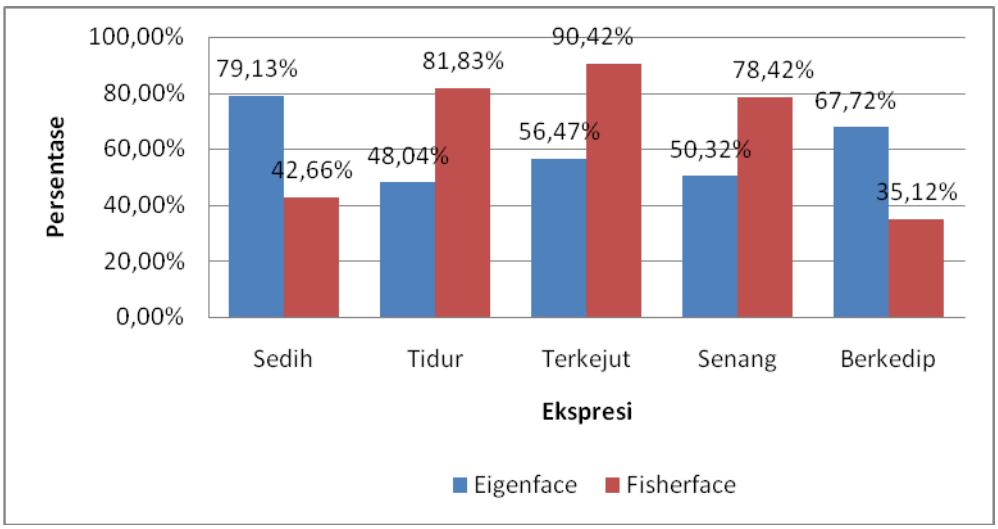
4.5.2.2 Kasus II

Pengujian pada kasus II yaitu pengujian terhadap satu orang dengan berbagai ekspresi wajah, dengan menggunakan nilai *thresshold* yang sudah didapatkan pada pengujian kasus I yang dianggap sesuai untuk pengenalan citra.

Tabel 4.11 Hasil Uji Coba Pengenalan Kasus II- *Eucledean Distance*

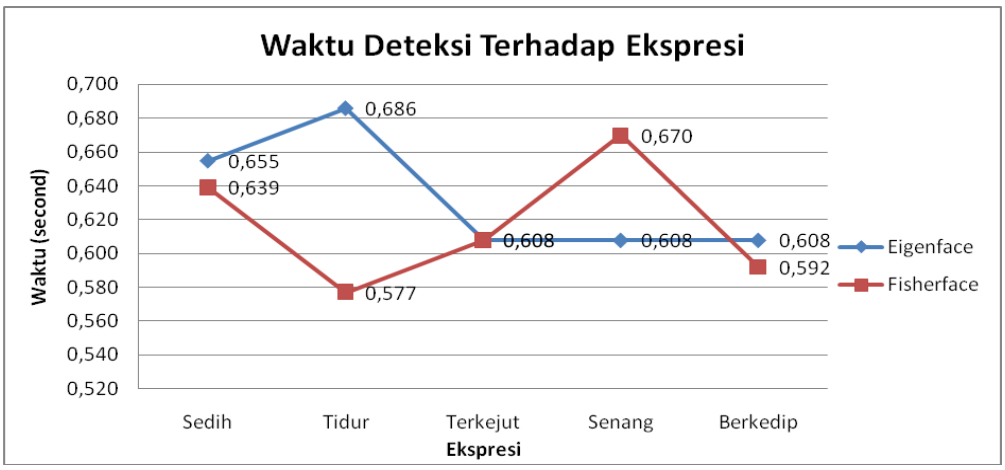
KASUS II						
Ekspresi	**Eigenface**		**Ket.**	**Fisherface**		**Ket.**
	Akurasi (%)	**Waktu (s)**		**Akurasi (%)**	**Waktu (s)**	
Sedih	79,13%	0,655	Dikenali	42,66%	0,639	Tidak
Tidur	48,04%	0,686	Tidak	81,83%	0,577	Dikenali
Terkejut	56,47%	0,608	Dikenali	90,42%	0,608	Dikenali
Senang	50,32%	0,608	Dikenali	78,42%	0,670	Dikenali
Berkedip	67,72%	0,608	Dikenali	35,12%	0,592	Tidak

Gambar 4.14 dibawah ini adalah grafik yang menunjukan persentase hasil pengenalan masing-masing algoritma dalam mengenali pola wajah pada setiap dokumen uji.



Gambar 4.14 Grafik Tingkat Akurasi Pengenlan Kasus 1I - Eucledean Distance

Gambar 4.15 dibawah ini adalah grafik yang menunjukan waktu yang dibutuhkan masing-masing algoritma dalam mengenali pola wajah pada masing-masing dokumen uji. Sumbu y adalah satuan waktu dalam detik.



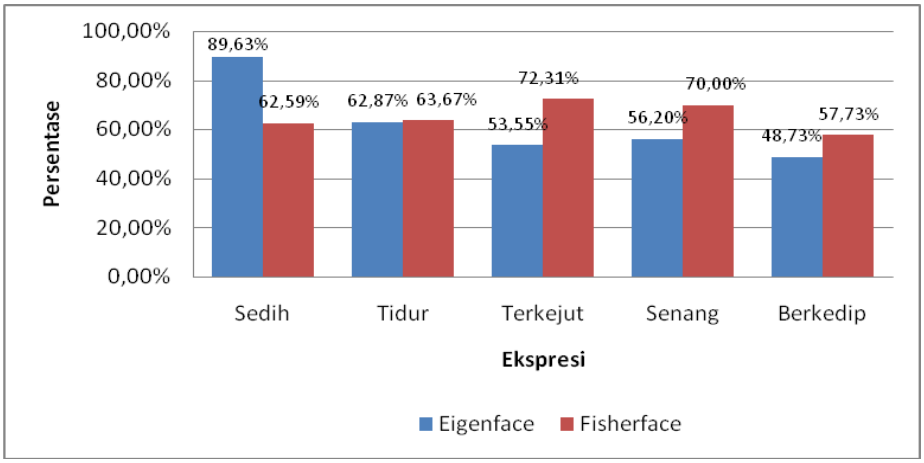
Gambar 4.15 Grafik Waktu Pengenlan Kasus 1I - Eucledean Distance

c. Metode pengenalan : *Eucledean Distance* dengan *Thresshold* 70

Tabel 4.12 Hasil Uji Coba Pengenalan Kasus II- *Eucledean Distance*

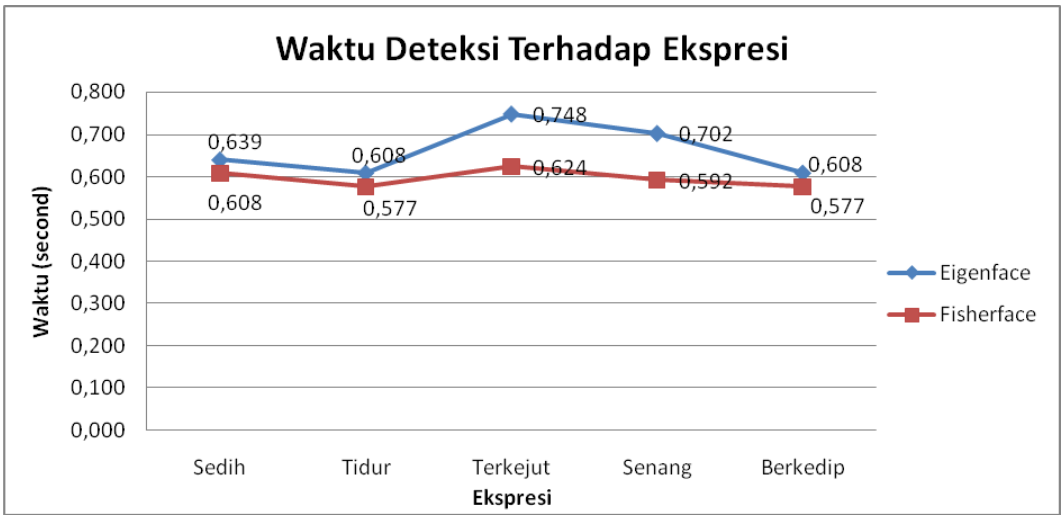
Kasus II						
Ekspresi	Eigenface		Ket.	Fisherface		Ket.
	Akurasi (%)	Waktu (s)		Akurasi (%)	Waktu (s)	
Sedih	89,63%	0,639	Dikenali	62,59%	0,608	Dikenali
Tidur	62,87%	0,608	Dikenali	63,67%	0,577	Dikenali
Terkejut	53,55%	0,748	Dikenali	72,31%	0,624	Dikenali
Senang	56,20%	0,702	Dikenali	70,00%	0,592	Dikenali
Berkedip	48,73%	0,608	Dikenali	57,73%	0,577	Dikenali

Gambar 4.16 dibawah ini adalah grafik yang menunjukan persentase hasil pengenalan masing-masing algoritma dalam mengenali pola wajah pada setiap dokumen uji.



Gambar 4.16 Grafik Tingkat Akurasi Pengenlan Kasus II - Eucledean Distance

Gambar 4.17 dibawah ini adalah grafik yang menunjukan waktu yang dibutuhkan masing-masing algoritma dalam mengenali pola wajah pada masing-masing dokumen uji. Sumbu y adalah satuan waktu dalam detik.



Gambar 4.17 Grafik Waktu Pengenlan Kasus 1I - Eucledean Distance

Analisis Pengujian Kasus II :

Percobaan ini bertujuan untuk melihat akurasi pengenalan sistem terhadap nilai batas ambang *thresshold* yang dianggap sesuai untuk pengujian dan sudah diuji pada Kasus Pengujian I didapatkan nilai *thresshold* 60 dan 70. Citra wajah yang digunakan sama dengan pelatihan tetapi dengan menggunakan beberapa ekspresi wajah. Selain itu, akurasi pengenalan sistem terhadap data yang sudah dilatihkan, tampak pada percobaan ini.

Dari Tabel 4.11 dan 4.12, tampak bahwa baik *Eigenface* maupun *Fisherface* memiliki pengenalan yang baik untuk data yang telah dilatihkan. Pada percobaan ini metode *Fisherface* memiliki akurasi pengenalan yang lebih baik dibandingkan metode *Eigenface*, bahkan untuk dimensi ciri yang lebih kecil. Hal ini tampak baik pada Tabel 4.11 dan Tabel 4.12, untuk *threshold* = 60 dan 70. Fisherface dapat mencapai akurasi pengenalan hingga 90%.

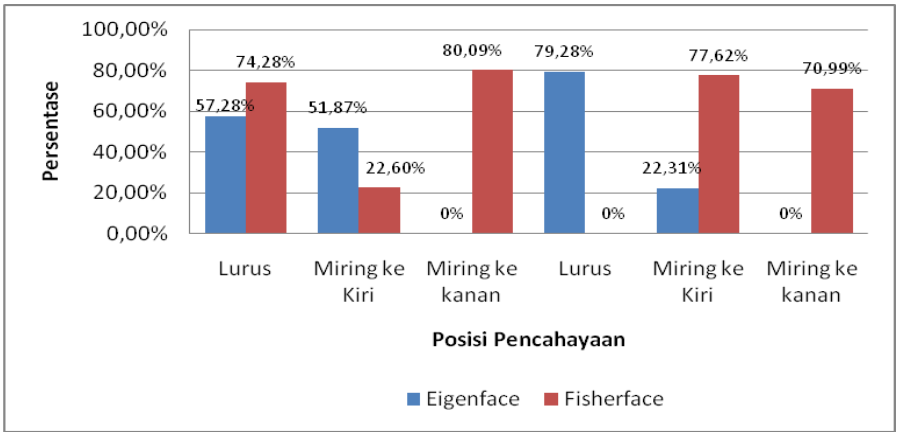
4.5.2.3 Kasus III

Untuk pengujian dengan kasus III, dilakukan kepada dua orang yang berbeda dengan masing-masing mempunyai citra pelatihan sebanyak 5 dengan berbagai posisi pencahayaan dengan menggunakan nilai *thresshold* 60. Sedangkan untuk pengujian menggunakan 3 posisi pencahayaan yang berbeda yaitu dari posisi lurus, miring ke kiri dan miring ke kanan. Untuk hasil pengujian dapat dilihat pada tabel 4.13 dibawah ini :

Tabel 4.13 Hasil Uji Coba Pengenalan Kasus III- *Eucledean Distance*

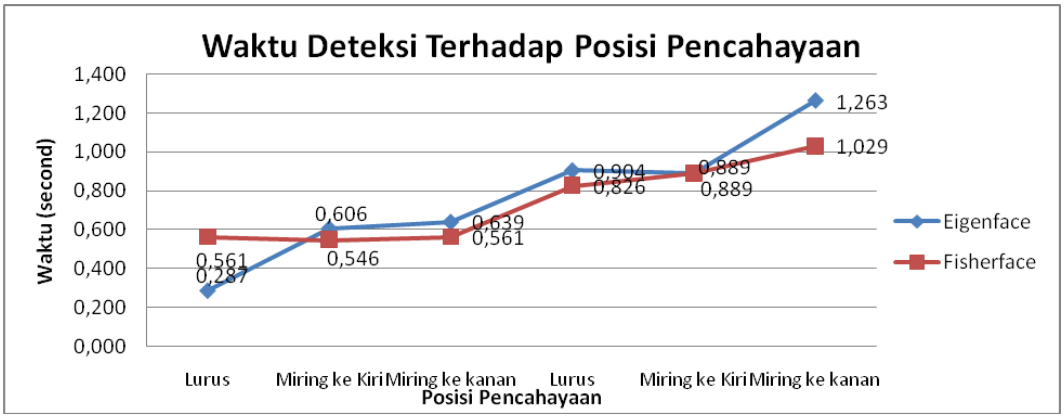
KASUS III							
Orang ke	**Pose**	**Eigenface**		**Ket.**	**Fisherface**		**Ket.**
		Akurasi (%)	**Waktu (s)**		**Akurasi (%)**	**Waktu (s)**	
1	Lurus	57.28%	0,287	Dikenali	74,28%	0,561	Dikenali
	Miring ke Kiri	51.87%	0,606	Dikenali	22,60%	0,546	Tidak
	Miring ke kanan	0%	0,639	Tidak	80,09%	0,561	Dikenali
2	Lurus	79.28%	0,904	Dikenali	0%	0,826	Tidak
	Miring ke Kiri	22.31%	0,889	Tidak	77,62%	0,889	Dikenali
	Miring ke kanan	0%	01,263	Tidak	70,99%	1,029	Dikenali

Gambar 4.18 dibawah ini adalah grafik yang menunjukan persentase hasil pengenalan masing-masing algoritma dalam mengenali pola wajah pada setiap dokumen uji.



Gambar 4.18 Grafik Tingkat Akurasi Pengenlan Kasus II1 - Eucledean Distance

Gambar 4.19 dibawah ini adalah grafik yang menunjukan waktu yang dibutuhkan masing-masing algoritma dalam mengenali pola wajah pada masing-masing dokumen uji. Sumbu y adalah satuan waktu dalam detik.



Gambar 4.19 Grafik Waktu Pengenlan Kasus 1II - Eucledean Distance

Analisis Pengujian Kasus III :

Percobaan ini bertujuan untuk melihat akurasi pengenalan sistem terhadap nilai batas ambang *thresshold.* Citra wajah yang digunakan untuk pengujian sama dengan data yang sama dengan citra pelatihan tetapi dengan menggunakan

beberapa citra uji dari orang yang berbeda. Selain itu, akurasi pengenalan sistem terhadap data yang sudah dilatihkan, tampak pada percobaan ini.

Dari Tabel 4.13, tampak bahwa baik *Eigenface* maupun *Fisherface* memiliki pengenalan yang baik untuk data yang telah dilatihkan. Pada percobaan ini metode *Fisherface* memiliki akurasi pengenalan yang lebih baik dibandingkan metode *Eigenface*, bahkan untuk dimensi ciri yang lebih kecil. Hal ini tampak baik pada Tabel 4.13, untuk *threshold* = 60. *Fisherface* dapat mencapai akurasi pengenalan hingga 80%.

Uji coba dengan beberapa pose pengujian terhadap 2 orang dengan masing-masing memiliki 5 pose pelatihan. Pengaruh tresshold terhadap hasil pengenalan sangat berpengaruh disamping pengaruh pencahayaan yang harus konstan. Terbukti dari hasil pengujian diatas bahwa pose lurus ke depan dengan pencahayaan tetap atau cukup memberikan hasil yang maksimal.

4.5.3 Kesimpulan Pengujian

Berdasarkan hasil pengenalan, algoritma *fisherface* dapat mengenali citra wajah lebih baik dibandingkan dengan algoritma *eigenface.* Hal ini terlihat pada hasil pengenalan yang telah diuji diatas, yang mengilustrasikan persentase hasil pengenalan kedua algoritma terhadap citra pelatihan atau dokumen uji.

Berdasarkan perhitungan kompleksitas kedua algoritma, algoritma *fisherface* memiliki kompleksitas yang lebih rumit dibandingkan dengan algoritma *eigenface*.

Algoritma *fisherface* membutuhkan waktu yang lebih singkat dibandingkan dengan algoritma *eigenface* dalam mengenali citra karakter yang terlihat pada pengujian yang telah dilakukan.

Dari berbagai hasil uji coba di atas, tampak bahwa tidak ada nilai *threshold* khusus, yang dapat menghasilkan akurasi pengenalan terbaik untuk suatu kasus. Besarnya nilai *threshold* menentukan banyaknya informasi citra yang akan diikutsertakan dalam klasifikasi. Bertambahnya nilai *threshold* memiliki 2 kemungkinan dampak, yaitu :

- Betambahnya informasi yang membuat suatu kelas semakin berbeda dengan kelas lainnya, sehingga pengenalan akan kelas tersebut semakin baik
- Bertambahnya informasi yang membuat suatu kelas semakin mirip dengan kelas lain, sehingga pengenalan akan kelas tersebut semakin buruk

Tabel 4.14 adalah hasil pengukuran kedua algoritma berdasarkan hasil pengenalan, kompleksitas, pengembangan, dan waktu.

Tabel 4.14 Hasil Pengukuran Kedua Algoritma.

No.	Pengukuran	*Eigenface*	*Fisherface*
1.	Hasil pengenalan	Buruk	Baik
2.	Proses	Lebih sedikit	Lebih banyak
3.	Kompleksitas	Lebih sederhana	Lebih rumit
4.	Pengembangan	Sulit dikembangkan	Bisa dikembangkan
5.	Waktu	Lebih lama	Lebih singkat

Berdasarkan tabel 4.14, algoritma *fisherface* lebih unggul dibandingkan dengan algoritma *eigenface*. Keunggulan tersebut terlihat pada hasil pengukuran kedua algoritma berdasarkan hasil pengenalan, pengembangan, dan waktu.

Dengan percobaan yang telah dilakukan didapatkan beberapa analisis awal sebagai berikut:

1. Perubahan pose tidak terlalu mempengaruhi akurasi pengenalan, yang terlihat pada persentase pengenalan wajah.
2. Pengenalan pada pencahayaan yang normal ternyata memberikan hasil yang lebih baik, ini disebabkan pada pencahayaan yang tinggi, detail wajah yang lebih jelas, sehinggga kompleksitas warna wajah lebih tinggi.

Dari sini bisa ditentukan hasil analisisnya, yaitu bahwa keberhasilan pengenalan dipengaruhi oleh beberapa hal, diantaranya :

1. Jarak antar subjek dan kamera.

   Untuk pengenalan yang baik, jarak antar subjek dan kamera saat pengambilan wajah dan proses deteksi sebaiknya dibuat konstan.

2. Pencahayaan.

   Pencahayaan yang terlalu tinggi akan membuat detail gambar yang lebih rumit, mengakibatkan matriks dari gambar menjadi lebih kompleks, sehingga pengenalan menjadi lebih sensitive dan menurunkan kualitas pengenalan.

3. Pose Wajah.

   Untuk pengenalan pose wajah dari berbagai posisi tidak terlalu mempengaruhi hasil pengenalan. Tetapi jika distorsi pose (miring ke kanan atau ke kiri) terlalu banyak sudut kemiringan akan menurunkan kualitas pengenalan.

Berdasarkan hasil pengujian dengan kasus *sample* uji yang telah dilakukan memberikan kesimpulan bahwa pada proses masih memungkinkan untuk terjadinya kesalahan pada sintaks tetapi secara fungsional sistem sudah dapat menghasilkan *output* yang diharapkan.