

**PENGENALAN WAJAH MENGGUNAKAN METODE
LINEAR DISCRIMINANT ANALYSIS**

SKRIPSI

Oleh:
AKBAR WIDIANTO P
0810962001-96

UNIVERSITAS BRAWIJAYA



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2011**



**PENGENALAN WAJAH MENGGUNAKAN METODE
LINEAR DISCRIMINANT ANALYSIS**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar
Sarjana Ilmu Komputer

Oleh:

**AKBAR WIDIANTO P
0810962001-96**

UNIVERSITAS BRAWIJAYA



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2011**



LEMBAR PENGESAHAN SKRIPSI

**PENGENALAN WAJAH MENGGUNAKAN METODE
LINEAR DISCRIMINANT ANALYSIS**

Oleh :

Akbar Widianto P
0810962001-96

Setelah dipertahankan di depan Majelis Penguji
Pada tanggal 14 Maret 2011

dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Komputer dalam bidang Ilmu Komputer

Pembimbing I,

Pembimbing II,

Dewi Yanti L., S.Kom, M.Kom
NIP. 198111162005012004

Edy Santoso, S.Si., M.Kom
NIP. 19740414200321004

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Abdul Rouf Alghofari, M.Sc
NIP.196709071992031001



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

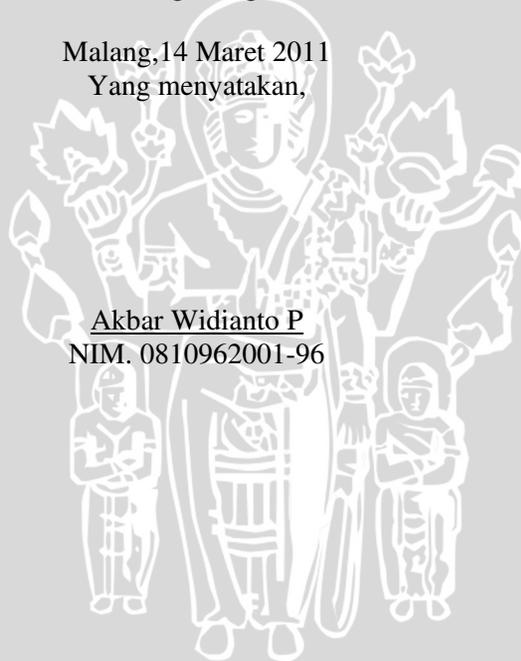
Nama : Akbar Widiyanto P
NIM : 0810962001-96
Jurusan : Matematika
Program Studi : Ilmu Komputer
Penulis skripsi berjudul : Pengenalan Wajah Menggunakan Metode
Linear Discriminant Analysis

Dengan ini menyatakan bahwa :

1. Isi dari Skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Skripsi ini.
2. Apabila dikemudian hari ternyata Skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 14 Maret 2011
Yang menyatakan,



Akbar Widiyanto P
NIM. 0810962001-96



ABSTRAK

Beberapa penelitian tentang *face recognition* pernah dibuat. Salah satunya adalah PCA (*Principal Component Analysis*). Hasil penelitian menunjukkan bahwa menggunakan metode PCA dapat mengenali wajah dengan tingkat keakuratan 74 %.

Pada skripsi ini akan dibuat *face recognition* menggunakan LDA (*Linear Discriminant Analysis*) dengan PCA (*Principal Component Analysis*) untuk reduksi dimensi. Metode LDA dipilih karena untuk *feature extraction* dan pengenalan wajah menggabungkan distribusi sampel dan mempertimbangkan informasi yang diskriminatif dari *within-class scatter matrix* dan *between-class scatter matrix*.

Hasil yang didapatkan berdasarkan penelitian ini adalah tingkat akurasi menggunakan database ORL yaitu 92% dan Yale 80%. Dengan tingkat akurasi ini, maka metode LDA ini cukup handal untuk diimplementasikan pada sistem identifikasi yang menggunakan pengenalan wajah yang memiliki latar yang gelap sehingga porsi wajah lebih penuh.

Kata kunci : *Linear Discriminant Analysis (LDA), feature extraction, within-class scatter matrix, between-class scatter matrix*



ABSTRACT

Some research on face recognition has ever made. One is the PCA (Principal Component Analysis). The results showed that using the PCA method to identify faces with 74% accuracy level.

This research will be made face recognition using LDA (Linear Discriminant Analysis) with PCA (Principal Component Analysis) for dimension reduction. LDA method was chosen due to feature extraction and face recognition combining the distribution of the sample and consider the discriminatory information from within-class scatter matrix and between-class scatter matrix.

Results obtained on the basis of this research is the level of accuracy using ORL database is 92% and 80% of Yale. With this accuracy level, the LDA method is reliable enough to be implemented in the identification face recognition system that the image of his face has a dark background so that the portion of the face and a little fuller.

Keywords : *Linear Discriminant Analysis (LDA), feature extraction, within-class scatter matrix, between-class scatter matrix*



KATA PENGANTAR

Puji syukur kehadiran Allah SWT, hanya dengan rahmat dan karunia yang telah diberikan kepada penulis, sehingga dapat menyelesaikan skripsi yang berjudul “*Pengenalan Wajah Menggunakan Metode Linear Discriminant Analysis*”.

Skripsi ini merupakan salah satu syarat untuk memenuhi persyaratan akademis untuk menyelesaikan studi di program Sarjana Ilmu Komputer Universitas Brawijaya. Pada kesempatan ini penulis mengucapkan banyak terima kasih atas segala bantuan dan dedikasi moral maupun material dalam rangka penyusunan skripsi ini.

1. Dewi Yanti Liliana, S.Kom., M.Kom dan Edy Santoso, S.Si., M.Kom., selaku dosen pembimbing yang telah membimbing dengan bijaksana dan sabar dalam membimbing dengan baik penyusunan skripsi ini.
2. Dr. Abdul Rouf Alghofari, M.Sc., selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya.
3. Drs. Marji, MT., selaku Ketua Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya.
4. Agus Wahyu Widodo, ST, selaku dosen pembimbing akademis Segenap Bapak dan Ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya.
5. Segenap staf dan karyawan Jurusan Matematika Universitas Brawijaya yang telah membantu penyusunan skripsi ini.
6. Ayah, Ibu, dan saudara-saudara, serta keluarga, terima kasih atas dukungan dan doanya.
7. Teman-teman SAP angkatan 2008, teman-teman Prodi Ilmu Komputer dan teman-teman lain yang selalu memberi dukungan dan doanya.
8. Seluruh pihak yang tidak dapat disebut secara langsung yang telah memberikan bantuan demi terselesaikannya skripsi ini.

Penulis menyadari bahwa masih banyak kekurangan dalam penulisan laporan ini yang disebabkan oleh keterbatasan kemampuan dan pengalaman. Oleh karena itu, Penulis sangat menghargai saran dan kritik yang sifatnya membangun demi perbaikan penulisan dan mutu isi penelitian ini.

Penulis berharap semoga laporan ini dapat memberikan manfaat kepada pembaca dan bisa diambil manfaatnya untuk pengembangan selanjutnya.

Malang, 01 Maret 2011

Penulis



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
DAFTAR <i>SOURCECODE</i>	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
1.6 Sistematika Penulis	3
BAB II TINJAUAN PUSTAKA	5
2.1 Wajah	5
2.2 Citra	5
2.3 Warna Tingkat Keabuan (<i>Grayscale</i>)	7
2.4 Metode Ekstraksi Fitur Wajah	8
2.4.1 Ekstraksi Fitur	8
2.4.2 Fitur Wajah	9
2.5 Pengenalan Menggunakan LDA (<i>Linear Discriminant Analysis</i>)	10
2.5.1 PCA (<i>Principal Component Analysis</i>)	10
2.5.2 LDA (<i>Linear Discriminant Analysis</i>)	12
2.5.3 Jarak <i>Euclidean</i> (Jarak Terdekat)	13

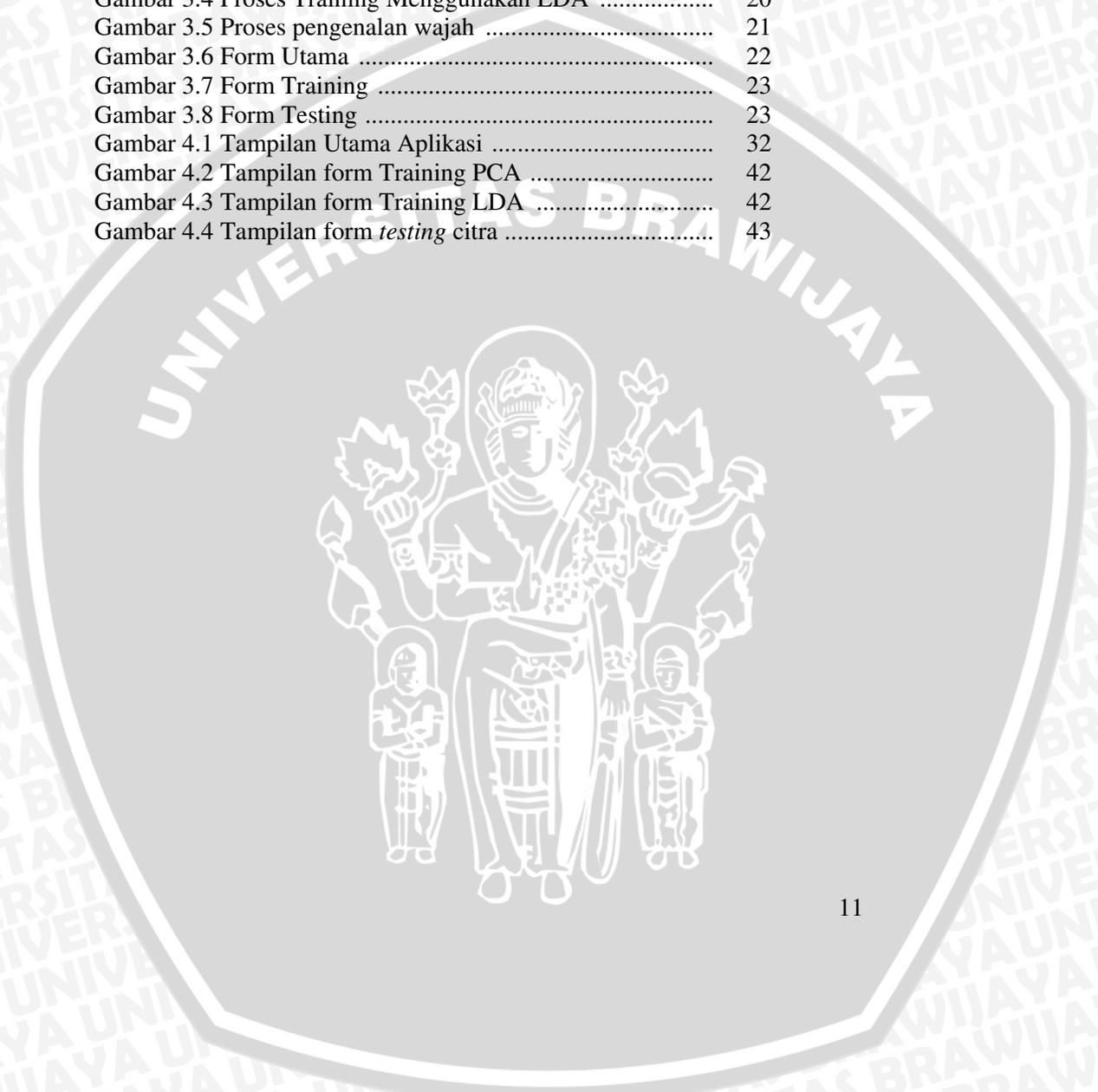
BAB III METODOLOGI DAN PERANCANGAN	15
3.1 Analisis Sistem	17
3.1.1 Deskripsi Sistem	17
3.1.2 Deskripsi Data	17
3.2 Perancangan Proses	17
3.2.1 Proses Pelatihan Pengenalan Wajah dengan LDA	17
3.2.2 Proses PCA	19
3.2.3 Proses Training LDA	20
3.2.4 Proses Pengenalan Wajah Menggunakan Pencarian Jarak Terdekat	21
3.3 Perancangan Antarmuka	22
3.3.1 Form Utama	22
3.3.2 Form Training	23
3.3.3 Form Testing	23
3.4 Perhitungan Manual	24
3.4.1 Perhitungan Data Training	24
3.4.1.1 Perhitungan reduksi dimensi dengan PCA	24
3.4.1.2 Perhitungan LDA.....	26
3.4.2 Perhitungan Data Uji	27
3.5 Metode Pengujian	29
 BAB IV HASIL DAN PEMBAHASAN	 31
4.1 Lingkungan Implementasi	31
4.1.1 Lingkungan Perangkat Keras	31
4.1.2 Lingkungan Perangkat Lunak	31
4.2 Implementasi Program	32
4.2.1 Proses <i>Training</i>	32
4.2.1.1 Proses PCA	24
4.2.1.2 Proses LDA	36
4.2.2 Proses Pengenalan	39
4.3 Penerapan Aplikasi	41
4.4 Skenario Pengujian	43
4.4.1 Data Pengujian	43
4.4.2 Hasil Pengujian	44
4.5 Analisa Hasil	46
 BAB V KESIMPULAN DAN SARAN	 47
5.1 Kesimpulan	47
5.2 Saran	47

DAFTAR PUSTAKA 49
LAMPIRAN 51



DAFTAR GAMBAR

	Halaman
Gambar 2.1 Wajah manusia	5
Gambar 2.2 Data matriks dua dimensi	6
Gambar 2.3 citra <i>grayscale</i>	8
Gambar 3.1 Alur penelitian	16
Gambar 3.2 Alur Proses Pelatihan Pengenalan Wajah	18
Gambar 3.3 Proses PCA	19
Gambar 3.4 Proses Training Menggunakan LDA	20
Gambar 3.5 Proses pengenalan wajah	21
Gambar 3.6 Form Utama	22
Gambar 3.7 Form Training	23
Gambar 3.8 Form Testing	23
Gambar 4.1 Tampilan Utama Aplikasi	32
Gambar 4.2 Tampilan form Training PCA	42
Gambar 4.3 Tampilan form Training LDA	42
Gambar 4.4 Tampilan form <i>testing</i> citra	43



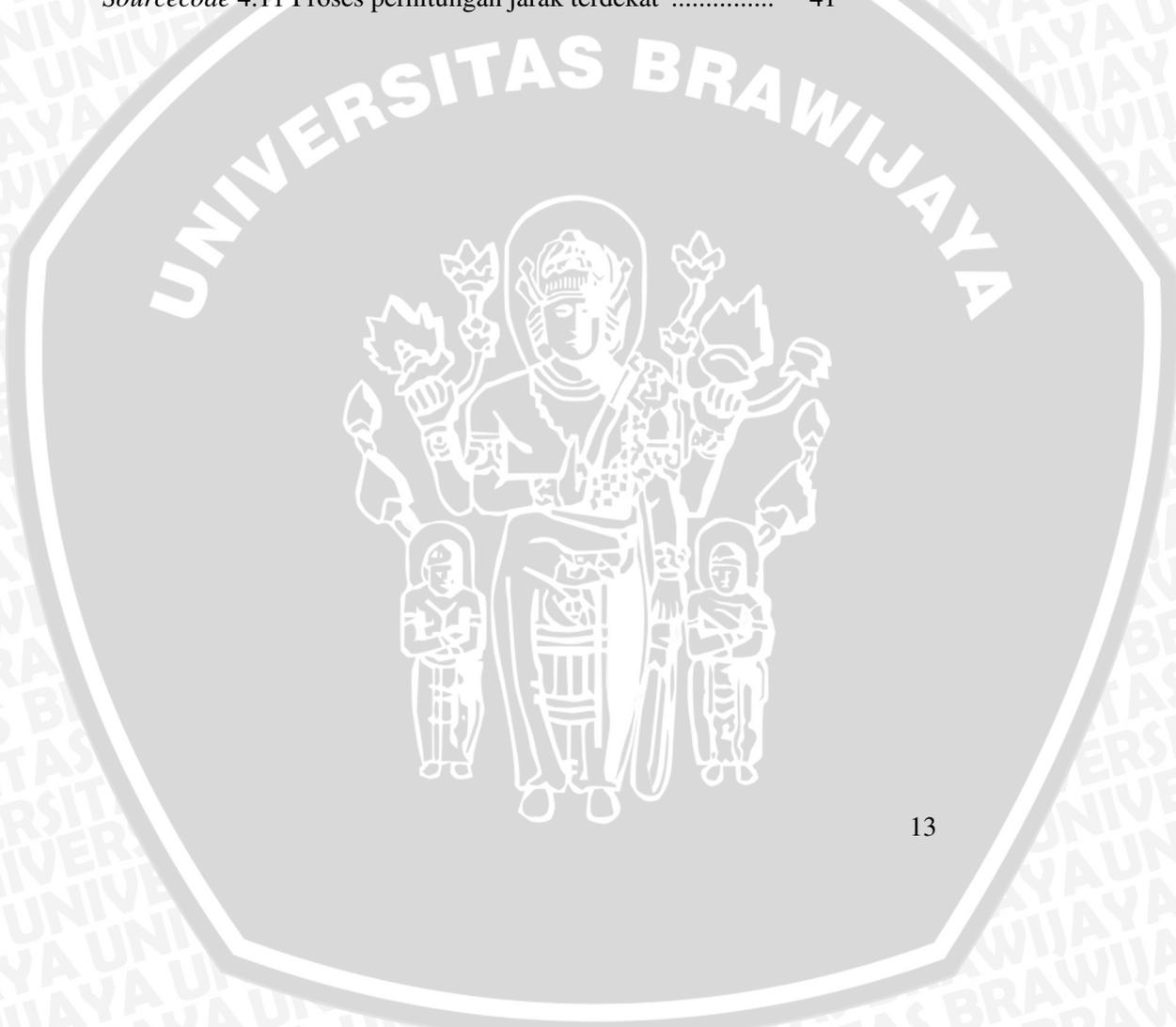
DAFTAR TABEL

	Halaman
Tabel 3.1 Contoh Hasil Percobaan Pengujian Pengenalan Wajah Dengan LDA	29
Tabel 4.1 Hasil pengujian ORL	44
Tabel 4.2 Hasil pengujian YALE	45



DAFTAR SOURCECODE

	Halaman
<i>Sourcecode</i> 4.1 Proses Citra <i>Grayscale</i> ke Matriks	33
<i>Sourcecode</i> 4.2 Proses mendapatkan matriks kovarian	34
<i>Sourcecode</i> 4.3 Proses matriks V PCA	35
<i>Sourcecode</i> 4.4 Proses fitur PCA	35
<i>Sourcecode</i> 4.5 Proses <i>scatter within</i> dan <i>scatter between</i>	37
<i>Sourcecode</i> 4.6 Proses matriks kovarian LDA	37
<i>Sourcecode</i> 4.7 Proses matriks V LDA	38
<i>Sourcecode</i> 4.8 Proses fitur LDA	38
<i>Sourcecode</i> 4.9 Proses Citra <i>Grayscale</i> ke Matriks	39
<i>Sourcecode</i> 4.10 Proses fitur data uji	40
<i>Sourcecode</i> 4.11 Proses perhitungan jarak terdekat	41



DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Citra <i>Training</i> dan Citra Uji	51
Lampiran 2 Hasil Pengujian	60



BAB I PENDAHULUAN

1.1 Latar Belakang

Teknologi *computer vision* dan *image processing* saat ini terus berkembang sangat pesat. Perkembangan ini ditunjang oleh kemajuan teknologi mikroprosesor dan perkembangan mikroelektronik. Berbagai aplikasi *computer vision* dan *image processing* yang telah dikembangkan saat ini misalnya dalam sistem keamanan, dunia robotika, dan lain sebagainya.

Computer vision mempunyai tujuan utama untuk membuat keputusan yang berkaitan dengan obyek fisik nyata dan pemandangan (*scenes*) berdasarkan citra yang didapat dari sensor. *Image processing* atau pengolahan citra merupakan salah satu jenis teknologi untuk menyelesaikan masalah mengenai pemrosesan citra.

Dalam skripsi ini, akan dibuat *software* yang dapat mengenali wajah seseorang. *Software* ini dapat diaplikasikan pada sistem keamanan yang digunakan diberbagai tempat seperti hotel, bandara, ATM dan masih banyak lain.

Beberapa penelitian tentang *face recognition* pernah dibuat. Salah satunya adalah PCA (*Principal Component Analysis*). Hasil penelitian menunjukkan bahwa menggunakan metode PCA dapat mengenali wajah dengan tingkat keakuratan 74 %. (Juwei Lu, K.N. Plataniotis and A.N. Venetsanopoulos, 2002).

Pada skripsi ini akan dibuat *face recognition* menggunakan LDA (*Linear Discriminant Analysis*) dengan PCA (*Principal Component Analysis*) untuk reduksi dimensi. Metode LDA dipilih karena untuk *feature extraction* dan pengenalan wajah dengan menggabungkan distribusi sampel dan mempertimbangkan informasi yang diskriminatif dari *within-class scatter matrix* dan *between-class scatter matrix*. Sebelum masuk proses LDA, akan dilakukan terlebih dahulu dimensi reduksi menggunakan PCA. Reduksi dimensi digunakan untuk memudahkan kompleksitas perhitungan dengan tidak mengurangi hasil akhir pengenalan. Dengan melakukan ini, bisa mengurangi sensitivitas metode untuk varian substansial antara citra wajah disebabkan oleh ukuran besar, ekspresi atau variasi iluminasi. Metode LDA diharapkan dapat menghasilkan suatu sistem yang mampu memberikan *output* berupa pengenalan wajah dengan

tingkat akurasi yang lebih baik. (Wankuo Yang., Hui Yan., Jianguo Wang., Jingyu Yang., 2008).

1.2 Perumusan Masalah

Berdasarkan uraian pada latar belakang masalah, maka dalam skripsi ini dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana perancangan dan implementasi untuk pengenalan wajah dengan metode LDA
2. Bagaimana tingkat akurasi yang dihasilkan dari pengenalan wajah menggunakan metode LDA

1.3 Batasan Masalah

Dari permasalahan diatas, berikut ini diberikan batasan masalah untuk menghindari melebarnya masalah yang akan diselesaikan:

1. Menggunakan database wajah manusia Yale dan ORL face database untuk *training* wajah.
2. Citra wajah adalah *grayscale*.
3. Wajah tidak memakai aksesoris penutup kepala dan kerudung.

1.4 Tujuan Penelitian

Tujuan penelitian yang ingin dicapai adalah:

1. Merancangan dan mengimplementasikan metode LDA untuk pengenalan wajah.
2. Mengetahui tingkat akurasi pengenalan wajah menggunakan metode LDA.

1.5 Manfaat Penelitian

1. Menyediakan aplikasi yang mampu mengenali wajah manusia sehingga dapat digunakan untuk sistem keamanan dan absensi.
2. Mengetahui tingkat akurasi dari penerapan LDA.

1.6 Sistematika Penulisan

Skripsi ini disusun berdasarkan sistematika penulisan sebagai berikut:

1. **BAB I PENDAHULUAN**
Berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.
2. **BAB II TINJAUAN PUSTAKA**
Berisi penjelasan tentang wajah, citra, *grayscale* dan LDA (*Linear Discriminant Analysis*).
3. **BAB III METODOLOGI DAN PERANCANGAN SISTEM**
Pada bab ini akan dijelaskan mengenai metode-metode yang digunakan dalam pembuatan perangkat lunak pengenalan wajah dengan metode LDA.
4. **BAB IV HASIL DAN PEMBAHASAN**
Pada bab ini berisi tentang penjelasan implementasi sistem dan hasil pengujian yang dilakukan.
5. **BAB V KESIMPULAN DAN SARAN**
Berisi kesimpulan dari seluruh rangkaian penelitian serta saran kemungkinan pengembangannya.



BAB II

TINJAUAN PUSTAKA

Pada bab ini akan membahas tentang dasar teori yang menunjang penelitian, yaitu mengenai wajah, citra, *grayscale* dan LDA (*Linear Discriminant Analysis*).

2.1 Wajah

Wajah manusia memiliki bagian-bagian yang unik di setiap individu yang berbeda. Misalkan pada warna mata, seperti hitam, coklat, dan biru. Bentuk muka adalah yang paling signifikan dalam pengenalan selain mulut, hidung, pipi, mata, dahi, bahkan rambut. Bentuk muka manusia tergantung struktur tengkorak manusia.

Wajah manusia adalah bagian depan kepala manusia termasuk dagu, mulut, hidung, pipi, mata, dan dahi. Wajah merupakan salah satu tanda pengenal biometri yang cukup terpercaya, selama pada masih dalam kondisi normal. Wajah digunakan sebagai pengenal fisik yang digunakan untuk mengenali seorang individu (Mohamad Aditya Rahman, 2010). Citra wajah manusia ditunjukkan pada Gambar 2.1.



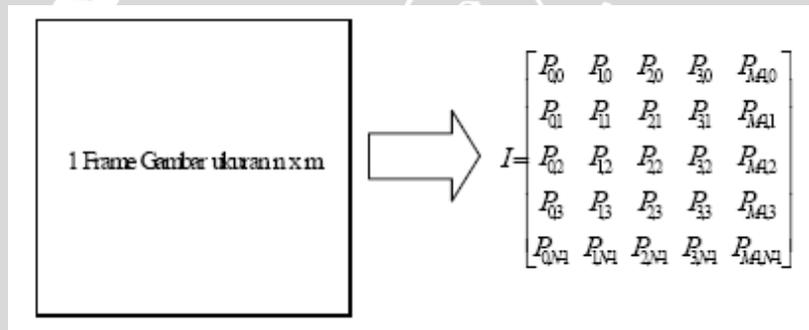
Gambar 2.1 Wajah manusia

2.2 Citra

Citra adalah representasi dari dua dimensi untuk bentuk fisik nyata tiga dimensi. Citra dalam perwujudannya dapat bermacam-macam, mulai dari gambar hitam-putih pada sebuah foto (yang tidak bergerak) sampai pada gambar berwarna yang bergerak pada pesawat televisi. Proses transformasi dari bentuk tiga dimensi ke bentuk dua dimensi untuk menghasilkan citra akan dipengaruhi oleh bermacam-macam faktor yang mengakibatkan penampilan citra suatu benda tidak sama persis dengan bentuk fisik nyatanya. Faktor-faktor tersebut merupakan efek degradasi

atau penurunan kualitas yang dapat berupa rentang kontras benda yang terlalu sempit atau terlalu lebar, distorsi, keaburan (*blur*), keaburan akibat objek citra yang bergerak (*motion blur*), noise atau gangguan yang disebabkan oleh interferensi peralatan pembuat citra, baik itu berupa transducer, peralatan elektronik ataupun peralatan optik karena pengolahan citra digital dilakukan dengan komputer digital maka citra yang akan diolah terlebih dahulu ditransformasikan ke dalam bentuk besaran-besaran diskrit dari nilai tingkat keabuan pada titik-titik elemen citra. Bentuk dari citra ini disebut citra digital. Elemen-elemen citra digital apabila ditampilkan dalam layar monitor akan menempati sebuah ruang yang disebut dengan piksel (*picture element / piksel*).

Citra dinyatakan dalam bentuk data matriks 2 dimensi, dimana setiap titik data mewakili satu piksel. Dalam hubungannya dengan data video, maka satu gambar (*image*) dikenal sebagai satu *frame*. Misalnya sebuah gambar dikatakan resolusinya sebesar 800 x 600 maka berarti panjang piksel horizontalnya 800 dan panjang piksel vertikalnya 600 dan jumlah total keseluruhan piksel dari gambar tersebut yaitu 480000 atau dapat dikatakan bahwa gambar tersebut terdiri dari 480000 piksel. Dimensi matriks yang mewakili 1 *frame* citra dengan ukuran M x N ditunjukkan pada Gambar 2.2.



Gambar 2.2 Data matriks dua dimensi

Dalam pengolahan citra, piksel mempunyai hubungan antara piksel satu dengan yang lainnya. Sebuah piksel p pada koordinat (x,y) mempunyai empat tetangga horisontal dan vertikal yang memiliki koordinat-koordinat sebagai berikut, (x+1,y+1), (x+1,y-1), (x,y+1), dan (x,y-1) Kumpulan dari piksel diatas yang disebut 4-neighbours of p dapat dinyatakan sebagai N4(p), kecuali jika p(x,y) posisinya terletak digaris batas gambar, sehingga

jumlah piksel tetangga tidak terdiri dari 4 tetangga. Selain 4 tetangga diatas, p juga mempunyai 4 tetangga diagonal ditunjukkan pada persamaan 2.1

$$(x+1,y+1), (x+1,y-1), (x-1,y+1), \text{ dan } (x-1,y-1) \quad (2.1)$$

Piksel-piksel tersebut dinyatakan sebagai $N_p(p)$. Gabungan dari $N_4(p)$ dan $p(p)$ didefinisikan sebagai 8-neighbours of p dan dinyatakan sebagai $N_8(p)$. Hubungan antar piksel merupakan suatu konsep yang sangat penting, yang digunakan untuk mendefinisikan batas-batas dari suatu obyek serta bagian-bagian daerah kecil dari suatu gambar. Hubungan antar piksel ada beberapa kriteria, diantaranya adalah prinsip kedekatan kedua piksel yang sesuai dengan konsep yang telah ditentukan, seperti konsep 4-neighbours atau 8-neighbours. Selain itu, apakah kedua piksel tersebut memiliki nilai grayscale yang sesuai dengan kriteria yang diinginkan. Sebagai contoh, bila dua piksel mempunyai nilai masing-masing 0 dan 1 dan keduanya merupakan bagian dari 4-neighbours, maka dinyatakan bahwa kedua piksel tersebut tidak ada hubungan, hal ini karena keduanya memiliki nilai yang berbeda.

2.3 Warna Tingkat Keabuan (*Grayscale*)

Kecerahan dari citra yang disimpan dengan cara pemberian nomor pada tiap-tiap pikselnya. Semakin tinggi nomor pikselnya maka makin terang (putih) piksel tersebut. Sedangkan semakin kecil nilai suatu piksel, mengakibatkan warna pada piksel tersebut menjadi gelap. Dalam sistem kecerahan yang umum terdapat 256 tingkat untuk setiap piksel Skala kecerahan seperti ini dikenal sebagai *grayscale*.

Dalam sistem komputasi maupun fotografi, suatu citra *grayscale* atau tingkat keabuan citra digital adalah citra di mana nilai setiap piksel memiliki satu nilai sampel, yaitu hanya membawa informasi intensitas citra. Foto semacam ini, juga dikenal sebagai foto hitam-putih, yang tersusun secara eksklusif dari nuansa abu-abu yang bervariasi dari nilai hitam pada intensitas terlemah menuju nilai putih yang terkuat.

Citra *grayscale* memiliki perbedaan dari satu-bit citra hitam-putih, dalam konteks pencitraan komputer citra yang memiliki hanya dua warna yaitu, hitam dan putih (juga disebut *bilevel* atau *gambar biner*). Citra *grayscale* juga disebut citra monokromatik yang menunjukkan tidak adanya variasi warna. (Stephen Johnson, 2006). Citra *grayscale* ditunjukkan pada Gambar 2.3.



Gambar 2.3 citra grayscale
Sumber: Stephen Johnson, 2006

2.4 Metode Ekstraksi Fitur Wajah

2.4.1 Ekstraksi Fitur

Ekstraksi Fitur adalah proses untuk mendapatkan ciri-ciri/fitur pembeda yang dapat digunakan untuk membedakan antara wajah antar individu yang berbeda, bagi sebagian besar aplikasi *pattern recognition*, teknik ekstraksi fitur yang handal merupakan kunci utama dalam penyelesaian masalah *pattern recognition* (M Turk dan Pentland, 1991).

Banyak metode yang digunakan untuk ekstraksi fitur. Beberapa metode yang umum digunakan yaitu, PCA (*Principal Component Analysis*), DCT (*Discrete Cosinus Transform*), *Gabor Filter*, metode *moment*, dan *wavelet transform*. Dalam skripsi ini hanya menggunakan metode LDA (*Linear Discriminant Analysis*).

2.4.2 Fitur Wajah

Dalam dunia matematika, *eigenvalue* (nilai eigen), *eigenvector* (vektor eigen), dan *eigenspace* (ruang eigen) memiliki keterkaitan dalam suatu ruang lingkup bidang aljabar linier. Kesemuanya itu merupakan hal yang penting dalam fitur wajah, khususnya dengan metode PCA. Kata *eigen* berasal dari Jerman yang berarti bawaan, berbeda, dan diri. Dalam aljabar linier, dipelajari mengenai transformasi linier yang diwakili oleh matriks yang bekerja pada vektor. *Eigenvalue*, *eigenvector* dan *eigenspaces* adalah properti dari sebuah matriks. Ketiganya dapat memberikan informasi penting tentang sebuah matriks dan dapat digunakan dalam faktorisasi matriks. Selain itu, ketiganya juga dapat digunakan dalam

bidang matematika terapan yang beragam seperti keuangan dan mekanika kuantum.

Pada umumnya, sebuah matriks yang berperan sebagai vektor dapat diubah magnitude dan arah. Sebuah matriks dapat berperan pada sebuah vektor yang tepat dengan cara merubah nilai magnitude-nya dan tanpa merubah arahnya. Vektor inilah yang disebut *eigenvector*. Matriks juga dapat dihasilkan dari hasil perkalian nilai *eigenvector* dengan faktor pengali yang disebut *eigenvalue*. Jika nilai *eigenvalue* bernilai positif maka arahnya tidak berubah, sedangkan yang bernilai negatif arahnya terbalik. Sedangkan *eigenspace* merupakan himpunan *eigenvector* yang memiliki *eigenvalue* yang sama.

Secara formal jika matriks A adalah transformasi linier, vektor x adalah *eigenvector* dari matriks A dan λ adalah matriks skalar seperti yang ditunjukkan dengan formula pada 2.2.

$$Ax = \lambda x \quad (2.2)$$

Nilai matriks skalar λ dapat dikatakan sebagai *eigenvalue* dari matriks A yang bersesuaian dengan *eigenvector* x.

Proses menghitung *eigenvalue* dan *eigenvector* dari matriks bujur sangkar dengan persamaan 2.2 dibentuk menjadi persamaan 2.3 berikut :
Persamaan 2.3 yaitu tahap-tahap penurunan rumus pencarian *eigenvalue* dan *eigenvector* dari sebuah matrik bujur sangkar.

$$\begin{aligned} \text{a. } & Ax - \lambda Ix = 0 \\ \text{b. } & (A - \lambda I)x = 0 \\ \text{c. } & (A - \lambda I)^{-1} = x \\ \text{d. } & \det(A - \lambda I) = 0 \end{aligned} \quad (2.3)$$

Keterangan :

1. *Eigenvalue* (λ) dari matriks bujur sangkar (A).
2. I adalah matriks identitas.
3. Proses invers digunakan untuk mencari *eigenvector*.
4. Jika kedua sisi dalam persamaan dikalikan dengan invers untuk mendapatkan solusi $x = 0$. Jadi untuk menghilangkan invers dengan mengasumsikan dari Aljabar linier bahwa determinan sama dengan nol.

- Setelah didapatkan nilai eigen, maka substitusikan kembali ke persamaan untuk mendapatkan vektor eigen.

2.5 Pengenalan Menggunakan LDA (*Linear Discriminant Analysis*)

Pengenalan wajah menggunakan metode LDA yaitu melakukan input data latih, kemudian melakukan reduksi dimensi dengan metode PCA (*Principal Component Analysis*), setelah dilakukan reduksi dimensi kemudian melakukan perhitungan LDA (*Linear Discriminant Analysis*).

2.5.1 PCA (*Principal Component Analysis*)

PCA (*Principal Component Analysis*) melibatkan suatu prosedur matematika yang mengubah beberapa variabel yang berkorelasi ke dalam sejumlah kecil variabel disebut komponen utama. Komponen utaman ini sebagai pengganti variabel lainnya.

PCA ditemukan pada tahun 1901 oleh Karl Pearson. Sekarang banyak digunakan sebagai analisis data eksplorasi dan untuk membuat model prediksi. PCA melibatkan perhitungan dekomposisi *eigenvalue* dari suatu data matrik kovarian atau dekomposisi nilai singular dari matrik data.

PCA digunakan untuk mereduksi *dimensional space* dari fitur-fitur wajah. Dengan cara ini diharapkan juga akan mereduksi kompleksitas waktu yang digunakan untuk pemrosesan. Harapan dari *dimensional space* yang rendah merupakan ekstraksi dari struktur prinsip/penting dari sekumpulan data set.Reduksi dimensi dilakukan dengan memilih beberapa nilai dari nilai *eigenvector* yang berhubungan dengan *eigenvalue* terbesar. (M Turk dan Pentland, 1991)

Langkah-langkah untuk mereduksi dimensi fitur citra :

- Mengubah matriks citra menjadi vektor citra.

$$A_i = \begin{matrix} N_{1,1} & N_{2,1} & \cdots & N_{1,P} \\ N_{2,1} & N_{2,2} & \cdots & N_{2,P} \\ \vdots & \vdots & \ddots & \vdots \\ N_{i,1} & N_{i,2} & \cdots & N_{i,P} \end{matrix} \rightarrow \begin{bmatrix} N_{1,1} \\ \vdots \\ N_{1,P} \\ \vdots \\ N_{i,P} \end{bmatrix}$$

Keterangan :

A = Matrik

$N_{i,p}$ = piksel ke $-i$ dan ke $-p$

2. Mencari nilai rata-rata citra data *training* \bar{z} ditunjukkan pada persamaan 2.4

$$\bar{z}_{PCA} = \frac{1}{Y} \sum_{i=1}^Y z_i \quad (2.4)$$

Keterangan :

\bar{z}_{PCA} = rata-rata citra data *training*
 Y = jumlah data *training*
 z_i = vector citra *training*

3. Kovarian matriks dirumuskan pada persamaan 2.5.

$$x_i = A^T (A) \quad (2.5)$$

Keterangan :

x_i = kovarian matriks
 A = Kumpulan matriks data *training*
 A^T = Hasil transpose dari matriks A

4. Mencari nilai *eigenvalue* dan *eigenvector* dari kovarian matriks dengan persamaan 2.2.

5. Kemudian mencari nilai matriks V_{PCA} dengan persamaan 2.6.

$$V_{PCA} = A * \text{eigenvector} \quad (2.6)$$

6. Setelah itu ambil beberapa fitur penting dari *eigenvector* berdasarkan *eigenvalue*, sehingga didapatkan jumlah dimensi fitur yang lebih kecil dari fitur aslinya dengan cara memproyeksikan nilai citra asli ke dalam ruang fitur dengan persamaan 2.7.

$$f_{PCA} = \sum_{i=1}^N \left[z_i - \bar{z}_{PCA} \right]^T \times V_{PCA} \quad (2.7)$$

Keterangan :

f_{PCA} = proyeksi citra ke ruang fitur
 V_{PCA} = nilai perkalian *eigenvector* dengan komponen

2.5.2 LDA (*Linear Discriminant Analysis*)

LDA merupakan suatu teknik untuk menemukan kombinasi linier fitur-fitur yang terbaik dalam memisahkan 2 atau lebih kelas obyek. Langkah utama LDA adalah bagaimana menggabungkan kontribusi masing-masing sampel pelatihan ke dalam matriks scatter. *Within scatter matrix* dan *between scatter matrix* ditunjukkan pada persamaan 2.8 dan 2.9:

$$S_w = \sum_{i=1}^c \sum_{x_j \in W_i} (x_j - m_i)(x_j - m_i)^T \quad (2.8)$$

$$S_b = \sum_{i=1}^c (m_i - m)(m_i - m)^T \quad (2.9)$$

Keterangan :

m = rata-rata dari semua contoh.

c = jumlah kelas.

X_j = data training.

S_w = *within scatter matrix*

S_b = *between scatter matrix*

Kemudian mencari matriks kovarian LDA menggunakan persamaan 2.10

$$C = S_b S_w^{-1} \quad (2.10)$$

Keterangan :

C = Matriks kovarian.

Setelah itu mencari nilai *eigenvector* dan *eigenvalue*. Lalu mencari matriks Vektor LDA menggunakan persamaan 2.11

$$V_{LDA} = F_{PCA} * \text{eigenvector} \quad (2.11)$$

Kemudian mengitung nilai fitur LDA menggunakan persamaan 2.12.

$$f_{LDA} = \sum_{i=1}^N \left(z_i - \bar{z}_{LDA} \right)^T \times V_{LDA} \quad (2.12)$$

Keterangan :

f_{LDA} = proyeksi citra ke ruang fitur LDA

\bar{z}_{LDA} = rata-rata dari fitur PCA.

V_{LDA} = Matriks V LDA.

(Juwei Lu, K.N. Plataniotis and A.N. Venetsanopoulos, 2002)

2.5.3 Jarak Euclidean (Jarak Terdekat)

Jarak *euclidean* digunakan untuk pengenalan, menghitung jarak dari fitur *training* dengan fitur uji menggunakan persamaan 2.13.

$$d = \sqrt{\sum_{i=1}^n (b_i - a_i)^2} \quad (2.13)$$

Keterangan :

d = jarak *euclidean*

n = banyaknya data

a_i = data *training* ke- i

b_i = data uji ke- i

(K.Etemad and R.Chellapa, 1997)

UNIVERSITAS BRAWIJAYA

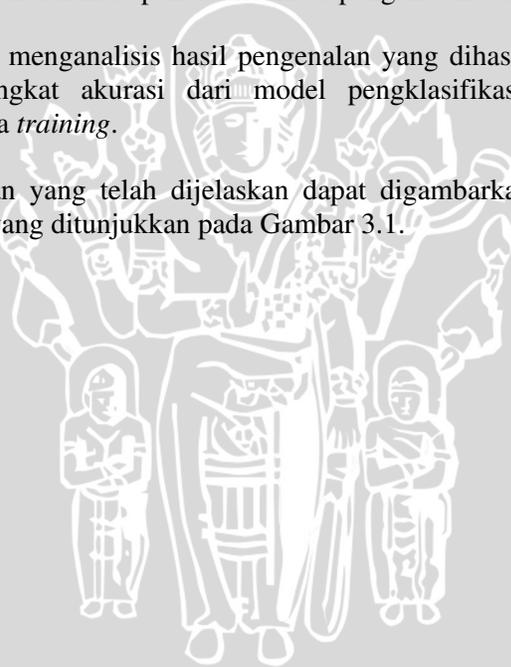


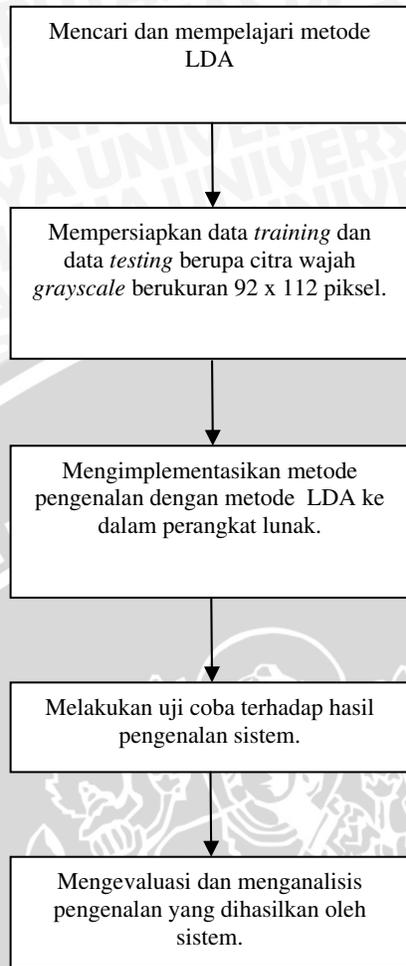
BAB III METODOLOGI DAN PERANCANGAN

Pada bab ini dibahas tentang metodologi dan perancangan yang digunakan dan seluruh tahapan yang digunakan dalam penelitian ini. Tahapan-tahapan yang harus dilakukan dalam penelitian ini dapat diidentifikasi sebagai berikut :

1. Mencari dan mempelajari literatur-literatur yang berkaitan dengan permasalahan pengenalan wajah melalui sebuah citra *grayscale* menggunakan metode LDA yang berasal dari buku dan sumber lain dari internet.
2. Mempersiapkan data citra wajah yaitu berupa data *training* dan data *testing*. Citra wajah yang digunakan berbentuk citra *grayscale* dan berukuran 92 x 112 piksel.
3. Mengimplementasikan metode pengenalan dengan metode LDA menggunakan data latih yang telah disiapkan sebagai proses pembelajaran yang akan digunakan dalam pengenalan ke sebuah sistem perangkat lunak.
4. Melakukan uji coba terhadap hasil pengenalan yang dilakukan oleh sistem. Hasil pengenalan merupakan informasi pengenalan dari individu yang diuji.
5. Mengevaluasi dan menganalisis hasil pengenalan yang dihasilkan oleh sistem berupa tingkat akurasi dari model pengklasifikasian yang dihasilkan dari data *training*.

Tahapan-tahapan yang telah dijelaskan dapat digambarkan dengan sebuah diagram alur yang ditunjukkan pada Gambar 3.1.





Gambar 3.1. Alur penelitian.

3.1 Analisis sistem

3.1.1 Deskripsi Sistem

Pada sistem ini akan diimplementasikan mengenai proses yang digunakan untuk mengenali suatu wajah individu dengan metode LDA. Sistem yang dibangun ini bertujuan agar dapat digunakan untuk mengenali wajah seseorang dari masukan berupa citra wajah.

3.1.2 Deskripsi Data

Sistem yang dibangun ini menggunakan data latih dan data uji yang berasal dari YALE database dan ORL database. YALE database diperoleh dari <http://cvc.yale.edu/projects/yaleface/yalefaces.html>. Database ini terdiri dari 165 citra dari 15 subjek. Terdapat 11 citra setiap subjeknya, terdiri dari berbagai ekspresi wajah dan terdapat tambahan aksesoris. ORL database diperoleh dari <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>. Database ini berisi 400 citra dengan subjek yang berbeda dan tiap subjek ada 10 citra wajah. Tiap citra berisi wajah berbagai ekspresi dan citra wajah dengan penambahan aksesoris.

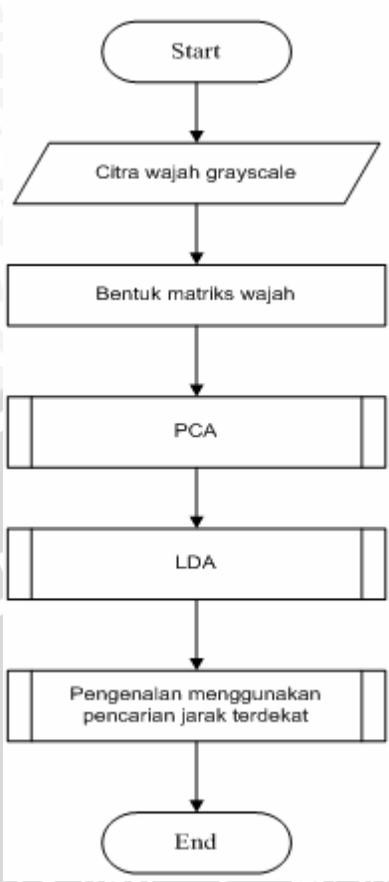
YALE database dan ORL database berisi citra wajah *grayscale*. Sistem akan menggunakan YALE database dan ORL database yang telah disesuaikan ukuran dan format data citranya menjadi 92 x 112 pixel dengan format citra JPG. Dalam database berisi citra wajah dengan beragam ekspresi dan jenis kelamin yang berbeda dan acak.

Data set yang berasal dari YALE database dan ORL database yang akan digunakan adalah 25 citra untuk citra *training* dan 25 citra untuk citra uji .

3.2 Perancangan Proses

3.2.1 Proses Pelatihan Pengenalan Wajah dengan LDA

Pada perancangan sistem pengenalan wajah ini diperlukan beberapa tahapan yang perlu dilalui agar dapat membentuk sistem pengenalan wajah yang diharapkan dengan metode LDA. Dalam proses pengenalan wajah ini memerlukan inputan berupa data citra dari YALE database dan ORL database. Proses pengenalan wajah dimulai dengan memasukkan citra uji, kemudian akan dicari kemungkinan tertinggi yang menentukan objek terdekat dari data latih. Pada Gambar 3.2 dapat dilihat proses pelatihan sistem pengenalan wajah dengan LDA.



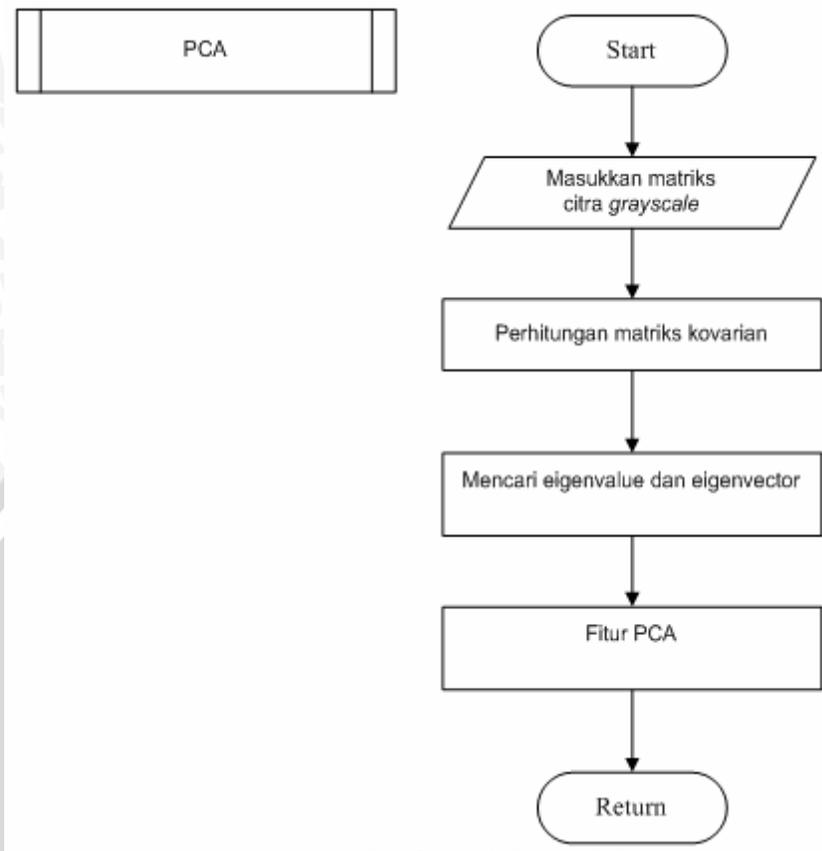
Gambar 3.2 Alur Proses Pelatihan Pengenalan Wajah

Berikut ini penjelasan dari tahapan proses pelatihan :

1. Pertama adalah memasukkan citra wajah *grayscale*.
2. Proses selanjutnya membentuk matriks wajah dari citra *grayscale*.
3. fitur PCA (reduksi dimensi) matriks citra wajah dengan PCA.
4. Setelah itu menghitung LDA, untuk membentuk matriks *scatter within* dan *scatter between* untuk mencari fitur LDA.
5. Dilakukan perhitungan jarak terdekat untuk pengenalan wajah.

3.2.2 Proses PCA

Pada Gambar 3.3 dapat dilihat proses PCA.



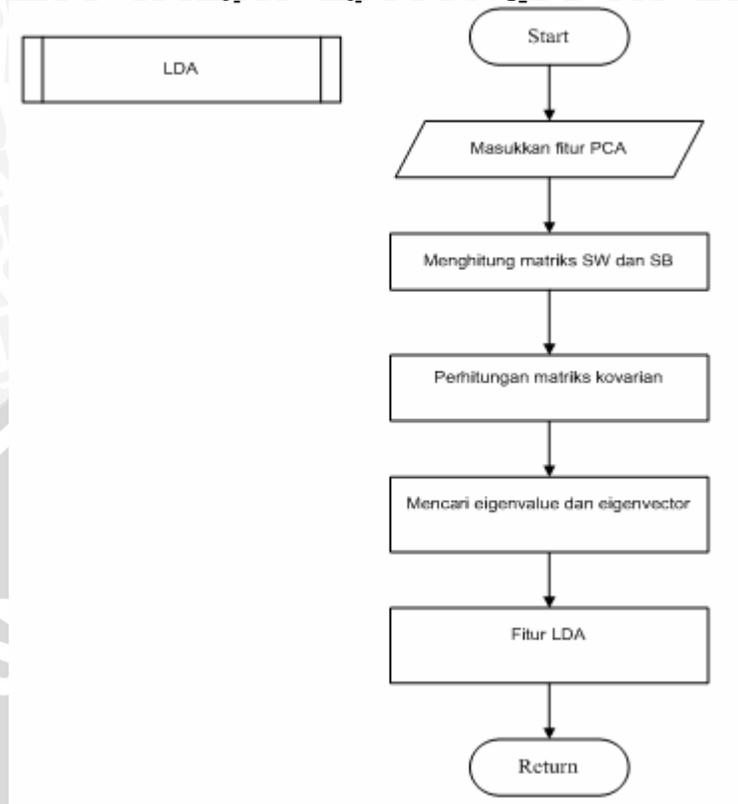
Gambar 3.3 Proses PCA

Berikut ini penjelasan dari tahapan proses PCA :

1. Masukkan matriks citra *grayscale*.
2. Setelah itu menghitung matriks kovarian dari hasil perkalian matriks citra dengan matriks citra yang sudah di *transpose* menggunakan persamaan 2.5.
3. Mencari *eigen value* dan *eigen vector* dengan menggunakan nilai dari matriks kovarian.
4. Menghitung fitur PCA menggunakan persamaan 2.7.

3.2.3 Proses Training LDA

Pada Gambar 3.4 dapat dilihat proses training LDA.



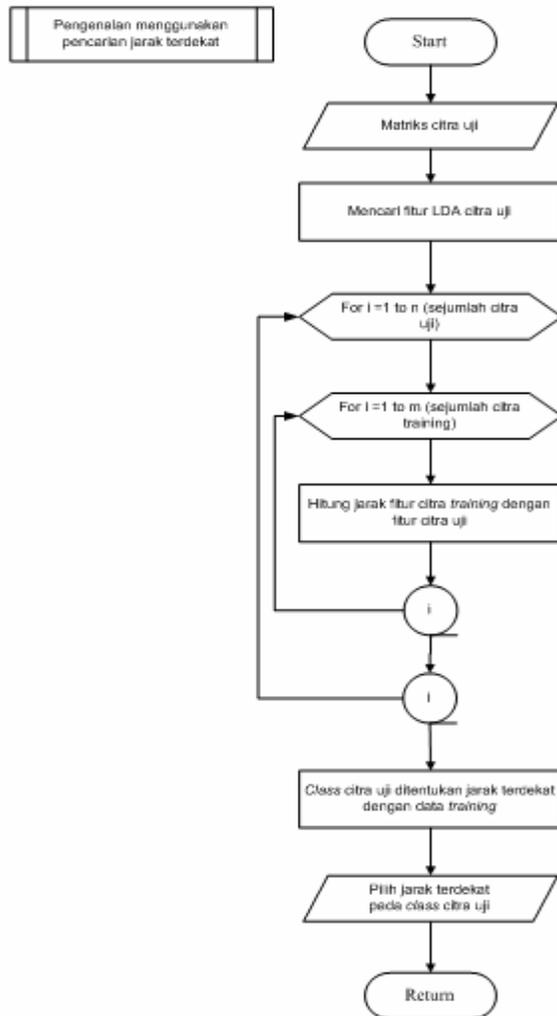
Gambar 3.4 Proses Training Menggunakan LDA

Berikut ini penjelasan dari tahapan proses training LDA :

1. Menghitung matriks S_w dan matriks S_b dari fitur PCA menggunakan persamaan 2.8 dan 2.9.
2. Setelah itu matriks S_w dan matriks S_b digunakan untuk mendapatkan matriks kovarian menggunakan persamaan 2.10.
3. Mencari *eigen value* dan *eigen vector* menggunakan nilai dari matriks kovarian.
4. Menghitung fitur LDA menggunakan persamaan 2.12.

3.2.4 Proses Pengenalan Wajah Menggunakan Pencarian Jarak Terdekat

Pada Gambar 3.5 dapat dilihat proses pengenalan menggunakan pencarian jarak terdekat.



Gambar 3.5 Proses pengenalan wajah

Berikut ini penjelasan tahapan proses pengenalan menggunakan pencarian jarak terdekat :

1. Pertama adalah masukkan matriks citra uji.
2. Mencari fitur LDA citra uji.
3. Hitung jarak fitur citra *training* terhadap fitur citra data uji menggunakan rumus jarak *euclidean* menggunakan persamaan 2.13
4. *Class* citra uji ditentukan jarak terdekat dengan data *training*, proses ini dilakukan sejumlah citra uji.
5. Pilih jarak terdekat pada *class* citra uji.

3.3 Perancangan Antarmuka

Pada subbab ini akan dijelaskan mengenai perancangan antar muka yang digunakan dalam aplikasi pengenalan wajah dengan metode LDA (*Linear Discriminant Analysis*). Perancangan program menggunakan bahasa pemrograman C#. Program terdiri dari form utama yang terdiri dari tombol menampilkan form training dan form pengenalan wajah.

3.3.1 Form Utama.

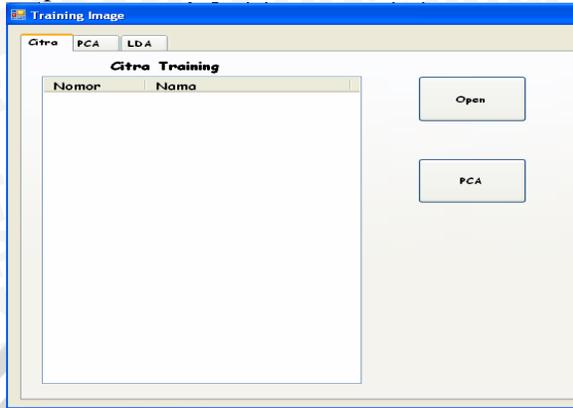
Pada form utama terdapat tombol untuk menampilkan form training dan form pengenalan wajah. Untuk Form Utama dapat dilihat pada Gambar 3.6.



Gambar 3.6 Form Utama

3.3.2 Form Training

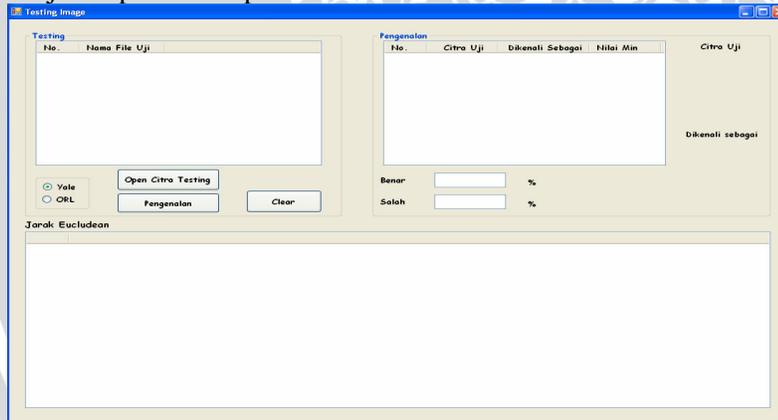
Pada form ini akan digunakan untuk melakukan training data training menggunakan metode LDA. Kemudian hasil training objek wajah disimpan dalam file teks. Input citra training merupakan citra berukuran 92 x 112 pixel berbentuk *grayscale* dan berformat jpg. Untuk Form Training dapat dilihat pada Gambar 3.7.



Gambar 3.7 Form Training

3.3.3 Form Testing

Form digunakan untuk menguji data testing. Pengenalan ini berdasarkan pada data *training*. Hasil dari menu dalam form ini akan menunjukkan informasi dari data yang diuji. Untuk Form Pengenalan Wajah dapat dilihat pada Gambar 3.8.



Gambar 3.8 Form Testing

3.4 Perhitungan Manual

Pada perhitungan manual menunjukkan proses keseluruhan secara umum. Data untuk perhitungan manual terdiri dari dua objek wajah yang masing-masing terdiri dari dua buah data *training*. Data berbentuk *grayscale*. Tahapan perhitungan dimulai dari reduksi dimensi sampai hasil pengenalan.

3.4.1 Perhitungan Data Training

Data *training* memakai empat objek wajah dengan masing-masing dianggap memiliki 2 ekspresi yang berbeda.

Contoh citra data *training* menggunakan ORL database, matriks dari citra *grayscale* :

ORL

Data latih

A1	132	100	163	158
A2	171	169	159	171
B1	113	68	138	121
B2	127	90	144	131

Data uji

Label A

A3	142	85	155	163
A4	113	58	167	138

Label B

B3	129	53	144	115
B4	129	75	141	127

3.4.1.1 Perhitungan reduksi dimensi dengan PCA

Pada PCA nilai piksel dari citra diubah kedalam ruang fitur. Langkah pertama adalah mencari rata-rata piksel dari seluruh citra data *training*,

135.75	106.75	151	145.25
--------	--------	-----	--------

Kemudian menghitung kovarian matriks dengan persamaan 2.5 :

Matriks Kovarian

Xi	78957	92407	63328	69934
	92407	112324	73448	82224
	63328	73448	51078	56194
	69934	82224	56194	62126

nilai *eigenvalue* dan *eigenvector* dari kovarian matriks.

Eigenvector

0.5106	0.3285	-0.7894	-0.0912
0.6056	-0.7756	0.0492	0.1713
0.4087	0.4903	0.3918	0.6626
0.4534	0.2241	0.4701	-0.7234

Eigenvalue

301359	3001.314	124.277	0.3282445
--------	----------	---------	-----------

Kemudian mencari nilai dari matriks V PCA pada persamaan 2.6

274.712	-5.40531	8.18125	0.25702
221.996	-44.787	-1.680721	-0.21999
301.19758	30.150702	0.906958	-0.35754
293.0698	7.95371	-7.32777	0.293169

Setelah itu memproyeksikan nilai citra kedalam ruang fitur PCA berdasarkan persamaan 2.7

	Fitur 1	Fitur 2	Fitur 3	Fitur 4
A1	4822.3663	785.34007	-101.88042	-0.03152
A2	33459.004	-2528.269	2.3297547	0.054636
B1	-25874.571	1270.9874	44.91249	0.215917
B2	-12406.789	471.94095	54.638174	-0.23903

3.4.1.2 Perhitungan LDA

a. *scater within* menggunakan persamaan 2.8

250359487	-26412997	778803.09	-914.955554
-26412797	2904619.3	-88270.75	19.50669572
778803.889	-88270.751	2738.5874	1.138493083
-914.95556	19.506696	1.1384931	0.053600096

b. *scater between* menggunakan persamaan 2.9

2930926653	-133443378	-7621872	1769.62541
-133443378	6075599	347019.37	-80.57001
-7621871.8	347019.37	19820.67	-4.601909
1769.62542	-80.570011	-4.601911	0.00106845

Kemudian mencari matriks kovarian LDA menggunakan persamaan 2.10

45345230.5	-2690344.4	32745.828	-1531.78166
-2690344.4	159618.83	-1942.819	90.881006
32745.828	-1942.8184	23.64723	-1.1061683
-1531.7816	90.881006	-1.106168	0.051744

Setelah itu mencari nilai *eigenvector* dan *eigenvalue*

Eigenvector

0.0223978	0.0112411	-0.00000000006875	-0.999685
0.878379	0.477307	0.00000000463	0.0250471
0.4774388	-0.878664	0.00023136	0.00081668
-0.000115	0.0002033	0.999999973	-0.000019011

Eigenvalue

819960153	7113985472	-258357007	-9.7570049
-----------	------------	------------	------------

Mencari rata-rata dari fitur PCA yang dianggap sebagai data *training* di LDA

0.000000000000909494	0.0000000000008526	0	0
----------------------	--------------------	---	---

Lalu mencari matriks V LDA menggunakan persamaan 2.11

749.1954	518.5759	-0.05505	-4801.26
-1470.2576	-832.691	0.05505	-33511.8
558.3179	276.32955	0.22036	25898.32
162.7442	37.7859	-0.22636	122414.8

Setelah itu menghitung fitur LDA, berdasarkan persamaan 2.12

	Fitur 1	Fitur 2	Fitur 3	Fitur 4
A1	2401355.74	1818663.6	-245.3245	-5110554.13
A2	28785849.1	19456950	-1980.8611	-7585763486
B1	-21228680	-14463853	1504.6896	82803398.82
B2	-9958525.1	-6811759.9	721.49606	45164790.18

3.4.2 Perhitungan Data Uji

Untuk pengenalan dapat dilakukan dengan menghitung jarak dari fitur *training* dengan fitur uji menggunakan persamaan jarak *euclidean*.

Sebelum menghitung jarak *euclidean* terlebih dahulu harus menghitung fitur dari data uji.

Data uji

Label A

A3	142	85	155	163
A4	113	58	167	138

Label B

B3	129	53	144	115
B4	129	75	141	127

Mencari hasil kurang rata-rata PCA data uji dengan cara matrik citra dikurangi rata-rata PCA pada data *training*.

6.25	-21.75	4	17.75
-22.75	-48.75	16	-7.25
-6.75	-53.75	-7	-30.25
-6.75	-31.75	-10	-18.25

Setelah itu mencari fitur PCA dari data uji dengan cara hasil kurang rata-rata PCA data uji dikalikan matriks V PCA dari data *training*.

3295.3124	1200.6317	-38.751	10.1648
-14377.617	2727.758	-36.5506	-2.96903
-24760.355	1988.4662	250.4317	3.7239
-17263.1909	10009.6452	122.80171	3.4748

Kemudian mencari fitur LDA, pertama menghitung hasil kurang rata-rata LDA data uji dengan cara fitur PCA dari data uji dikurangi rata-rata LDA dari data *training*.

3295.3124	1200.6317	-38.751	10.1648
-14377.617	2727.758	-36.5506	-2.96903
-24760.355	1988.4662	250.4317	3.7239
-17263.1909	10009.6452	122.80171	3.4748

Fitur LDA dari data uji dengan cara hasil kurang rata-rata LDA data uji dikalikan matriks V LDA dari data *training*.

683613.4905	698789.9537	-126.401	-56934429.34
-14803043.63	-9737480.723	934.1807	-23364883.34
-21333476.35	-14426562.18	1528.574	58775855.19
-14348814.61	-9758934.231	1033.071	52273592.75

Setelah mendapatkan fitur dari data uji, kemudian menghitung jarak dari fitur *training* dengan fitur uji menggunakan persamaan jarak *euclidean* dengan persamaan 2.13

No	Nilai Minimal	Index	Dikenali sebagai	Data uji
1	5241614.98	0	A	A3
2	35437966.07	0	A	A4
3	19303798.56	3	B	B3
4	8859771.61	3	B	B4

Dari perhitungan menggunakan jarak *euclidean* dihasilkan nilai minimal yang menunjukkan kedekatan jarak antara fitur *training* dengan

fitur uji. Kemudian dicari lokasi nilai minimal berdasarkan urutan index data *training*.

3.5 Metode Pengujian

Pengujian hasil penelitian pengenalan wajah dengan metode LDA. Percobaan akan dilakukan dengan data *set* yang beda, dan data uji dari YALE database dan ORL database. Hasil percobaan dapat dicontohkan seperti Tabel 3.1.

Tabel 3.1 Contoh Hasil Percobaan Pengujian Pengenalan Wajah Dengan LDA

Percobaan	Nama Database Uji	Akurasi (%)
1	Yale	
2	ORL	

UNIVERSITAS BRAWIJAYA



BAB IV HASIL DAN PEMBAHASAN

4.1 Lingkungan Implementasi

Perangkat lunak ini akan digunakan untuk proses pengenalan wajah (*face recognition*) dengan menggunakan metode LDA yang akan digunakan untuk proses pengenalnya. Inputan yang digunakan adalah berupa citra wajah berukuran 92 x 112 berformat jpg. Perangkat lunak ini menggunakan 2 jenis data base wajah yaitu Yale dan ORL.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem pengenalan wajah dengan metode LDA adalah sebagai berikut:

1. Processor Intel(R) Core(TM)2 Duo CPU T5750 @2.00GHz.
2. Memory 2.99 Gb.
3. Harddisk 320 GB.
4. VGA Mobile Intel 965 chipset 384 Mb.
5. Monitor 14'.
6. Keyboard.
7. Mouse.

4.1.2 Lingkungan Perangkat Lunak

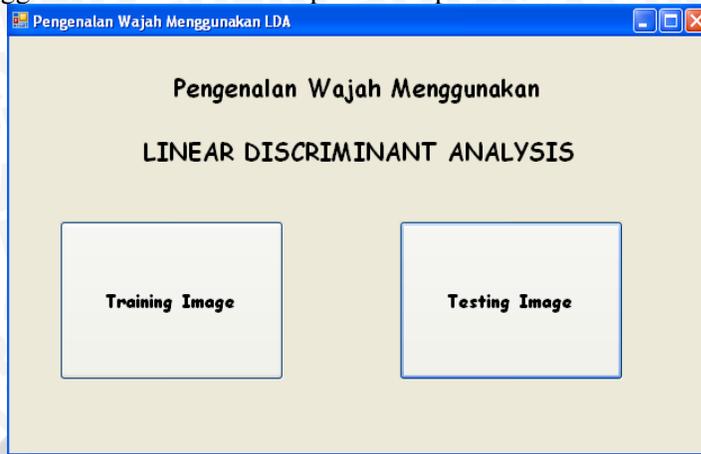
Perangkat lunak yang digunakan untuk pengembangan sistem pengenalan wajah dengan metode LDA adalah sebagai berikut:

1. Windows XP profesional.
2. Microsoft Visual Studio 2008 bahasa pemrograman C#.
3. Adobe Photoshop CS3.



4.2 Implementasi Program

Tampilan form utama untuk aplikasi sistem pengenalan wajah dengan menggunakan metode LDA dapat dilihat pada Gambar 4.1.



Gambar 4.1 Tampilan Utama Aplikasi

4.2.1 Proses *Training*

4.2.1.1 Proses PCA

PCA (*Principal Component Analysis*) merupakan proses untuk mengekstraksi fitur dari citra training. Nilai fitur ini akan digunakan untuk proses LDA selanjutnya.

Langkah pertama untuk proses ekstraksi fitur adalah melakukan konversi dari citra menjadi matriks kemudian melakukan perhitungan untuk mendapatkan nilai matriks kovarian terhadap nilai piksel citra *grayscale*. Proses ini dapat dilihat pada *sourcecode* 4.1

```
Matgray = new int[JumCit, DmsCit];

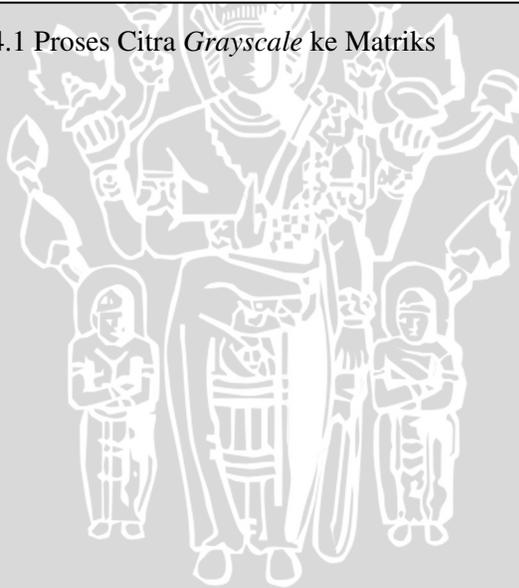
//Tampilkan ke listview   citra ke vector
for (int a = 0; a < JumCit; a++)
{
Bitmap newBitmap = new
Bitmap(listView1.Items[a].SubItems[1].Text);

int x = 0;
for (int i = 0; i < newBitmap.Width; i++)
{
for (int j = 0; j < newBitmap.Height; j++)
{

//mengambil nilai piksel dari citra asli
Color originalColor = newBitmap.GetPixel(i,
j);

//membuat nilai grayscale dari piksel
int grayScale = (int)((originalColor.R * .3) +
(originalColor.G * .59) + (originalColor.B *
.11));
```

Sourcecode 4.1 Proses Citra Grayscale ke Matriks



Kemudian mencari nilai matriks kovarian Proses ini dapat dilihat pada *sourcecode 4.2*

```
//hitung rata2 (z)
for (int i = 0; i < DmsCit; i++)
for (int j = 0; j < JumCit; j++)
rataPCA[i] += Matgray[j, i];

for (int j = 0; j < DmsCit; j++)
{
    rataPCA[j] /= JumCit;
}
// A = Kumpulan hasil pengurangan vektor citra
training dengan rata-rata data training
for (int i = 0; i < JumCit; i++)
    for (int j = 0; j < DmsCit; j++)
    {
        hasilkurangrataPCA[i, j] = Matgray[i, j]
- rataPCA[j];
    }
//perkalian matriks Xi(matkov)
matkov = new double[JumCit, JumCit];
for (int i = 0; i < JumCit; i++)
    {
        for (int j = 0; j < JumCit; j++)
        {
            double c = 0;
            for (int k = 0; k < DmsCit; k++)
            {
                c = c + Matgray[i, k] * Matgray[j, k];
            }
            matkov[i, j] = c;
        }
    }
```

Sourcecode 4.2 Proses mendapatkan matriks kovarian

Setelah mendapatkan matriks kovarian, kemudian mencari *eigen value* dan *eigen vector*. Setelah mendapatkan *eigen*, kemudian melakukan reduksi dimensi berdasarkan *eigen value* dan *eigen vector*. Kemudian mencari nilai matriks vektor V, dapat dilihat di *sourcecode* 4.3

```
for (int i = 0; i < DmsCit; i++)
{
    ListViewItem tampil = new ListViewItem("");
    for (int j = 0; j < jumlahkomponen; j++)
    {
        double f = 0;
        for (int k = 0; k < JumCit; k++)
        {
            f = f + Matgray[k, i] * Eigenvector[k,
j];
        }
        vectorVpca[i, j] = f;
    }
}
```

Sourcecode 4.3 Proses matriks V PCA

Setelah mendapatkan matriks V PCA, Kemudian menghitung fitur PCA, Proses fitur PCA (reduksi dimensi) dapat dilihat pada *sourcecode* 4.4

```
for (int i = 0; i < JumCit; i++)
{
    ListViewItem tampil = new ListViewItem("");
    for (int j = 0; j < jumlahkomponen; j++)
    {
        double f = 0;
        for (int k = 0; k < DmsCit; k++)
        {
            f = f + hasilkurangrataPCA[i, k] *
vectorVpca[k, j];
        }
        fitureigenPCA[i, j] = f;
    }
}
```

Sourcecode 4.4 Proses fitur PCA

4.2.1.2 Proses LDA

Proses *training* LDA merupakan proses pelatihan terhadap sistem dengan memasukkan hasil ekstraksi fitur PCA yang menghasilkan fitur LDA.

Langkah pertama yaitu menghitung nilai *scatter within* (SW) dan *scatter between* (SB). Dapat dilihat pada *sourcecode* 4.5

```
//sw
if (dimension == 1)
{
    cov = new double[cols, cols];
    for (int i = 0; i < cols; i++)
    {
        for (int j = i; j < cols; j++)
        {
            double s = 0.0;
            for (int k = 0; k < rows; k++)
                s += (matrix[k, j] - mean[j]) *
                    (matrix[k, i] - mean[i]);
            s /= divide;
            cov[i, j] = s;
            cov[j, i] = s;
        }
    }
}
else
```



```

//sb
{
cov = new double[rows, rows];
    for (int i = 0; i < rows; i++)
    {
        for (int j = i; j < rows; j++)
        {
            double s = 0.0;
            for (int k = 0; k < cols; k++)
                s += (matrix[j, k] - mean[j]) *
(matrix[i, k] - mean[i]);
            s /= divide;
            cov[i, j] = s;
            cov[j, i] = s;
        }
    }
}

```

Sourcecode 4.5 Proses scatter within (SW) dan scatter between (SB)

Kemudian mencari nilai matriks kovarian Proses ini dapat dilihat pada sourcecode 4.6

```

for (int i = 0; i < JumCit; i++)
{
    for (int j = 0; j < jumlahkomponen; j++)
    {
        double b = 0;
        for (int k = 0; k < jumlahkomponen; k++)
        {
            b = b + SB[i,k] * swinvr[k,j];
        }
        Covar[i, j] = b;
    }
}

```

Sourcecode 4.6 Proses matriks kovarian LDA

Menghitung matriks V LDA, dapat dilihat pada *sourcecode 4.7*

```
for (int i = 0; i < JumCit; i++)
{
    for (int j = 0; j < jumlahkomponen; j++)
    {
        double f = 0;
        for (int k = 0; k < jumlahkomponen; k++)
        {
            f = f + fitureigenPCA[i, k] *
EigenVec[k, j];
        }
        vectorVlda[i, j] = f;
    }
}
```

Sourcecode 4.7 Proses matriks V LDA

Setelah mendapatkan matriks V LDA, Kemudian menghitung fitur LDA, Proses fitur LDA dapat dilihat pada *sourcecode 4.8*

```
for (int i = 0; i < JumCit; i++)
{
    for (int j = 0; j < jumlahkomponen; j++)
    {
        double f = 0;
        for (int k = 0; k < JumCit; k++)
        {
            f = f + hasilkurangrataLDA[i, k] *
vectorVlda[k, j];
        }
        fituregenLDA[i, j] = f;
    }
}
```

Sourcecode 4.8 Proses fitur LDA

4.2.2 Proses Pengenalan

Pada proses pengenalan dilakukan perhitungan jarak terdekat antara fitur *training* dengan fitur uji. Sebelum melakukan perhitungan jarak terdekat, menghitung fitur dari data uji. Proses perhitungan fitur data uji dimulai dengan melakukan konversi dari citra menjadi matriks, dapat dilihat pada *sourcecode* 4.9

```
MatgrayTes = new double[JumCitTes,
DmsCitTes];
for (int a = 0; a < JumCitTes; a++)
{
    Bitmap newBitmap = new
    Bitmap(listView1.Items[a].SubItems[1].Text);
    int x = 0;
    for (int i = 0; i < newBitmap.Width; i++)
    {
        for (int j = 0; j < newBitmap.Height; j++)
        {
            Color originalColor = newBitmap.GetPixel(i,
            j);
            int grayScale = (int)((originalColor.R * .3)
            + (originalColor.G * .59) + (originalColor.B
            * .11));
            MatgrayTes[a, x++] = grayScale;
        }
    }
}
```

Sourcecode 4.9 Proses Citra *Grayscale* ke Matriks

Kemudian menghitung fitur data uji, dapat dilihat pada *sourcecode*

4.10

```
for (int i = 0; i < JumCitTes; i++)
{
    for (int j = 0; j < jumlahkomponentes;
j++)
    {
        double f = 0;
        for (int k = 0; k < DmsCitTes; k++)
        {
            f = f + hasilkurangratatesPCA[i, k] *
vectorVpca[k, j];
        }
        fitureigentesPCA[i, j] = f;
    }
    for (int i = 0; i < JumCitTes; i++)
    {
        for (int j = 0; j < jumlahkomponentes;
j++)
        {
            double f = 0;
            for (int k = 0; k < JumCitTes; k++)
            {
                f = f + hasilkurangratatesLDA[i, k] *
vectorVlda[k, j];
            }
            fitureigentesLDA[i, j] = f;
        }
    }
}
```

Sourcecode 4.10 Proses fitur data uji.

Setelah diperoleh fitur data uji, kemudian menghitung jarak antara fitur *training* dengan fitur uji. Proses dapat dilihat pada *sourcecode* 4.11

```

for (int i = 0; i < JumCitTes; i++)
{
    qq = mat.GetRow(fitureigentesLDA, i);
    for (int j = 0; j < JumCit ; j++)
    {
        ww = mat.GetRow(fitureigenLDA, j);
        hasilakhir[i, j] = mat.Euclidean(qq,
ww);
    }
}
public double SquareEuclidean(double[] a,
double[] b)
{
    double d = 0.0;
    for (int i = 0; i < a.Length; i++)
    {
        double u = a[i] - b[i];
        d += u * u;
    }
    return d;
}
public double Euclidean( double[] a, double[]
b)
{
    return
System.Math.Sqrt(SquareEuclidean(a, b));
}

```

Sourcecode 4.11 Proses perhitungan jarak terdekat

4.3 Penerapan Aplikasi

Aplikasi pengenalan wajah menggunakan LDA (*Linear Discriminant Analysis*) ini menggunakan data set (data *training* dan data *testing*) menggunakan gambar *grayscale* berformat jpg dengan ukuran citra 92 x 112 piksel.

Proses *training* PCA (*Principal Component Analysis*) dilakukan dengan data citra *grayscale* berformat jpg dengan ukuran 92 x 112 piksel. Pada *training* PCA dilakukan perhitungan mencari matrik kovarian, kemudian digunakan untuk menghitung *eigen value* dan *eigen vector*.

Setelah itu menghitung fitur PCA. Tampilan *training* PCA dapat dilihat pada Gambar 4.2

The screenshot shows the 'Training Image' software interface. The 'PCA' tab is selected, displaying the following data:

Eigen Value							
1	2	3	4	5	6	7	8
412956..	61296..	43722..	31443..	18674..	17744..	13474..	118186..

Eigen Vector							
1	2	3	4	5	6	7	8
0.21346..	0.1769..	0.1400..	0.1452..	-0.129..	0.1391..	-0.1731..	-0.407..

Fitur PCA								
Nama	Fitur1	Fitur2	Fitur3	Fitur4	Fitur5	Fitur6	Fitur7	Fitur8
A1.jpg	58122..	110662..	59663..	42516..	-25641..	24423..	-23352..	-46761.. 971

Gambar 4.2 Tampilan form Training PCA

Pada perhitungan *training* PCA dihasilkan fitur PCA yang digunakan sebagai input di LDA. Pada LDA dilakukan perhitungan untuk mencari nilai matrik *scatter within* (SW) dan *scatter between* (SB). Kemudian dilakukan perhitungan mencari matrik kovarian, kemudian digunakan untuk menghitung *eigen value* dan *eigen vector*. Setelah itu menghitung fitur LDA. Hasil perhitungan LDA adalah fitur LDA, dapat dilihat pada Gambar 4.3

The screenshot shows the 'Training Image' software interface. The 'LDA' tab is selected, displaying the following data:

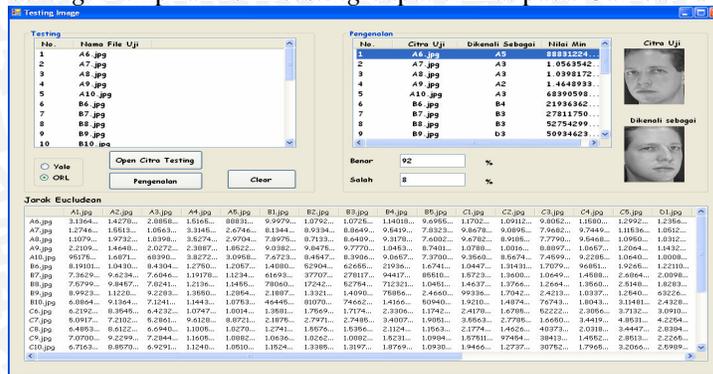
Scatter Within								
	1	2	3	4	5	6	7	8
A1.jpg	3.0825..	95063..	56287..	-28036..	-48699..	-78064..	70736..	1073..

Scatter Between								
	1	2	3	4	5	6	7	8
A1.jpg	4.4080..	2.3503..	-2.385..	-2.149..	-1.470..	-98037..	-1.934..	-8.75..

Fitur LDA								
Nama	Fitur1	Fitur2	Fitur3	Fitur4	Fitur5	Fitur6	Fitur7	Fitur8
A1.jpg	-88489..	346169..	-22913..	-21077..	131147..	-95010..	-38111..	-64598..

Gambar 4.3 Tampilan form Training LDA

Setelah melalui tahapan *training* LDA, selanjutnya dapat dilakukan pengujian. Data uji mempunyai format, jumlah dan ukuran sama dengan data *training*. Tampilan form *testing* dapat dilihat pada Gambar 4.4.



Gambar 4.4 Tampilan form *testing* citra.

4.4 Skenario Pengujian

Pada sistem pengenalan wajah dengan menggunakan metode *LDA* (*Linear Discriminant Analysis*) akan dilakukan proses pengujian. Pengujian dilakukan dengan menggunakan dua jenis database wajah, yaitu ORL dan Yale. Pengujian bertujuan untuk mengetahui tingkat akurasi metode pengenalan ini cocok digunakan untuk database ORL atau Yale.

4.4.1 Data Pengujian

Database yang digunakan untuk pengenalan wajah menggunakan metode *LDA* ada dua, yaitu ORL database dan Yale database.

Citra ORL merupakan standar wajah satu kepala. Wajah ini memiliki variasi pencahayaan yang berbeda-beda, ekspresi yang berbeda (mata terbuka/tertutup, senyum/tidak senyum, menengok kesamping) dan aksesoris berupa kacamata. Latar pada tiap citra berwarna gelap.

Citra ORL yang digunakan adalah 50 citra, 25 untuk citra *training* dan 25 untuk citra uji. Data uji yang digunakan merupakan data citra berformat jpg dan memiliki ukuran 92 x 112 piksel, serta berbentuk citra *grayscale*. Untuk citra *training* menggunakan 5 individu, dan setiap individu memiliki 5 pose yang berbeda. Untuk data uji menggunakan 5 individu, dan setiap individu memiliki 5 pose yang berbeda.

Citra Yale merupakan standar wajah satu kepala. Wajah ini memiliki variasi pencahayaan yang berbeda-beda, ekspresi yang berbeda (mata

terbuka/tertutup, senyum/tidak senyum) dan aksesoris berupa kacamata. Latar pada tiap citra berwarna terang.

Citra Yale yang digunakan adalah 50 citra, 25 untuk citra *training* dan 25 untuk citra uji. Data uji yang digunakan merupakan data citra berformat jpg dan memiliki ukuran 92 x 112 piksel, serta berbentuk citra *grayscale*. Untuk citra *training* menggunakan 5 individu, dan setiap individu memiliki 5 pose yang berbeda. Untuk citra uji menggunakan 5 individu, dan setiap individu memiliki 5 pose yang berbeda.

4.4.2 Hasil Pengujian

Hasil pengujian aplikasi pengenalan wajah dengan menggunakan metode LDA didapatkan dari hasil skenario pengujian yang telah disebutkan pada subbab sebelumnya.

Tabel 4.1 Hasil pengujian ORL

Kelas	Citra Uji	Dikenali Sebagai	Akurasi (%)
A	A6	A	100
	A7	A	
	A8	A	
	A9	A	
	A10	A	
B	B6	B	80
	B7	B	
	B8	B	
	B9	D	
	B10	B	
C	C6	C	80
	C7	D	
	C8	C	
	C9	C	
	C10	C	
D	D6	D	100
	D7	D	
	D8	D	
	D9	D	

	D10	D	
E	E6	E	100
	E7	E	
	E8	E	
	E9	E	
	E10	E	
		Rata-rata akurasi	92

Tabel 4.2 Hasil pengujian YALE

Kelas	Citra Uji	Dikenali Sebagai	Akurasi (%)
A	A6	D	60
	A7	A	
	A8	A	
	A9	B	
	A10	A	
B	B6	D	80
	B7	B	
	B8	B	
	B9	B	
	B10	B	
C	C6	D	80
	C7	C	
	C8	C	
	C9	C	
	C10	C	
D	D6	D	100
	D7	D	
	D8	D	
	D9	D	
	D10	D	
E	E6	D	80

	E7	E	
	E8	E	
	E9	E	
	E10	E	
		Rata-rata akurasi	80

Pada pengujian menggunakan ORL dihasilkan akurasi sebesar 92%, sedangkan menggunakan Yale dihasilkan akurasi sebesar 80%.

4.5 Analisa Hasil

Berdasarkan data pengujian yang dilakukan, maka akan didapatkan nilai perubahan akurasi terhadap setiap database wajah yang digunakan.

Pada hasil pengujian terdapat beberapa hal yang mempengaruhi akurasi pengenalan wajah. Hasil akurasi ORL lebih bagus daripada Yale, karena pada ORL database citranya memiliki warna latar yang berwarna gelap dan memiliki wajah yang lebih besar. Sedangkan pada Yale memiliki citra dengan warna latar terang dan untuk wajahnya lebih kecil.

Pada ORL database memiliki nilai matriks *scatter within* lebih besar dibandingkan dengan *scatter within* Yale database. *Scatter within* adalah menunjukkan perbedaan variasi dalam satu individu atau kelas. Sehingga pada ORL dengan nilai *scatter within* yang besar menunjukan nilai variasi yang kecil, sebaliknya pada Yale banyak nilai *scatter within* yang kecil bahkan bernilai minus, maka pada Yale menunjukan perbedaan variasi yang besar dalam satu individu atau kelas. Untuk nilai *scatter between* pada ORL dan Yale memiliki jarak yang tidak terlalu jauh, sehingga variasi untuk antar kelasnya hampir sama.

Kesimpulannya adalah database ORL lebih baik dikarenakan variasi dalam satu individunya sangat kecil dibandingkan Yale. Pada proses pengenalan, ORL memiliki akurasi sebesar 92 % dan Yale 80%.



BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Permasalahan pengenalan wajah dapat diaplikasikan dengan menggunakan metode LDA (*Linear Discriminant Analysis*) untuk ekstraksi fitur sebagai *classifier*-nya.
2. Akurasi pengenalan wajah pada dataset ORL sebesar 92% lebih tinggi dari pada menggunakan Yale dataset yang memiliki hasil akurasi 80%, dengan menggunakan metode LDA (*Linear Discriminant Analysis*).
3. Metode LDA (*Linear Discriminant Analysis*) lebih cocok digunakan untuk pengenalan citra wajah yang memiliki latar yang gelap dan sedikit sehingga porsi wajah lebih penuh.

5.2 Saran

1. Pengenalan pola ini dapat dikembangkan lebih lanjut dengan menambahkan fuzzy sehingga dapat menambah tingkat akurasi.
2. Pada penggunaan algoritma pencarian eigen pada ekstraksi fitur lain seperti metode Jacobian, sehingga dapat menambah tingkat akurasi maupun mempercepat waktu pengenalan.

DAFTAR PUSTAKA

Johnson, Stephen, 2006, *Stephen Johnson on Digital Photography*. O'Reilly. [ISBN 059652370X](#).

Jolliffe, I.T, 2002, *Principal Component Analysis*, Series: *Springer Series in Statistics*, 2nd ed., Springer, NY, XXIX, 487 p. 28 illus. [ISBN 978-0-387-95442-4](#).

Lu, Juwei., K.N. Plataniotis., and A.N. Venetsanopoulos, 2003, *Face Recognition Using LDA Based Algorithms*, Department of Electrical and Computer Engineering University of Toronto, Canada.

ORL Face Database

<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>.

Rahman, Muhammad Aditya, 2001, *Sistem Pengenalan Wajah Menggunakan Webcam Untuk Absensi Dengan Metode Template Matching*, Politeknik Elektronika Negeri Surabaya, Surabaya.

Turk, M. dan A. Pentland. 1991, *Eigenfaces for recognition*. Journal of Cognitive Neuroscience, 3:71–86.

Shaw, PJA, 2003, *Multivariate statistics for the Environmental Sciences*, Hodder-Arnold.

Yale Facedatabase <http://cvc.yale.edu/projects/yaleface/yalefaces.html>

Yang, Wankuo., Hui Yan., Jianguo Wang., Jingyu Yang, 2008, *Face Recognition Using Complete Fuzzy LDA*, School of Computer Science and Technology, Nanjing University of Science and Technology, China.

LAMPIRAN

Lampiran 1

1. Citra training

Data *training* berasal dari database ORL dan Yale. ORL 25 citra yang terdiri dari 5 individu, tiap individu memiliki 5 ekspresi yang berbeda.

A. ORL database

Kelas	Label	Gambar
A	A1	
	A2	
	A3	
	A4	
	A5	
B	B1	
	B2	
	B3	
	B4	



	B5	
C	C1	
	C2	
	C3	
	C4	
	C5	
D	D1	
	D2	
	D3	
	D4	
	D5	
E	E1	

E2	
E3	
E4	
E5	

B. Yale database

Kelas	Label	Gambar
A	A1	
	A2	
	A3	
	A4	
	A5	
B	B1	
	B2	



	B3	
	B4	
	B5	
C	C1	
	C2	
	C3	
	C4	
	C5	
D	D1	
	D2	
	D3	
	D4	

	D5	
E	E1	
	E2	
	E3	
	E4	
	E5	

2. Citra uji.

Data uji berasal dari database ORL dan Yale. ORL 25 citra yang terdiri dari 5 individu, tiap individu memiliki 5 ekspresi yang berbeda.

A. ORL database.

Kelas	Label	Gambar
A	A6	
	A7	
	A8	
	A9	

	A10	
B	B6	
	B7	
	B8	
	B9	
	B10	
C	C6	
	C7	
	C8	
	C9	
	C10	
D	D6	

	D7	
	D8	
	D9	
	D10	
E	E6	
	E7	
	E8	
	E9	
	E10	

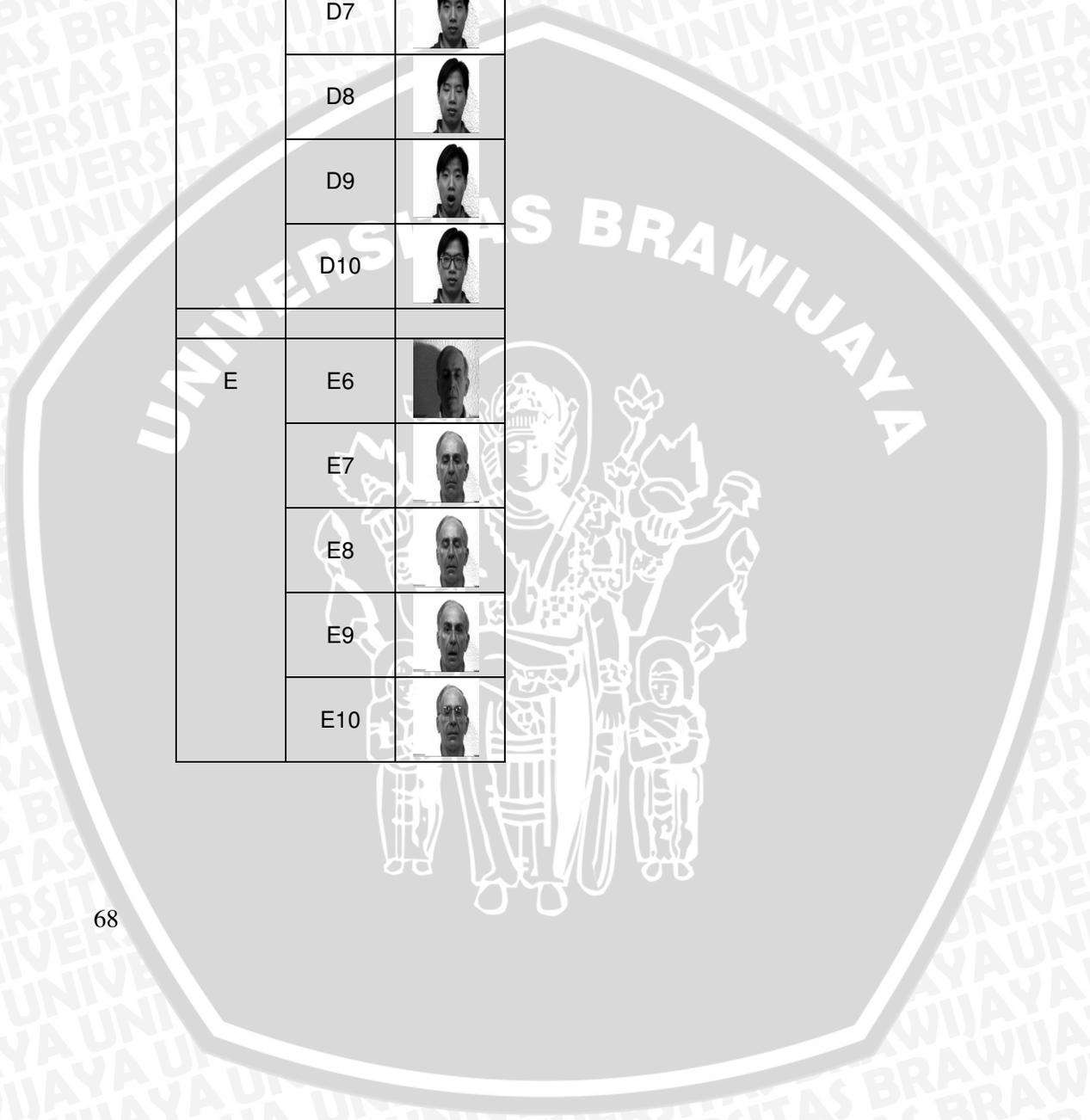
B. Yale database

Kelas	Label	Gambar
A	A6	
	A7	



	A8	
	A9	
	A10	
B	B6	
	B7	
	B8	
	B9	
	B10	
C	C6	
	C7	
	C8	
	C9	

	C10	
D	D6	
	D7	
	D8	
	D9	
	D10	
E	E6	
	E7	
	E8	
	E9	
	E10	



Lampiran 2

Hasil dari pengenalan wajah menggunakan metode LDA.
Menggunakan ORL database 25 citra dan Yale database 25 citra.

A. ORL database

Kelas	Citra Uji	Dikenali Sebagai	Keterangan	Akurasi (%)
A	A6 	A5 	Benar	100
	A7 	A3 	Benar	
	A8 	A3 	Benar	
	A9 	A2 	Benar	
	A10 	A3 	Benar	
B	B6 	B4 	Benar	80
	B7 	B3 	Benar	
	B8 	B3 	Benar	

	B9 	D3 	Salah	
	B10 	B1 	Benar	
C	C6 	C3 	Benar	80
	C7 	D4 	Salah	
	C8 	C3 	Benar	
	C9 	C3 	Benar	
	C10 	C3 	Benar	
D	D6 	D5 	Benar	100
	D7 	D4 	Benar	

70



	D8 	D2 	Benar	
	D9 	D2 	Benar	
	D10 	D4 	Benar	
E	E6 	E5 	Benar	100
	E7 	E1 	Benar	
	E8 	E5 	Benar	
	E9 	E5 	Benar	
	E10 	E5 	Benar	
		Rata-rata akurasi		92



B. Yale database

Kelas	Citra Uji	Dikenali Sebagai	Keterangan	Akurasi (%)
A	A6 	D1 	Salah	60
	A7 	A2 	Benar	
	A8 	A2 	Benar	
	A9 	B1 	Salah	
	A10 	A2 	Benar	
B	B6 	D1 	Salah	80
	B7 	B3 	Benar	
	B8 	B5 	Benar	
	B9 	B2 	Benar	

	B10 	B2 	Benar		
C	C6 	D1 	Salah	80	
	C7 	C5 	Benar		
	C8 	C5 	Benar		
	C9 	C3 	Benar		
	C10 	C2 	Benar		
	D	D6 	D1 		Benar
D7 		D3 	Benar		
D8 		D3 	Benar		
D9 		D3 	Benar		



				
	D10	D2		
			Benar	
E	E6	D1	Salah	
				
	E7	E2	Benar	
				
	E8	E2	Benar	80
				
	E9	E2	Benar	
				
	E10	E2	Benar	
				
		Rata-rata akurasi		80