

**PENERAPAN *PRINCIPAL COMPONENT ANALYSIS* UNTUK
IDENTIFIKASI IRIS MATA**

SKRIPSI



**Disusun Oleh :
RATIH YANUARI (09.18.069)**

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

LEMBAR PERSETUJUAN DAN PENGESAHAN

PENERAPAN *PRINCIPAL COMPONENT ANALYSIS* UNTUK
IDENTIFIKASI IRIS MATA

SKRIPSI

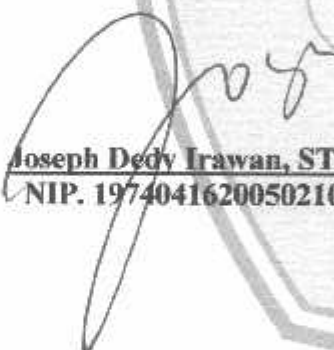
*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelara Sarjana Teknik Informatika Strata Satu (S-1)*


Disusun Oleh :
Ratih Yanuari
09.18.069

Diperiksa dan disetujui oleh


Dosen Pembimbing I

Dosen Pembimbing II


Joseph Dedy Irawan, ST, MT
NIP. 197404162005021002


Febriana Santi Wahyuni, SKom, MKom
NIP. P. 1031000425

Mengetahui,
Ketua Jurusan Teknik Informatika S-1


Joseph Dedy Irawan, ST, MT
NIP. 197404162005021002

JURUSAN TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013





PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
MALANG

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : Ratih Yanuari
Nim : 09.18.069
Program Studi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul:

**“PENERAPAN *PRINCIPAL COMPONENT ANALYSIS* UNTUK
IDENTIFIKASI IRIS MATA”**

Adalah Skripsi saya sendiri bukan duplikat serta mengutip atau menyadur seluruhnya karya orang lain kecuali dari sumber aslinya.

Malang, 1 Agustus 2013

Yang membuat pernyataan



Ratih Yanuari

KATA PENGANTAR

Puji dan Syukur penyusun panjatkan kehadiran Tuhan Yang Maha Esa, karena telah memberikan rahmat-Nya sehingga penyusun dapat menyelesaikan skripsi dengan judul PENERAPAN PRINCIPAL COMPONENT ANALYSIS UNTUK IDENTIFIKASI IRIS MATA sesuai dengan waktu yang ditentukan.

Penyusunan skripsi ini merupakan salah satu persyaratan untuk menyelesaikan program pendidikan Strata Satu (S-1) Teknik Informatika, Fakultas Teknologi Industri di Institut Teknologi Nasional Malang.

Pada kesempatan ini penyusun mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ir. Soeparno, MT selaku Rektor ITN Malang.
2. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri ITN Malang.
3. Joseph Dedy Irawan, ST, MT selaku Ketua Jurusan Teknik Informatika ITN Malang.
4. Joseph Dedy Irawan, ST, MT selaku dosen pembimbing I.
5. Febriana Santi Wahyuni, SKom, MKom selaku dosen pembimbing II.
6. Semua pihak yang telah membantu terselesaikannya skripsi ini.

Penyusun menyadari bahwa skripsi masih jauh dari sempurna, oleh karena itu penyusun mengharapkan kritik dan saran dari pembaca. Semoga skripsi ini bisa bermanfaat bagi para pembaca sekalian.

Malang, Agustus 2013

Penyusun

PENERAPAN PRINCIPAL COMPONENT ANALYSIS UNTUK IDENTIFIKASI IRIS MATA

Ratih Yanuari

Program Studi Teknik Informatika S-1
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Jl. Raya Karanglo Km 2 Malang
Email: yanuariratih@rocketmail.com

Dosen Pembimbing: 1. Joseph Dedy Irawan, ST, MT
2. Febriana Santi Wahyuni, SKom, MKom

Abstrak

Iris mata adalah membran mata, yang berfungsi untuk mengatur diameter dan ukuran pupil serta mengatur besar kecilnya intensitas cahaya yang masuk ke dalam retina. Pola iris mata mulai terbentuk sejak bulan ketiga kehamilan melalui proses yang dikenal dengan kekacauan monogenesis (chaotic morphogenesis) yakni perkembangan dan reaksi jaringan secara random terhadap kondisi-kondisi lingkungan yang berubah. Hal tersebut menimbulkan ide untuk membuat suatu teknik identifikasi biometric berdasarkan ciri alami manusia dengan iris mata yang memiliki keunikan tersendiri.

Metode-metode pengenalan citra sudah banyak dikembangkan dan diaplikasikan. Dalam tugas akhir ini dikembangkan teknik pengenalan citra dengan Principal Component Analysis, dimana beberapa citra akan dikenal bila sama atau mirip sesuai dengan citra yang ada di basis-data. Perancangan menggunakan program computer dengan menggunakan bahasa pemrograman Matlab. Hasil rancangan program digunakan untuk menguji metode PCA dengan menggunakan sejumlah citra iris mata dan menggunakan jarak euclidean sebagai metode pengenalan.

Adapun pengujian berdasarkan jumlah citra latih yaitu 10 citra latih, 20 citra latih dan 25 citra latih dimana hasil pengenalan yang sesuai antara citra yang sudah dilatih dengan citra yang diujikan dan memiliki prosentase terbesar terdapat pada pengujian dengan 25 citra latih sebesar 92%.

Kata kunci : citra digital, PCA, Euclidean

DAFTAR ISI

Halaman Judul	i
Lembar Persetujuan dan Pengesahan	ii
Lembar Keaslian.....	iii
Abstrak	iv
Kata Pengantar.....	v
Daftar Isi	vi
Daftar Gambar	ix
Daftar Tabel	xi
Bab I Pendahuluan	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metode Penelitian	3
1.6 Sistematika Penulisan	4
Bab II Landasan Teori.....	5
2.1 Biometrik.....	5
2.1.1 Keunggulan Biometrik	5
2.1.2 Karakteristik	6
2.1.3 Sistem Biometrik	6
2.2 Citra mata dan iris mata	7
2.3 Iris Biometrik.....	8
2.4 Pengolahan Citra Digital	9
2.5 Jenis Citra Digital	11
2.6 PCA	12
2.7 Varians dan Standart Deviasi.....	14
2.8 Standarisasi data	16
2.9 Covariance.....	16

2.10 EVD (Eigenvalue Decomposition)	17
2.11 Euclidean Distance.....	19
2.12 Matlab	20
2.12.1 Dekstop Tool Matlab	21
2.12.2 Graphic User Interface	22
 Bab III Perancangan	24
3.1 Analisa Perancangan Sistem.....	24
3.2 Diskripsi Perancangan Sistem	24
3.3 Analisa Perancangan Data.....	24
3.4 Analisa Kebutuhan Sistem	25
3.4.1 Perangkat Keras (Hardware)	25
3.4.2 Perangkat Lunak (Software).....	25
3.5 Analisa Arsitektur Sistem.....	26
3.6 Diagram Blok	26
3.6.1 Diagram Blok Proses Training.....	26
3.6.2 Diagram Blok Proses Pengujian	28
3.7 Diagram Alir (Flowchart).....	30
3.7.1 Flowchart Sistem	30
3.7.2 Flowchart Aplikasi	31
3.8 Algoritma Aplikasi.....	33
3.9 Tahap Pengambilan Citra Mata	34
3.10 Tahap Pra-proses Citra	34
3.11 Tahap Ekstraksi Ciri.....	35
3.12 Perancangan Menu Sistem	36
3.13 Perancangan Desain Aplikasi	37
3.13.1 Perancangan Tampilan Halaman Utama	37
3.13.2 Perancangan Tampilan Proses Citra.....	38
3.13.3 Perancangan Tampilan Pengujian	41
3.13.4 Perancangan Tampilan Profile	42
3.13.5 Perancangan Tampilan Sistem Kerja.....	43

Bab IV Implementasi dan Pengujian	45
4.1 Implementasi	45
4.2 Implementasi User Interface	45
4.2.1 Implementasi Halaman Utama	45
4.2.2 Implementasi Proses Citra	46
4.2.3 Implementasi Pengujian	48
4.2.4 Implementasi Profile	49
4.2.1 Implementasi Sistem Kerja.....	50
4.3 Implementasi Program	50
4.3.1 Implementasi Proses Ambil Citra	51
4.3.2 Implementasi Proses Citra	52
4.3.3 Implementasi Proses Pengujian	54
4.4 Pengujian Aplikasi	55
4.4.1 Pengaruh Jumlah Citra Latih	55
4.4.2 Pengaruh Pengnalan ID Citra Uji	59
 Bab V Penutup.....	 62
5.1 Kesimpulan.....	62
5.2 Saran	63
 Daftar Pustaka	 64
Lampiran	65

DAFTAR GAMBAR

Gambar 2.1 Bagan karakteristik biometrik	6
Gambar 2.2 Sistem pada proses biometrik	7
Gambar 2.3 Struktur mata	8
Gambar 2.4 Citra iris mata	8
Gambar 2.5 Proses iris scan	9
Gambar 2.6 Langkah-langkah pengolahan citra digital	10
Gambar 2.7 Langkah-langkah PCA	12
Gambar 2.8 Contoh matrik data citra	13
Gambar 2.9 Hubungan antara variabel X dan Y, Y dan Z, Z dan X.....	13
Gambar 2.10 Diagram PBO Matlab	22
Gambar 3.1 Diagram blok proses training pada sistem.....	27
Gambar 3.2 Diagram blok proses pengujian pada sistem	28
Gambar 3.3 Flowchart sistem	30
Gambar 3.4 Flowchart aplikasi pengolahan citra pada sistem	32
Gambar 3.5 Contoh citra iris mata	34
Gambar 3.6 Flowchart PCA.....	35
Gambar 3.7 Perancangan menu pada sistem.....	36
Gambar 3.8 Rancangan Halaman Utama.....	38
Gambar 3.9 Rancangan Proses Citra	39
Gambar 3.10 Rancangan Pengujian Citra.....	41
Gambar 3.11 Rancangan Profile.....	43
Gambar 3.12 Rancangan Sistem Kerja	43
Gambar 4.1 Implementasi Halaman Utama.....	46
Gambar 4.2 Implementasi Proses Citra	47
Gambar 4.3 Implementasi Pengujian.....	48
Gambar 4.4 Implementasi Profile	49
Gambar 4.5 Implementasi Sistem Kerja	50
Gambar 4.6 Proses Pengambilan Citra pada 'Proses Citra'	51
Gambar 4.7 Citra yang terdapat dalam komputer	52
Gambar 4.8 Hasil citra yang dikenali pada tampilan 'Proses Citra'	52

Gambar 4.9 Hasil citra yang tidak dikenali pada tampilan ‘Proses Citra’	53
Gambar 4.10 Hasil citra yang dikenali pada tampilan ‘Pengujian’	54
Gambar 4.11 Hasil citra yang tidak dikenali pada tampilan ‘Pengujian’	54

DAFTAR TABEL

Tabel 3.1 Komponen Tampilan Halaman Utama.....	38
Tabel 3.2 Komponen Tampilan Proses Citra.....	39
Tabel 3.3 Komponen Tampilan Pengujian.....	42
Tabel 3.4 Komponen Tampilan Profile	43
Tabel 3.5 Komponen Tampilan Sistem Kerja	44
Tabel 4.1 Tabel Hasil Pengujian 1.....	55
Tabel 4.2 Tabel Hasil Pengujian 2.....	56
Tabel 4.3 Tabel Hasil Pengujian 3.....	57
Tabel 4.4 Tabel Hasil Pengujian Akurasi Identitas	60

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi informasi pada saat ini telah berubah dengan sangat cepat seiring dengan kemajuan zaman. Perkembangannya hampir membuat setiap individu di dunia memerlukan komputer sebagai alat bantu menyelesaikan berbagai permasalahan. Penggunaan sistem analog atau manual kini sudah tergantikan dengan adanya sistem komputerisasi. Dalam sistem komputerisasi memiliki hasil lebih cepat, lebih mudah dilakukan pengontrolan dan juga memiliki hasil yang tidak berubah-ubah. Salah satu contohnya adalah dengan memanfaatkan anggota tubuh secara unik untuk membedakan antara satu orang dengan orang lain, yang biasa disebut biometrik. Biometrik adalah suatu metode untuk mengenali manusia berdasarkan pada satu atau lebih ciri-ciri fisik atau tingkah laku yang unik.

Biometrik sebagai salah satu sistem identifikasi dan verifikasi sebenarnya bukanlah suatu hal yang baru. Sistem biometrik menggunakan ciri-ciri fisiologis (*physiological*) seseorang dan tingkah laku seseorang perilaku (*behavioral*) untuk mengenali atau membuktikan identitas seseorang. Yang menggunakan ciri-ciri fisik misalnya wajah (*face recognition*), sidik jari (*fingerprint*), iris mata (*iris recognition*) dan geometri tangan (*hand geometry*). Sedangkan yang menggunakan perilaku/tingkah laku misalnya tulisan tangan, suara (*voice recognition*) dan tanda tangan (*signature*). Salah satu penggunaan sistem biometric sebagai salah satu sistem identifikasi yang telah berkembang dengan pesat adalah iris mata. Iris mata adalah daerah berbentuk gelang pada mata yang dibatasi oleh pupil dan sclera (bagian putih dari mata). Kelebihan iris mata adalah iris memiliki tingkat penerimaan kesalahan sangat kecil. Oleh karena itu untuk mempermudah proses pengidentifikasian dibuatlah sebuah aplikasi untuk mengenali iris mata berbasis matlab yang diharapkan dapat bermanfaat bagi pengenalan pola iris mata sebagai

verifikasi data dengan lebih cepat dan akurat serta pada studi kasus yang diterapkan.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah yang ingin diambil dari skripsi ini adalah bagaimana membangun dan menerapkan metode PCA (*Principal Components Analysis*) yang akan digunakan untuk pengenalan suatu pola pada iris mata.

1.3 Tujuan

Tujuan dari penulisan ini adalah membuat sistem yang dapat mendeteksi pola iris mata manusia secara otomatis menggunakan metode PCA (*Principal Components Analysis*) melalui citra mata sehingga dapat membantu dalam pengenalan iris mata.

1.4 Batasan Masalah

Batasan Masalah yang digunakan diharapkan mampu membatasi pembahasan agar permasalahan tidak melebar maka pembahasan dibatasi pada hal-hal sebagai berikut :

1. Citra masukan yang digunakan dalam penelitian ini adalah citra mata dalam RGB (Red, Green, Blue). Tanpa membahas proses mendapatkan citra sebelum digunakan.
2. Ekstensi Citra yang dipakai adalah .jpeg (*Join Photographic Experts Group*) berukuran 119x89 piksel.
3. Citra masukan dalam penelitian ini adalah citra mata yang dalam keadaan sehat.
4. Proses PCA yang dilakukan menggunakan EVD (*Eigen Value Decomposition*).
5. Citra didapatkan dari Database Iris mata yang sudah disediakan oleh sebuah pusat penelitian bidang biometrik bernama Phoenix-<http://phoenix.inf.upol.cz>
6. Program bantu yang digunakan adalah Matlab R2008b.

1.5 Metode Penelitian

Adapun metode yang dipergunakan dalam penyusunan skripsi ini adalah sebagai berikut :

1. Studi literature

Merupakan serangkaian kegiatan mengumpulkan data dan referensi yang berhubungan dengan penyusunan laporan ini. Sumber yang digunakan antara lain adalah buk, jurnal dan berbagai sumber dari internet. Studi literatur lebih di fokuskan pada pengenalan pola iris mata dan penerapan PCA untuk mendeteksi nilai kecocokan iris mata.

2. Perancangan sistem

Penerapan *principal components analysis* untuk mendeteksi iris mata, dimana citra akan dikenali jika citra tersebut sama atau mirip sesuai dengan citra yang ada didalam basis-data. Perancangan sistem PCA yaitu : tahap pertama memasukkan citra yang akan citraining, kemudian melakukan standarisasi citra dengan *z-score* agar variabelnya layak diandingkan, setelah itu proses pehitungan besaran hubungan antar dua variabel dengan *covariance*. Bila sudah diketahui nilai *covariancenya* proses selanjutnya adalah mencari nilai FVD (*eigen value decomposition*) yaitu sebuah bilangan saklar yang dapat mendefinisikan matrik sebuah citra. Dengan Proses terseut nilai PCA sebuah citra sudah diketahui. Berikutnya proses *training* untuk mencocokkan citra yang sudah ditraining dengan citra yang akan ditest pada folder database iris mata yang telah disimpan dan mengklasifikasi karakteriris mata yang sesuai, untuk proses terakhir yaitu proses menampilkan iris mata yang paling sesuai.

3. Implementasi

Pada pembuatan penerapan *principal components analysis* untuk mendeteksi iris mata ini yang digunakan adalah Matlab (*Matrix Laboratory*) versi R2008b. Perangkat lunak ini dipilih karena pengolahan citra pada dasarnya merupakan pengolahan matrik, sehingga Matlab adalah perangkat lunak yang tepat dalam hal ini. Pembuatan program dibagi ke dalam dua bagian, yaitu pembuatan interface dan fungsi-fungsi pengolahan citra.

4. Pengujian sistem

Tahap pengujian dilakukan jika semua bagian sistem telah selesai dibuat. Pengujian dilakukan ke semua bagian interface sistem untuk mengetahui apakah sistem berjalan sesuai dengan yang diharapkan.

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyusunan skripsi ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Berisi tentang Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan dan Manfaat, Metodologi, dan Sistematika Penulisan dari skripsi ini.

BAB II : LANDASAN TEORI

Berisi teori-teori yang didapat dari studi literatur dan konsep-konsep yang terkait dengan tugas akhir ini, beserta dengan penyelesaian masalah yang diambil dalam penyusunan tugas akhir.

BAB III: PERANCANGAN SISTEM

Berisi tentang desain dan perancangan sistem perangkat lunak yang akan dibangun pada aplikasi.

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Berisi tentang pembahasan hasil pengujian dan analisa mengenai cara kerja secara keseluruhan sampai pada hasil yang dikeluarkan.

BAB V : PENUTUP

Berisi tentang kesimpulan hasil pengujian secara keseluruhan dari pengerjaan tugas akhir dan saran-saran untuk memperbaiki kelemahan sistem yang dibuat demi pengemangan dan penyempurnaan di waktu mendatang.

BAB II

LANDASAN TEORI

2.1 Biometrik

Biometrik adalah pengenalan identifikasi menggunakan fisik manusia. Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola yang lainya. Ciri yang bagus adalah ciri yang memiliki daya pembeda yang tinggi sehingga pengelompokan pola berdasarkan ciri yang dimiliki dapat dilakukan dengan akurat.^[1]

2.1.1 Keunggulan Biometrik

Biometrik banyak dipergunakan dalam sebuah akurasi sistem daripada cara manual atau analog dikarenakan beberapa alasan. Diantaranya :

1. Proses verifikasi sistem analog kebanyakan berdasarkan benda. Dimana semua data-data penting yang dibutuhkan berada pada suatu benda seperti dokumen atau kartu kredit. Apabila terjadi kehilangan maka orang lain dapat dengan mudah memalsukannya.
2. Proses verifikasi analog yang berdasarkan pengetahuan. Selain penyembunyian dengan sebuah benda juga berupa pengetahuan verifikasi secara personal seperti, menggunakan *password*. Hal tersebut tetap terdapat kunci yang dapat membukanya bahkan jika menggunakan algoritma enkripsi terbaik.

Karena beberapa alasan tersebut sebuah informasi mudah bocor sehingga mudah dimanfaatkan oleh pihak-pihak yang tidak bertanggung jawab untuk berbuat kejahatan. Berikut ini beberapa keunggulan menggunakan biometrik diantaranya:

1. Setiap *person* pasti memiliki sendir-sendiri dan berbeda sehingga tidak dapat hilang atau lupa.
2. Sulit di duplikasi, di-*share* ataupun dipindah tangankan.
3. Keaslian lebih terjamin karena harus menghadirkan *person* sebagai alat validasi.

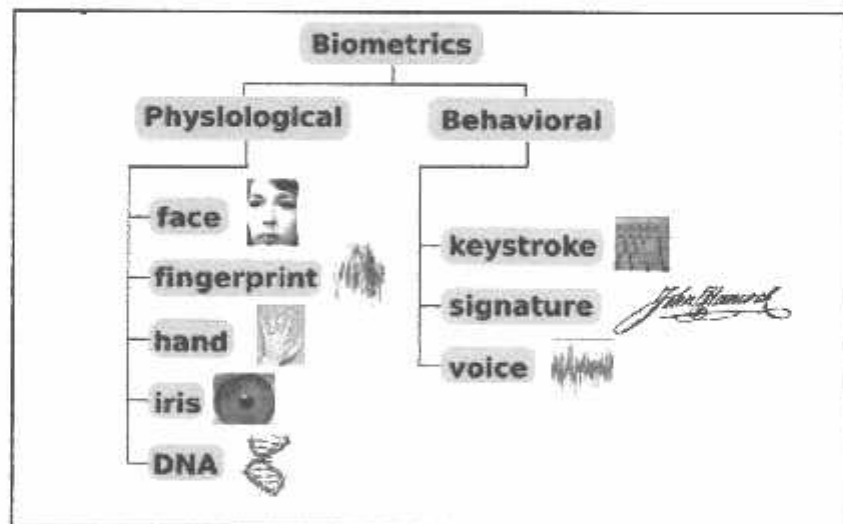
[1] Sukarnaij, Andik. *Iris dan Vein Biometric*

2.1.2 Karakteristik Biometrik

Bagian fisik manusia yang memiliki daya pembeda sangatlah banyak. Namun yang dapat memiliki keakuratan yang tinggi hanya beberapa. Untuk itu ada 2 macam karakteristik biometrik yang digunakan. Diantaranya:^[2]

1. *Physiological* : adalah karakteristik biometrik yang dihubungkan dengan bentuk tubuh/badan. Misalnya:
 - a. *Fingerprints*
 - b. *Face recognition*
 - c. *Hand geometry*
 - d. *Iris ,Retinal recognition*
2. *Behavioral* : adalah karakteristik biometrik yang dihubungkan dengan tingkah laku seseorang. Misalnya:
 - a. *Keystroke*
 - b. *Signature*
 - c. *Voice*

Adapun untuk bagan karakteristik biometrik terdapat pada gambar 2.1 berikut.



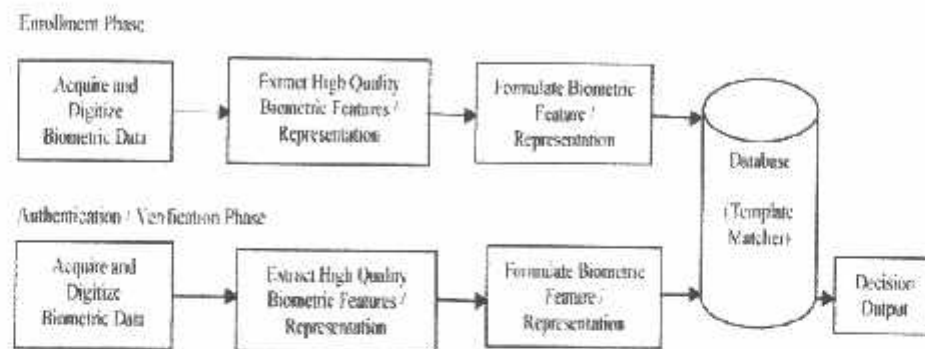
Gambar 2.1 Bagan karakteristik biometrik

2.1.3 Sistem Biometrik

Pada setiap karakteristik biometrik baik *Physiological* maupun *Behavioral* secara umum memiliki sistem yang terbagi menjadi 2 tahap yaitu *enrollment* and

[2] Sukamaji, Andik. *Iris dan Vein Biometric*

authentication/verification. Pada tahap awal biometrik *user* dilakukan *pre-processing* yaitu akuisi citra, proses pendigitalan citra dan proses penyimpanan citra sebagai basis data. Hal tersebut juga diterapkan pada tahap *authentication/verification*. Setelah itu akan mengeluarkan *output* berupa data yang telah dilakukan proses pencocokan dengan pola yang telah disimpan. Diagram untuk pengenalan sistem biometrik terdapat pada gambar 2.2. [3]



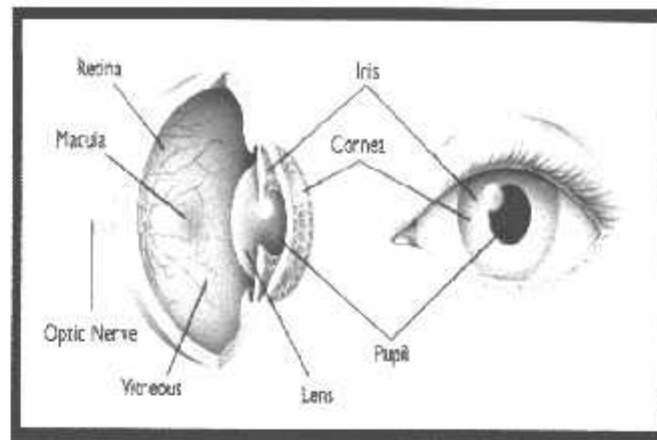
Gambar 2.2 Sistem pada proses biometrik

2.2 Citra Mata dan Iris Mata

Mata sering disebut sebagai jendela dunia karena mata merupakan organ untuk dapat melihat berbagai keadaan. Mata dapat mendeteksi cahaya yang ditimbulkan dari suatu benda sehingga benda tersebut dapat dilihat. Melalui mata pula dapat membedakan kondisi disekitar apakah terang atau gelap. Bagian-bagian pada organ mata bekerjasama mengantarkan cahaya dari sumbernya menuju ke otak untuk dapat dicerna oleh sistem saraf manusia. Bagian-bagian dari mata adalah:

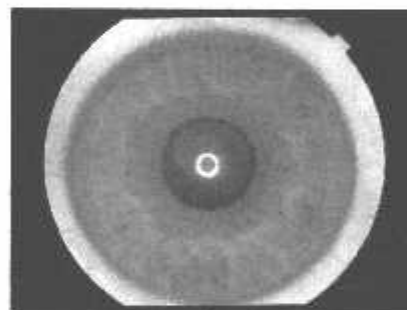
1. Kornea
2. Iris atau selaput pelangi
3. Pupil
4. Lensa mata
5. Retina
6. Sclera
7. Optik nerve
8. Verteus body

[3] Majid, Navid. 2009. *Keystroke Biometric*. School of Information Technology and Engineering (STIE)



Gambar 2.3 Struktur Mata

Iris adalah bagian yang berwarna yang tampak pada bola mata. Bagian iris terlihat sebagai lingkaran mata yang melingkupi bagian hitam pupil dengan warna-warna tertentu. Secara anatomi iris merupakan sebuah organ *internal* yang dilindungi, terletak di belakang *kornea* dan *aqueous humour*, serta berada di depan lensa mata. Bagian depan dari iris berbentuk tidak teratur, cenderung kasar serta memiliki alur yang tidak rata. Bagian ini dibentuk oleh lapisan yang terdiri dari sel pigmen dan *fibroblast*. Bagian bawah dari lapisan ini adalah jaringan ikat yang berkadar darah rendah (*poorly vascularized*) dengan beberapa serat, *fibroblast* dan *melanocyte*. Adapun gambar citra iris mata seperti gambar 2.4.^[4]



Gambar 2.4 Citra iris mata

2.3 Iris Biometric

Iris Biometric adalah teknologi untuk mengidentifikasi seseorang dengan cara mengambil citra digital beresolusi tinggi terhadap iris mata. Tekstur iris manusia berasal dari proses *chaotic morphogenetic* selama perkembangan embrio, dan

[4] Fahmi, S. T, M. Sc, *Perancangan Algoritma Pengolahan Citra Mata Menjadi Citra Polar Iris Sebagai Bentuk Antara Sistem Biometrik*

memiliki ciri yang mampu dipakai untuk identifikasi seseorang. Bila prinsip teknologi *retinal scanning* adalah memindai pola pembuluh darah kapiler pada retina dengan sumber cahaya intensitas rendah maka prinsip teknologi iris scanning adalah memindai pola warna dari mata dan pupil. Pemakaian iris dalam biometrik sistem memiliki keuntungan dan kerugian sebagai berikut:¹⁵⁾

1. Keuntungan

- Memiliki akurasi tinggi.
- Tidak memerlukan kontak dalam pengambilan data.
- Tidak mudah dipalsukan.
- Informasi iris relatif stabil, sehingga tidak diperlukan registrasi ulang.

2. Kerugian

- Memiliki harga yang mahal.
- Jika kesehatan mata terganggu, sistem tidak bisa digunakan.

Adapun proses iris scan secara umum terdapat pada gambar 2.5.

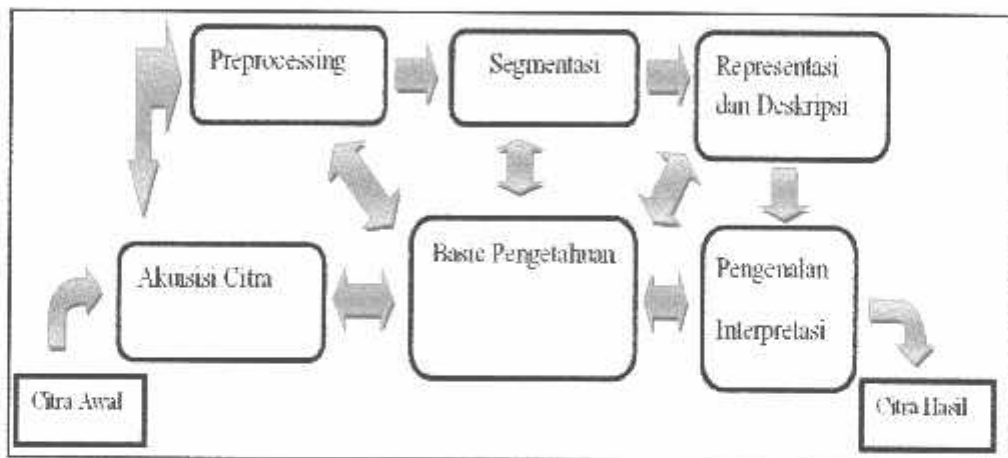


Gambar 2.5 Proses Iris Scan

2.4 Pengolahan Citra Digital

Pengolahan Citra merupakan disiplin ilmu yang bersifat multidisiplin yang memiliki banyak aspek bidang ilmu. Disiplin ilmu pengolahan citra digital berkaitan erat dengan disiplin ilmu grafika komputer dan computer vision sehingga atasan ketiganya hamper tidak jelas.

Pengertian secara lebih luas pengolahan citra digital adalah ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar (peningkatan kontras, transformasi warna, restorasi citra), transformasi gambar (rotasi, translasi, skala, transformasi geometrik), melakukan pemilihan citra citri yang optimal untuk tujuan analisis, melakukan proses penarikan informasi atau deskripsi objek, melakukan kompresi untuk penyimpanan data, transmisi data, dan waktu proses data. Input dalam pengolahan citra digital berupa citra sedangkan output-nya adalah citra hasil pengolahan. Gambar 2.6 berikut adalah langkah-langkah pengolahan citra secara umum.



Gambar 2.6 Langkah-langkah pengolahan citra digital

1. Akuisisi Citra

Tujuannya adalah untuk menentukan data yang diperlukan dan memilih metode perekaman citra digital. Tahap ini dimulai dari objek yang akan diambil gambarnya, persiapan alat-alat, sampai pada pencitraan. Pencitraan adalah kegiatan transformasi dari citra tampak menjadi citra digital. Hasil dari akuisisi citra ditentukan oleh kemampuan sensor untuk mendigitalisasi sinyal yang terkumpul pada sensor tersebut.

[6] Sutoyo, S. T. M. Sc, Mulyanto, Edy, S.Si. 2009. Teori Pengolahan Citra Digital. Penerbit Andi.

2. Preprocessing

Tahap ini diperlukan untuk menjamin kelancaran pada proses berikutnya. Proses yang dilakukan pada tahap ini antara lain : Peningkatan kualitas citra, menghilangkan noise, perbaikan citra (*Image Restoration*), transformasi, menentukan bagian citra yang akan diobservasi.

3. Segmentasi

Bertujuan untuk mempartisi citra menjadi bagian-bagian pokok yang mengandung informasi penting. Misalnya, memisahkan objek dan latar belakang.

4. Representasi dan deskripsi

Representasi merupakan suatu proses untuk merepresentasikan suatu wilayah sebagai suatu daftar titik-titik koordinat dalam kurva yang tertutup, dengan deskripsi luasan atau perimeternya. Kemudian melakukan deskripsi citra dg cara seleksi ciri dan ekstraksi ciri (*Feature Extraction and Selection*). Seleksi ciri bertujuan untuk memilih informasi kuantitatif dari ciri yg ada, yg dapat membedakan kelas-kelas object secara baik. Ekstraksi ciri bertujuan untuk mengukur besaran kuantitatif ciri setiap piksel, misalnya rata-rata, standar deviasi, koefisien variasi, *SignaltoNoise Ratio* (SNR), dan lain-lain.

5. Pengenalan dan interpretasi

Pengenalan bertujuan untuk memberi label pada sebuah objek yang informasinya disediakan oleh descriptor. Interpretasi bertujuan untuk memberi arti atau makna kepada kelompok objek-objek yang dikenali.

6. Basic Pengetahuan

Berguna untuk memandu operasi dari masing-masing modul proses dan mengontrol interaksi antara modul-modul tersebut. Selain itu juga digunakan sebagai referensi pada proses *template matching* atau pada pengenalan pola.

2.5 Jenis Citra Digital

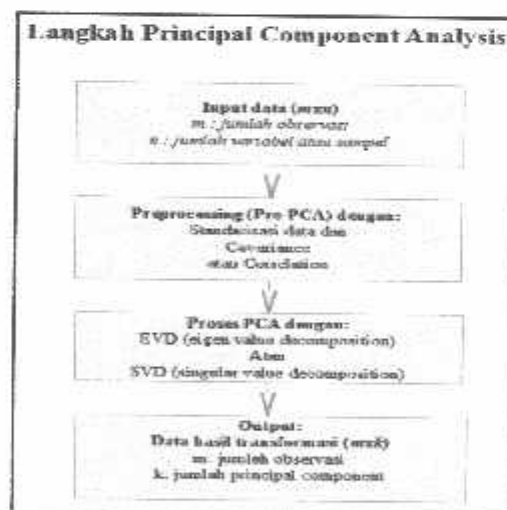
Cara penyimpanan citra digital didalam memori menentukan jenis citra digital yang terbentuk. Beberapa jenis citra digital yang sering digunakan adalah:

1. Citra biner yaitu citra yang hanya memiliki 2 warna hitam dan putih.
2. Citra *grayscale* yaitu citra yang warnanya tergantung dari ketersediaan jumlah bit didalam memori untuk menampung kebutuhan warna.
3. Citra warna yaitu merupakan citra yang memiliki kombinasi tiga warna dasar RGB (*Red Green Blue*).^[7]

2.6 PCA

Principal Component Analysis atau yang lebih populer dikenal dengan singkatannya “PCA”, adalah sebuah metode statistik yang banyak digunakan dalam sistem pengenalan pola (wajah, karakter, dan lain-lain) serta kompresi citra. Misalkan kita memiliki sebuah balok dengan informasi panjang (p), lebar (l) dan tinggi (t). Dari ketiga jenis informasi ini, bila kita olah dan kita ganti dengan informasi volume (v) dari balok tersebut, maka informasi volume dianggap sudah dapat menggambarkan panjang, lebar dan tinggi balok tadi. Karena $v = p \times l \times t$. Inilah konsep sederhana kinerja *Principal Component Analysis*.

PCA pada dasarnya bekerja pada sekelompok data observasi yang pada awalnya memiliki kemungkinan saling berelasi, kemudian proses PCA mengkonversi data tersebut sedemikian rupa sehingga yang tersisa adalah data yang tidak saling berelasi satu sama lain yang disebut dengan *Principal Component*. Gambar 2.7 merupakan langkah umum penyelesaian PCA.^[8]



Gambar 2.7 Langkah-Langkah PCA

[7] Sutoyo, S. T. M. Sc, Mulyarto, Edy, S. Si. 2009. Teori Pengolahan Citra Digital. Penerbit Andi.

[8] Muntasa, Arif. Konsep Pengolahan Citra dan Ekstraksi Ciri. Penerbit Graha Ilmu.

Pada gambar 2.9 (a) merupakan variabel X dan Y yang terlihat memiliki hubungan secara linear. Dari hubungan antara X dan Y dapat disimpulkan bahwa perubahan pada satu variabel berarti juga berpengaruh pada variabel lainnya. Walaupun besarnya perubahan tidak sama. Ini dikarenakan unit/satuan atau *range* yang berbeda. Sedangkan pada variabel X dan Z serta Y dan Z, perubahan nilainya 'seperti' tidak berhubungan secara linear. Satuan ataupun *range* yang berbeda tentu saja tidak dapat dibandingkan. Misalkan satuan Kilometer tidak dapat dibandingkan dengan satuan Celcius. Maka itu dengan keadaan data yang memiliki *range* yang berbeda sebaiknya dilakukan langkah standarisasi agar variabel-variabelnya layak untuk dibandingkan.

2) Pre-PCA

Dalam Pre-PCA ada beberapa tahapan yang dilakukan. Pertama proses standarisasi yaitu proses untuk mengukur seberapa jauh sebaran data secara keseluruhan yang direpresentasikan ke dalam suatu bilangan. Setelah langkah standarisasi proses selanjutnya menghitung nilai *covariance*.

3) Proses PCA

Data kemudian ditransformasi dari data asal ke variabel yang saling tidak berkorelasi dengan menggunakan *eigenvector* dari matriks korelasi. Cara menentukan *eigenvector* dapat dilakukan dengan 2 cara, yaitu: *Eigenvalue Decomposition* atau *Singular Value Decomposition*. Dalam studi kasus ini menggunakan EVD, dikarenakan hasil matrik *covariance* yang sudah distandarisasi adalah matrik simetris.

4) Output

Output yang dihasilkan diharapkan proses PCA dapat *men-reduce* ukuran matriks asal $m \times n$ ke ukuran $m \times k$ dengan tetap mempertahankan variasi data, yaitu data yang tidak redundan/tidak berkorelasi dan tidak menghilangkan informasi yang penting pada citra.

2.7 Varians dan Standar Deviasi

Untuk mengukur seberapa jauh sebaran data secara keseluruhan, perlu direpresentasikan ke dalam suatu bilangan. Dua diantaranya adalah dengan

mengukur menggunakan *varians* dan standar deviasi. Varians adalah kuadrat dari standar deviasi. Maka untuk mencari standar deviasi perlu dilakukan langkah pencarian nilai *varians*.

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2 \quad \dots\dots\dots (2.1)$$

Standart deviasi

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad \dots\dots\dots (2.2)$$

Dimana:

N : Jumlah variable,

x : Data obscrvasi,

μ : Mean per variable,

σ : Standar deviasi per variabel

Hasil yang diharapkan dari langkah tersebut, adalah:

1. Sebuah transformasi linear yang dilakukan menggunakan koordinat sistem yang baru (*principal component* terpilih).
 2. Sumbu-sumbu koordinat baru ini nantinya harus orthogonal (saling tegak lurus) karena mangandung informasi variabel yang tidak saling berkorelasi.
 3. Jumlah principal component yang dipilih adalah dimensi dari koordinat yang baru.
 4. Masing-masing sumbu koordinat yang dipilih harus meminimalisir jumlah dari kuadrat *error* dari sumbu tersebut ke setiap data *point*.
-

- c. *Covariance Zero* : yaitu $\text{cov}(x,y) = 0$ dimana kedua variable x dan variable y tidak berelasi linear antara satu sama lainnya.

Formula untuk menghitung *covariance* untuk data sampel :

$$\text{cov}(x,y) = \sum (x,y) - \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \quad \dots\dots\dots (2.4)$$

Formula untuk data populasi adalah:

$$\text{cov}(x,y) = \frac{1}{n} \sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y) \quad \dots\dots\dots (2.5)$$

Dimana:

\sum : simbol *covariance*

\sum : simbol penjumlahan

$(x_i - \bar{x})$: adalah nilai x_i dikurangi dengan rata-ratanya, \bar{x}

$(y_i - \bar{y})$: adalah nilai y_i dikurangi dengan rata-ratanya, \bar{y}

μ_y dan μ_x : adalah rata-rata pada populasi x dan y

2.10 EVD (Eigenvalue Decomposition)

Eigenvalue adalah sebuah bilangan skalar dan *eigenvector* adalah sebuah matriks yang keduanya dapat mendefinisikan matriks A . Dimana matriks A adalah matriks bujur sangkar dengan ukuran $n \times n$. Namun, tidak semua matriks bujur sangkar memiliki *eigenvalue* dan *eigenvector*. *Eigen decomposition* merupakan jenis matriks yang penting dalam statistic karena digunakan untuk mencari maksimum (atau minimum) dari fungsi yang melibatkan matriks tersebut. Berikut ilustrasi yang digambarkan melalui dua buah grafik dibawah ini :

$$\text{Matik } A = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \text{ dengan } x_1 = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \text{ dan } x_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

2.8 Standarisasi Data

Standarisasi merupakan proses untuk mengukur seberapa jauh sebaran data secara keseluruhan yang direpresentasikan ke dalam suatu bilangan. Untuk menghitung nilai standarisasi menggunakan z-score :

$$z = \frac{x - \mu}{\sigma} \dots\dots\dots (2.3)$$

Dimana:

z : nilai standar score,

x : data observasi,

μ : mean per variable,

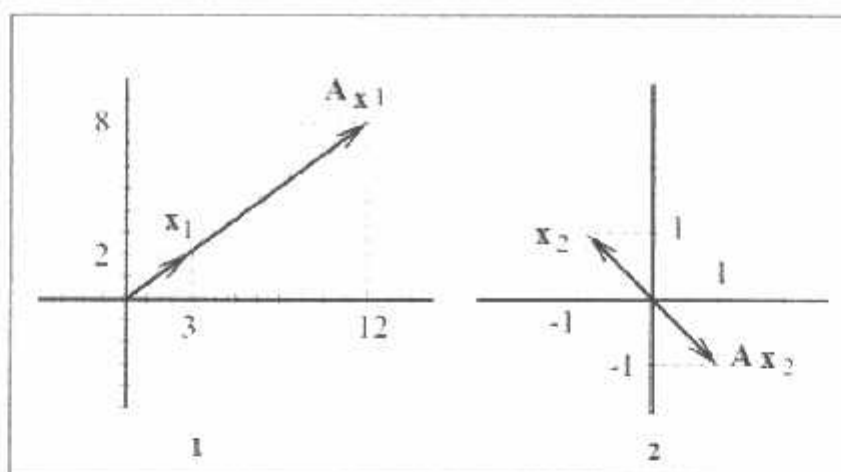
σ : standar deviasi per variabel

Sederhananya proses z-score merupakan proses dimana setiap data observasi pada sebuah variabel dikurangi dengan mean variabel tersebut dan dibagi dengan standar deviasinya. Berarti setiap baris per kolom dikurangi *mean* kolom tersebut, lalu dibagi dengan standar deviasi kolom yang sama. Hasil dari standarisasi membuat data berada pada range yang hampir sama dan dengan unit (deviasi) yang sama pula. Karena data sudah distandarisasi, maka langkah selanjutnya memproses *covariance*.

2.9 Covarience

Covariance atau kovarians merupakan proses yang digunakan untuk mengukur besarnya hubungan antara dua variabel. Jenis hubungan yang dapat terjadi antara dua buah variabel berdasarkan nilai *covariance*-nya adalah:

- a. *Covariance Positive* : yaitu $\text{cov}(x,y) > 0$ dimana variable x dan variable y mengalami perubahan dengan besaran yang sama.
- b. *Covariance Negative* : yaitu $\text{cov}(x,y) < 0$ dimana variable mengalami perubahan pada arah yang berlawanan. Dimana ketika nilai variable x berada pada nilai tinggi maka variable y berada pada nilai rendah dan begitu juga sebaliknya



Pada gambar 1 dengan x_1 dan A

maka

dimana $\lambda = 4$

atau bisa juga dituliskan dengan :

=

=

=

= adalah eigenvector A

$\lambda = 4$ adalah eigenvaluenya

melalui hasil yang keluar dapat disimpulkan bahwa λ adalah perpanjangan atau kelipatan dari x . Pada contoh gambar nomor 2 dengan penyelesaian yang sama namun memiliki arah yang berlawanan. Maka cara untuk menentukan nilai eigenvector dan eigenvalue dengan menggunakan persamaan berikut:

$$A u = \lambda u \dots\dots\dots (2.6)$$

Atau dapat ditulis dengan:

$$(A - \lambda I) u = 0 \dots\dots\dots (2.7)$$

Dimana $u \neq 0$

A = matrik bujur sangkar dengan ukuran $n \times n$

u – eigenvector matrik A (pada contoh diatas dilambangkan dengan x)

λ = skalar (eigenvalue yang berhubungan dengan vectoreigen)

2.11 Euclidean Distance (Jarak Euclidean)

Sebuah objek memiliki beberapa variasi pola yang bisa digunakan sebagai pengenalan objek tersebut. Proses pengenalan yang terjadi pada suatu sistem pengenalan pola pada umumnya adalah dengan membandingkan pola masukan dengan pola yang telah tersimpan pada sistem. Begitu juga pada pengenalan pola iris mata dengan metode PCA juga bekerja dengan prinsip yang sama.

Setelah melakukan proses pelatihan maka akan dilakukan suatu pola latih pada metode ini berupa vektor ciri yang berisi komponen utama dari sejumlah citra latih. Sejumlah vector cirri tersebut tersimpan dalam suatu matrik dan akan dikeluarkan pada proses pengenalan. Untuk proses pengenalan suatu citra uji memiliki dimensi yang sama telah disajikan di sistem. Citra uji tersebut kemudian di ekstraksi ciri dengan cara mengalikan dengan vektor eigen citra latih dan akan menghasilkan vektor ciri berisikan komponen utama yang memiliki dimensi yang sama dengan vektor ciri citra latih.

Setelah mendapat vektor ciri dari citra uji maka proses selanjutnya adalah membandingkan vector ciri dari citra uji dengan vector ciri dari citra latih. Perbandingan tersebut dilakukan dengan cara menghitung jarak euclidean yang secara matematis dapat dituliskan:

$$dist(i, k) = \sqrt{\sum_{j=1}^D (i_j - k_j)^2} \dots\dots\dots (2.8)$$

Dengan $dist(i, k)$ adalah jarak Euclidean antara vektor i dan vektor k ;

i_j adalah komponen ke j dari vektor i ;

k_j adalah komponen ke j dari vektor k ;

D adalah jumlah komponen pada vector i dan vektor k .

Dari hasil perhitungan jarak euclidean tersebut dapat ditentukan suatu citra wajah adalah mirip bila memiliki jarak yang paling terdekat.

2.12 Matlab

Matlab adalah kependekan dari *matrix laboratory*, dimana matlab merupakan bahasa pemrograman tingkat tinggi yang dikhususkan untuk komputasi teknis dan saintifik. matlab mengintegrasikan kemampuan komputasi, visualisasi, dan pemrograman dalam sebuah lingkungan yang tunggal dan mudah digunakan. Matlab memberikan system interaktif yang menggunakan konsep array/matrik sebagai standart variabel elementnya tanpa membutuhkan pendeklarasian array seperti pada bahasa lainnya.

Pada awalnya, matlab merupakan *interface* untuk memberikan kemudahan mengakses data numerik dari proyek LINPACK dan EISPACK yang kemudian dikembangkan hingga menjadi produk komersial dari perusahaan Mathworks, Inc. menggunakan bahasa C++ dan assembler (utamanya untuk fungsi-fungsi dasar matlab). Matlab telah berkembang menjadi sebuah *environment* pemrograman yang canggih yang berisi fungsi-fungsi *built-in* untuk melakukan tugas pengolahan sinyal, aljabar linier, dan kalkulasi matematis lainnya. Matlab juga berisi *toolbox* yang berisi fungsi-fungsi tambahan untuk aplikasi khusus . Matlab bersifat *extensible*, dalam arti bahwa seorang pengguna dapat menulis fungsi baru untuk ditambahkan pada *library* ketika fungsi-fungsi *built-in* yang tersedia tidak dapat melakukan tugas tertentu.^[9]

[9] Away, Gunaidi Ahdia. 2010. *The Shortcut of Matlab Programming*. Penerbit Informatika.

2.12.1 Desktop Tools Matlab

Ketika MATLAB dijalankan pertama kali, Matlab *desktop* tampil, berisi *tools (graphical user interfaces)* untuk mengatur file, variables, dan aplikasi Matlab. Pada Matlab yang dijalankan akan tampil *desktop* dengan ilustrasi sebagai berikut :

1. Windows Utama Matlab

Window ini adalah *window* induk yang melingkupi seluruh lingkungan kerja matlab.

2. Launch pad

Launch pad menyediakan akses yang mudah bagi pemakai dalam memilih opsi dari fungsi, *toolbox*, *demo* dan dokumentasi.

3. Workspace browser

Merupakan navigator bagi pemakai dalam penyediaan informasi berisi kumpulan variabel yang terbentuk pada *workspace* pada saat pemakaian dan disimpan di memory.

4. Current directory browser

Window ini berfungsi sebagai browser direktori aktif, yang hampir sama dengan *window explorer*.

5. Command history

Digunakan untuk menyimpan baris-baris perintah yang telah diketikkan di *command window*. Kita dapat melihat fungsi-fungsi yang digunakan sebelumnya, mengkopi dan menjalankannya kembali dari *command history*.

6. Command window

Digunakan untuk menjalankan seluruh fungsi-fungsi Matlab.

7. Help browser

Untuk mencari dan menampilkan dokumentasi semua produk *MathWorks*.

8. Variable editor

Menampilkan isi *array* dalam format tabel dan dapat digunakan untuk mengedit isi *array*.

9. Editor/Debugger

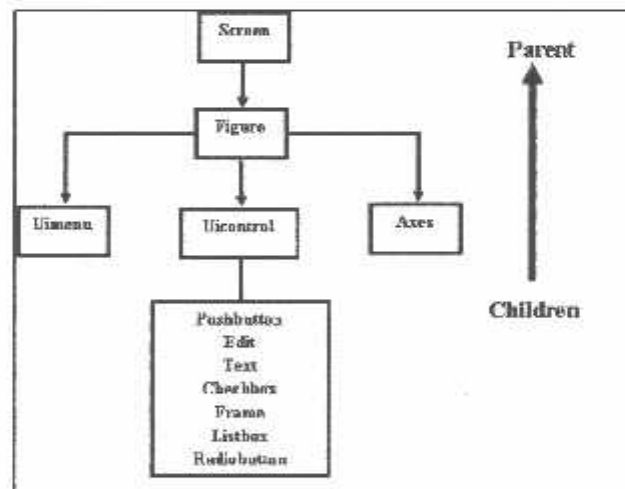
Digunakan untuk membuat, mengedit, dan men-*debug* M-files, yakni program yang dibuat untuk menjalankan fungsi-fungsi MATLAB.

10. Figure

Figure adalah jendela untuk menampilkan gambar grafik seperti perintah *plot* dan sejenisnya. Gambar pada jendela *figure* ini dapat disimpan ke dalam sebuah file yang sewaktu-waktu dapat dipanggil kembali.

2.12.2 Graphic User Interface (GUI) Matlab

Graphic *User Interface* (GUI) adalah media tampilan grafis sebagai pengganti perintah teks untuk user berinteraksi. GUI Matlab bisa menampilkan program yang dibuat jauh terlihat lebih menarik, efektif dan atraktif. Dalam GUI Matlab telah tersedia komponen-komponen standart untuk digunakan. Matlab menggunakan konsep Pemrograman Berbasis Objek (PBO) yaitu dimana setiap komponen diartikan sebagai objek yang dapat diberikan pekerjaan tertentu. Selain itu setiap objek pasti memiliki property untuk saling berinteraksi dengan objek lainnya. Dalam Matlab setiap objek memiliki hirarki objek yang dijabarkan dalam konsep *parent-children*. Berikut adalah diagramnya: ^[10]



Gambar 2.10 Diagram PBO Matlab

[10] Away, Gumudi Abdia. 2010. *The Shortcut of Matlab Programming*. Penerbit Informatika

Maksud dari diagram tersebut adalah setiap objek yang akan digunakan harus diposisikan pada objek *parent*-nya. Pada umumnya objek yang sering digunakan dalam matlab adalah sebagai berikut:

1. Objek *Figure* : merupakan objek tertinggi yang dapat kita gunakan dalam pemrograman window.
 2. Objek *Uicontrol* : merupakan objek yang paling dibutuhkan dalam berinteraksi dengan program. Uicontrol berisi komponen-komponen yang digunakan untuk mendisain form sebagai media interaksi.
 3. Objek *Uimenu* : pada dasarnya Uimenu mirip dengan Uicontrol khususnya pada menu pushbutton.
 4. Objek *Axes* : merupakan bagian yang sangat penting dalam Matlab untuk memvisualisasikan data. Tanpa menggunakan objek axes hasil eksekusi fungsi tidak dapat ditampilkan.
-

BAB III

PERANCANGAN DAN PEMBUATAN SISTEM

3.1 Analisa Perancangan Sistem

Pembahasan dalam bagian ini mengenai analisa sistem dalam melakukan pengolahan citra mata yang digunakan untuk mengidentifikasi iris mata. Data yang digunakan berupa sampel iris mata yang diambil dari database sebuah pusat penelitian bidang biometrik bernama Phoenix. Dimana iris mata tersebut diambil dari sebuah kamera khusus untuk mengambil citra mata.

3.2 Diskripsi Perancangan Sistem

Perancangan aplikasi menggunakan perangkat lunak komputasi numerik dan banyak menggunakan fungsi terapan yang terdapat di *Image Processing Toolbox*. Program ini terdiri dari beberapa bagian utama yaitu *pre-processing*, ekstraksi ciri dan metode *euclidean distance*. Untuk mengolah keseluruhan proses dimanfaatkan fungsi-fungsi dalam software MATLAB R2008b yang dijalankan dengan menggunakan sistem operasi Windows Xp.

Aplikasi dirancang hanya untuk mengidentifikasi pola iris mata dengan menggunakan PCA. *Output* dari aplikasi ini adalah informasi berupa citra mata mana yang cocok ataupun mendekati dengan citra latih yang sudah ada.

3.3 Analisa Perancangan Data

Pembangunan aplikasi harus mempunyai data-data yang akan diproses dan digunakan pada sistem untuk mengidentifikasi pola iris mata. Pada aplikasi yang akan dibangun data-data masukan berupa citra iris mata yang berekstensi adalah .jpeg (Join Photographic Experts Group) berukuran 119x89 piksel. Citra format jpeg dipilih karena salah satu format gambar yang sering digunakan untuk menyimpan citra atau hasil dari pengolahan citra. Citra jpeg mampu memberikan kedalaman warna hingga lebih dari 24 bit atau setara dengan lebih dari 16 juta warna.

3.4 Analisa Kebutuhan Sistem

Di dalam membangun *software* atau program perangkat lunak, dibutuhkan perangkat keras dan perangkat lunak yang mendukung. Dalam pembuatan aplikasi identifikasi pola iris mata dengan menggunakan PCA ini dibutuhkan beberapa perangkat pendukung antara lain :

3.4.1 Perangkat Keras (*Hardware*)

Dibawah ini beberapa kebutuhan perangkat keras (*hardware*) yang dibutuhkan untuk membuat aplikasi :

Seperangkat komputer dengan spesifikasi :

- a. *Processor* Intel(R) Dual-Core
- b. *RAM* 3578 Mb
- c. *Harddisk* 250 Gb

3.4.2 Perangkat Lunak (*Software*)

Sedangkan kebutuhan perangkat lunak (*software*) untuk membangun aplikasi identifikasi pola iris mata dengan menggunakan PCA, menggunakan:

- a. Sistem Operasi Microsoft Windows Xp

Sebagai sistem operasi yang dapat mendukung banyak aplikasi-aplikasi pemrograman yang berbasis windows secara maksimal. Windows Xp relatif lebih cepat dan stabil sehingga mampu memberi kemudahan bagi para penggunaanya untuk membuat aplikasi identifikasi iris mata ini.

- b. Programming MATLAB R2008b

Bahasa pemrograman sebagai media untuk berinteraksi antara manusia dengan komputer dan dibuat agar semakin cepat dan mudah. Matlab dikembangkan sebagai bahasa pemrograman sekaligus alat visualisasi yang menawarkan banyak kemampuan untuk menyelesaikan berbagai kasus yang berhubungan langsung dengan disiplin keilmuan matematika, seperti bidang rekayasa teknik, fisika, statistika, komputasi dan modelling.

Penulis menggunakan bahasa pemrograman ini dikarenakan Matlab R2008b adalah bahasa pemrograman level tinggi yang dikhususkan untuk komputasi teknis, mudah digunakan dan mengintegrasikan kemampuan komputasi, visualisasi dan pemrograman dalam sebuah lingkungan yang tunggal. Matlab memberikan sistem interaktif yang menggunakan konsep array atau matriks sebagai standar variabel elemennya tanpa membutuhkan pendeklarasian array seperti pada bahasa lainnya. GUIDE matlab cocok digunakan untuk aplikasi-aplikasi berorientasi sains untuk penelitian ini dan kemampuan grafisnya cukup handal dan tidak kalah dibandingkan dengan bahasa pemrograman lainnya.

3.5 Analisa Arsitektur Sistem

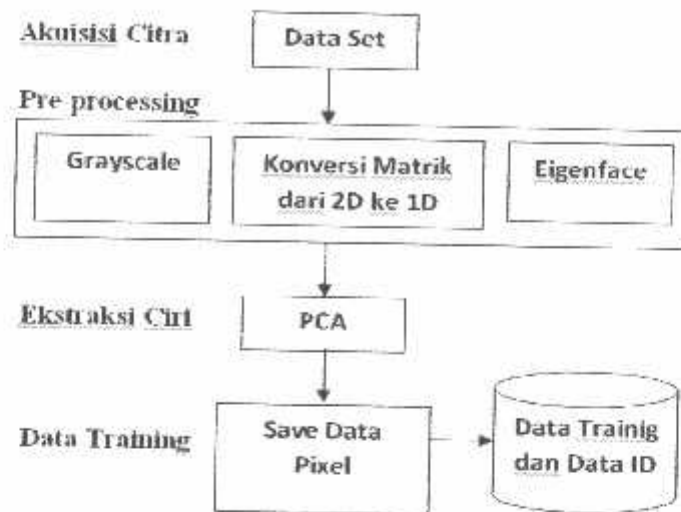
Pada hal ini akan diuraikan tentang perancangan arsitektur sistem. Penjabaran mengenai uraian arsitektur sistem digambarkan dalam 2 bentuk yaitu diagram blok (*block diagram*) dan diagram alir (*flowchart*).

3.6 Diagram Blok

Diagram Blok merupakan gambaran berupa bagian-bagian yang saling berhubungan dalam suatu sistem, dimana bagian-bagian tersebut diwakili oleh blok-blok yang dihubungkan dengan garis-garis yang menunjukkan hubungan dari blok-blok tersebut.

3.6.1 Diagram Blok Proses *Training*

Pada sistem identifikasi iris mata dengan PCA proses diawali dengan proses *training*, bertujuan untuk menyimpan jumlah citra dan nilai *pixel* dari masing-masing citra pada *dataset* sebagai data pembanding. Gambar 3.1 menunjukkan diagram blok proses *training* pada sistem identifikasi iris mata dengan PCA.



Gambar 3.1 Diagram blok proses *training* pada sistem

Diagram blok pada gambar 3.1 dapat dijelaskan sebagai berikut:

1. Akuisisi Citra

Proses akuisisi citra merupakan proses awal yang mendukung dapat jalannya proses-proses lain. Dalam proses ini akan menentukan data yang akan digunakan pada proses *training* sistem identifikasi iris dengan menggunakan 25 citra iris mata yang diambil dari dataset sebuah pusat penelitian bidang biometrik bernama Phoenix.

2. Pra-proses Citra

Proses awal yang bertujuan untuk memperbaiki dan menyederhanakan kualitas citra. Pada pra-proses citra terdapat operasi *grayscale* merupakan proses mengubah/konversi citra RGB (warna) menjadi aras keabuan. Proses konversi matrik untuk membuat matrik 2 dimensi menjadi 1 dimensi dan proses *eigenface*.

3. Ekstraksi Ciri Citra

Pada proses ekstraksi ciri ini digunakan untuk pengeksktraksian atau pengambilan area iris mata. Dalam ekstraksi ciri citra menggunakan PCA yang merupakan kumpulan dari perintah-perintah penghitungan dengan statistik.

4. Data Training

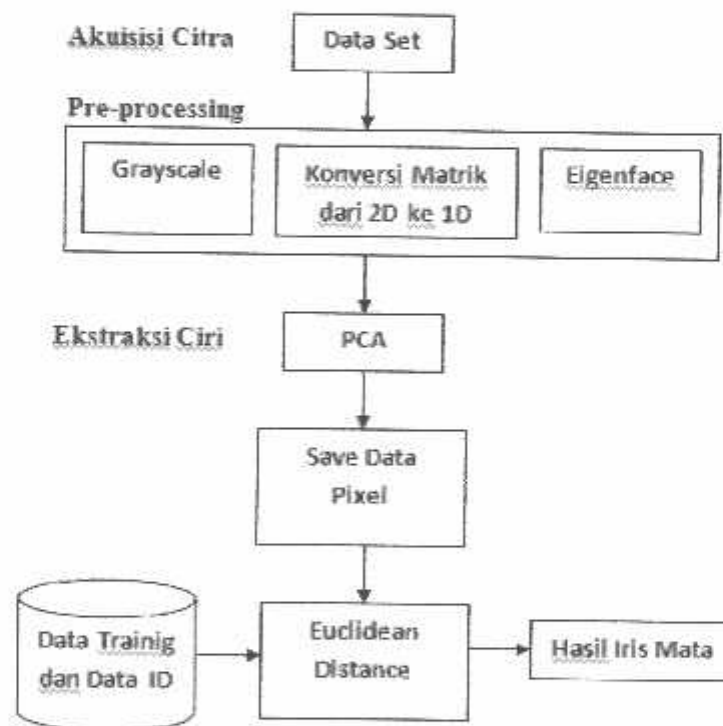
Pada Tahap Data *Training* akan dilakukan penyimpanan nilai *pixel* citra hasil dari ekstraksi ciri citra yang terdiri dari 25 citra iris mata.

5. Save Data Pixel

Pada Tahap Save Data *Pixel* akan dilakukan penyimpanan nilai *pixel* citra hasil dari ekstraksi ciri citra berupa file.mat.

3.6.2 Diagram Blok Proses Pengujian

Setelah proses *training* selesai dilakukan maka didapatkan sebuah *file* hasil *training* yang berisi jumlah citra dan nilai *pixel* citra dari masing-masing citra pada *dataset* yang telah di *training*. Data tersebut akan digunakan dalam proses pengujian sistem sebagai data pembandingan. Gambar 3.2 menunjukkan diagram blok proses pengujian pada sistem identifikasi iris mata menggunakan PCA.



Gambar 3.2 Diagram blok proses pengujian pada sistem

Diagram blok pada gambar 3.2 dapat dijelaskan sebagai berikut:

1. Akuisisi Citra

Menentukan data yang akan digunakan pada proses pengujian sistem yang telah disiapkan dalam dataset.

2. Pra-proses Citra

Proses awal yang bertujuan untuk memperbaiki dan menyederhanakan kualitas citra. Pada pra-proses citra terdapat operasi *grayscale* merupakan proses mengubah/konversi citra RGB (warna) menjadi aras keabuan. Proses konversi matrik untuk membuat matik 2 dimensi menjadi 1 dimensi dan proses *eigenface*.

3. Ekstraksi Ciri Citra

Pada proses ekstraksi ciri ini digunakan untuk pengekstraksian atau pengambilan area iris mata. Dalam ekstraksi ciri citra menggunakan PCA yang merupakan kumpulan dari perintah-perintah penghitungan dengan statistik.

4. Save Data *Pixel*

Pada Tahap Save Data *Pixel* akan dilakukan penyimpanan nilai *pixel* citra hasil dari ekstraksi ciri citra berupa file.mat.

5. Metode Euclidean Distance

Pada tahap ini akan dilakukan proses membandingkan pola masukan dengan pola yang telah tersimpan pada sistem antara data *training* dan data uji.

6. Hasil Iris Mata

merupakan *output* dari sistem yang berupa iris mata dan informasi identitas iris mata.

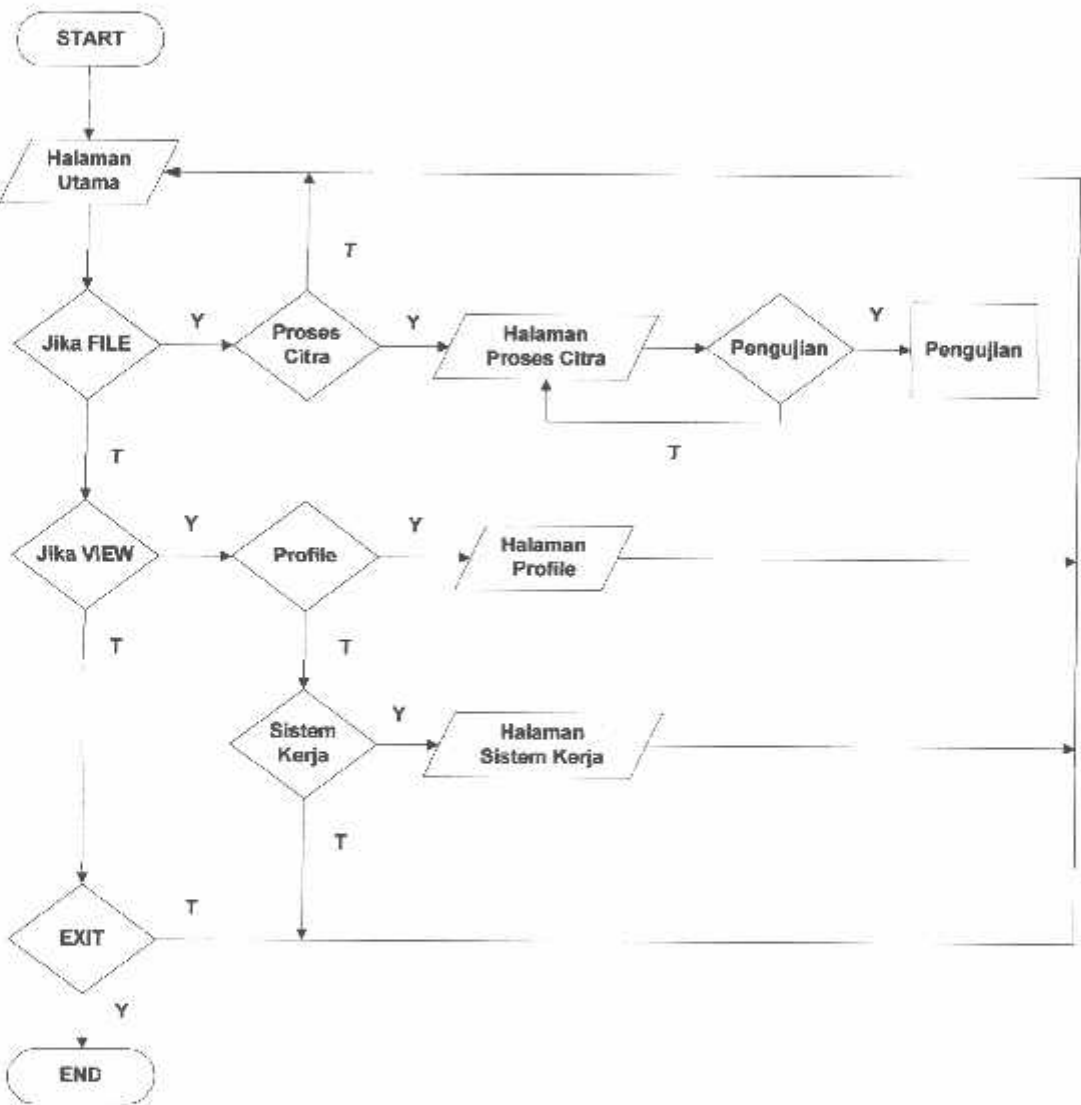
3.7 Diagram Alir (*Flowchart*)

Diagram alir merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah dengan disimbolkan dalam bentuk kotak, beserta urutannya dengan menghubungkan masing-masing langkah tersebut menggunakan tanda panah. Diagram ini dapat memberikan

solusi selangkah demi selangkah untuk menyelesaikan proses pada sistem aplikasi identifikasi iris mata dengan PCA.

3.7.1 Flowchart Sistem

Flowchart sistem merupakan gambaran diagram alir sistem, mulai dari awal (halaman utama) hingga akhir (exit) puntos mempermudah pemahaman pengguna terhadap jalanya proses aplikasi identifikasi iris mata dengan PCA ini. Gambar 3.3 menunjukkan flowchart pada sistem identifikasi iris mata menggunakan PCA.



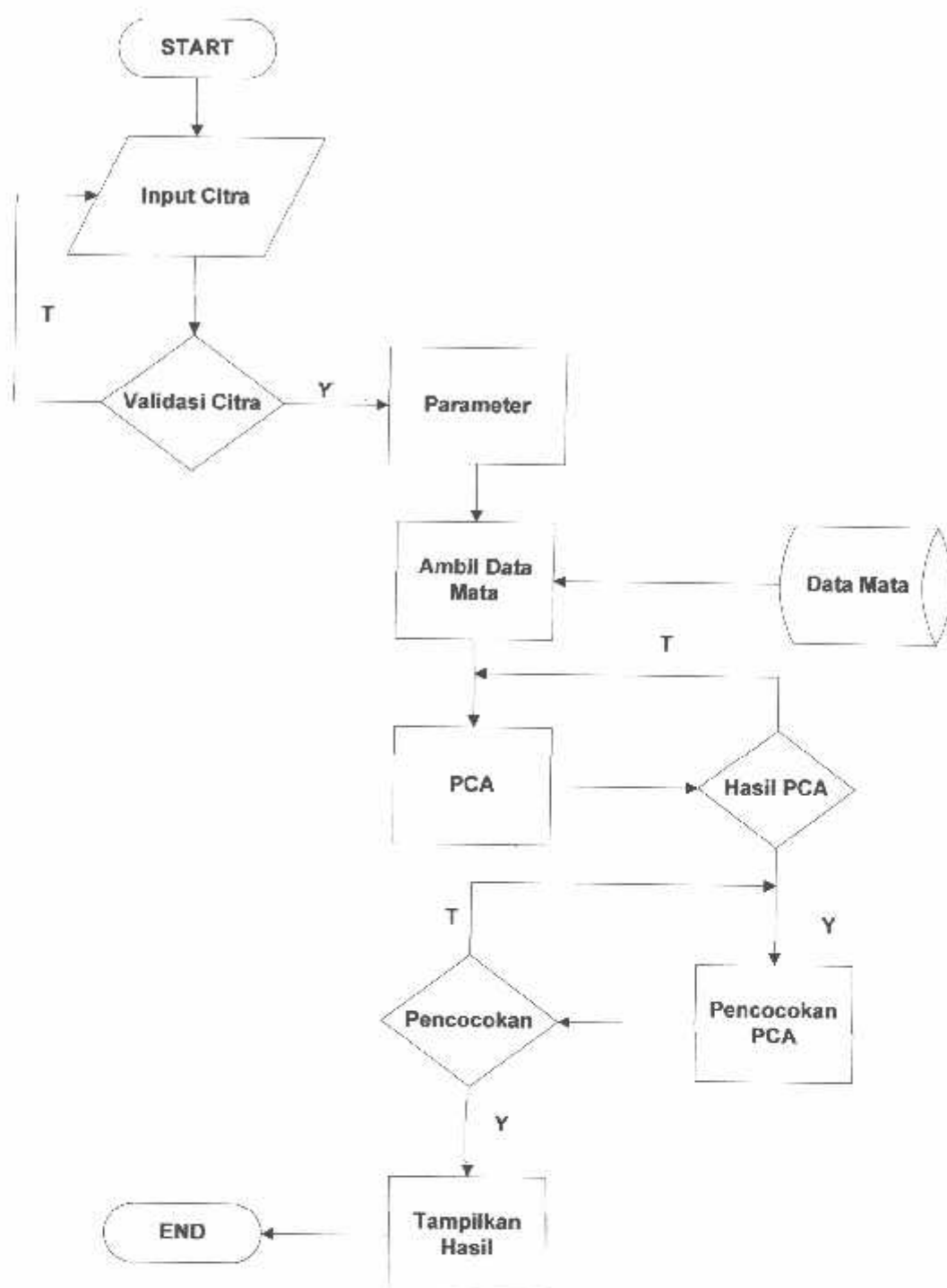
Gambar 3.3 Flowchart sistem

Flowchart sistem pada gambar 3.3 dapat dijelaskan sebagai berikut:

1. Dalam sistem penerapan PCA untuk identifikasi iris mata tampilan akan dimulai dengan menampilkan halaman utama aplikasi.
2. Pada halaman utama aplikasi terdapat 2 (dua) menu utama, yaitu: menu *file* dan menu *view*.
3. Pada menu *file* terdapat 2 (dua) sub menu, yaitu menu proses citra dan menu keluar. Jika *user* memilih menu proses citra, maka sistem akan menampilkan halaman untuk memproses citra mulai dari awal hingga proses mengidentifikasi kecocokan iris mata. Jika *user* memilih menu keluar maka akan ada *notification* 'ya' atau 'tidak', jika 'ya' maka *user* akan keluar, jika 'tidak' *user* akan tetap pada halaman utama.
4. Pada sub menu proses citra bila *user* memilih sub menu pengujian maka akan terhubung dengan proses pengujian. Proses pengujian digunakan untuk menguji jalannya sistem pengidentifikasian iris mata dengan PCA ini. Bila ingin keluar maka *user* bisa memilih membatalkan dan kembali ke menu proses citra.
5. Pada menu *view* didalam menampilkan 2 (dua) sub menu, yaitu menu profil dan menu sistem kerja. Jika *user* memilih menu profil, maka sistem akan menampilkan halaman profil penulis. Jika *user* memilih menu keterangan sistem kerja, maka sistem akan menampilkan halaman yang berisi tentang keterangan sistem jalannya aplikasi.

3.7.2 Flowchart Aplikasi

Flowchart aplikasi merupakan gambaran diagram alir aplikasi pengolahan citra, dimulai dari awal (akuisisi citra) proses citra hingga akhir hasil pengolahan citra (klasifikasi dan deteksi). Gambar 3.4 menunjukkan *flowchart* aplikasi pada sistem penerapan identifikasi iris mata menggunakan PCA.



Gambar 3.4 Flowchart aplikasi pengolahan citra pada sistem

Flowchart aplikasi pengolahan citra pada sistem identifikasi iris mata menggunakan PCA dapat dijelaskan dengan algoritma aplikasi sebagai berikut.

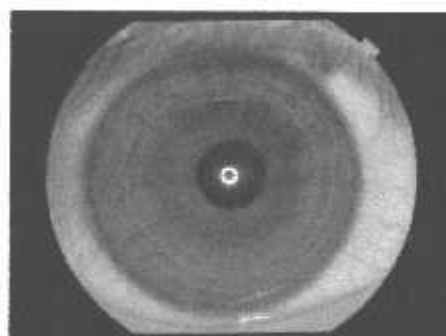
3.8 Algoritma Aplikasi

Dalam proses pembuatan aplikasi diperlukan suatu perencanaan algoritma, dalam sistem Pengidentifikasian iris mata dengan PCA akan dijelaskan perancangan algoritma sebagai berikut:

1. Akuisis Citra merupakan proses pengambilan citra yang dimulai dari:
 - a. Pengambilan citra uji pada dataset.
 - b. Pengambilan/*load* data citra dari aplikasi untuk dilakukan pemrosesan.
 2. Pra-proses Citra adalah proses awal yang bertujuan untuk memperbaiki dan menyederhanakan kualitas citra. Pada pra-proses citra terdapat operasi *grayscale* merupakan proses mengubah/konversi citra RGB (warna) menjadi aras keabuan. Proses konversi matrik untuk membuat matik 2 dimensi menjadi 1 dimensi dan proses *eigenface*.
 3. Ekstraksi Ciri Citra
 Pada proses ekstraksi ciri ini digunakan untuk pengekstraksian atau pengambilan area iris mata. Dalam ekstraksi ciri citra menggunakan PCA yang merupakan kumpulan dari perintah-perintah penghitungan dengan statistik.
 4. *Save Data Pixel*
 Pada Tahap *Save Data Pixel* akan dilakukan penyimpanan nilai *pixel* citra hasil dari ekstraksi ciri citra berupa file.mat.
 5. Metode Euclidean Distance
 Pada tahap ini akan dilakukan proses membandingkan pola masukan dengan pola yang telah tersimpan pada sistem antara data *training* dan data uji.
 6. Hasil Iris Mata
 merupakan *output* dari sistem yang berupa iris mata dan informasi identitas iris mata.
-

3.9 Tahap Pengambilan Citra Mata

Data yang akan digunakan adalah data citra yang didapatkan dari database iris mata oleh sebuah pusat penelitian bidang biometrik bernama Phoenix-<http://phoenix.inf.upol.cz>. Alasan pemilihan menggunakan dataset yang sudah ada dikarenakan terdapat beberapa aspek yang harus diperhatikan dalam pemotretan citra mata, diantaranya adalah warna, pencahayaan, posisi pengambilan maupun skala atau ukuran citra. Maka dari itu untuk mempermudah proses pengolahan citra selanjutnya digunakan dataset yang sudah ada. Adapun contoh citra irismata terdapat pada Gambar 3.5.



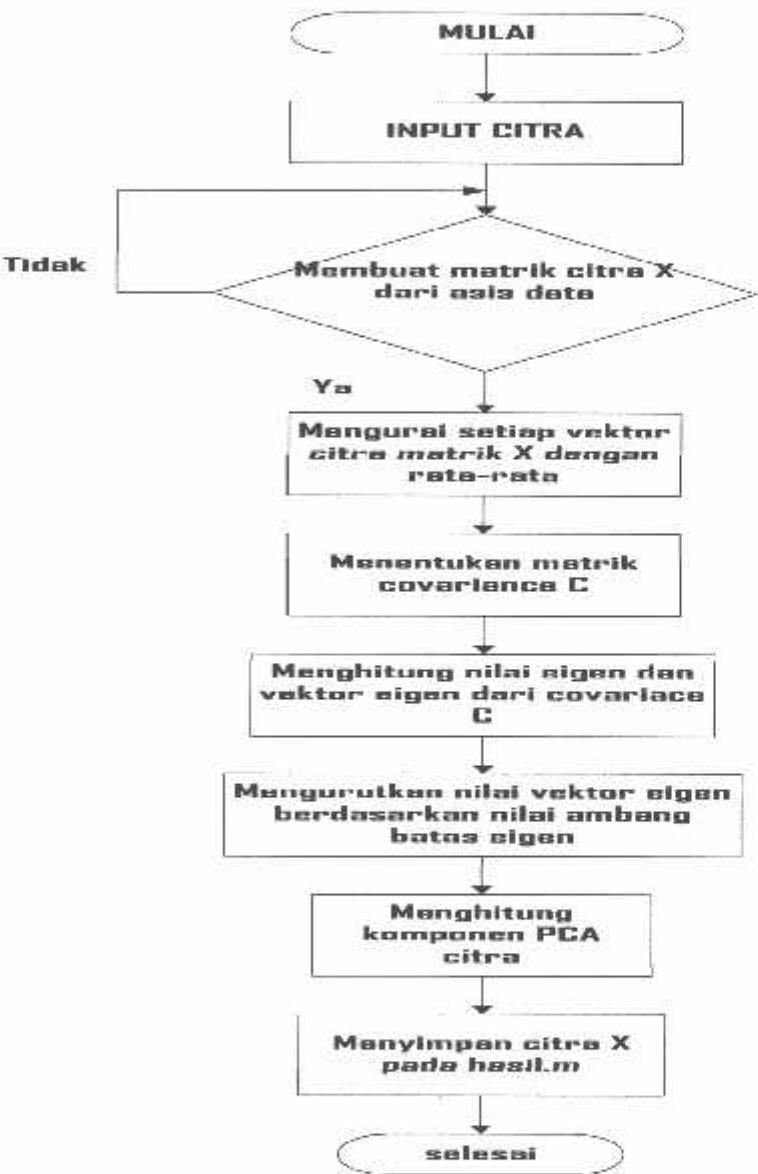
Gambar 3.5 Contoh Citra Iris Mata

3.10 Tahap Pra-proses Citra

Tahap pra-proses citra yaitu salah satu proses awal dalam pengolahan citra untuk memperbaiki kualitas citra dan menyederhanakan citra sehingga dapat mempermudah dalam proses selanjutnya. Dalam pra-proses citra terdapat operasi *grayscale* merupakan proses mengubah/konversi citra RGB (warna) menjadi aras keabuan. Proses tersebut membuat citra mudah diproses pada proses selanjutnya, karena citra ini terbentuk dari sebuah matrik yang nilainya berkisar antara 0 hingga 255. Citra *grayscale* memiliki warna dari hitam, keabuan dan putih. Pada pemrograman matlab fungsi untuk merubah citra RGB menjadi citra *Grayscale* adalah $x = \text{rgb2gray}(y)$. Proses selanjutnya adalah mengkonversi matrik pada citra. Pada proses konversi matrik membuat matrik pada citra dari 2 dimensi menjadi 1 dimensi. Selanjutnya hasil dari matrik yang sudah dikonversi tersebut dapat digunakan untuk mencari nilai *eigenface* pada citra untuk bisa diproses pada tahap-tahap lainnya.

3.11 Tahap Ekstraksi Ciri Citra

Pada proses ekstraksi ciri ini digunakan untuk pengekstraksian atau pengambilan area iris mata. Dalam ekstraksi ciri citra menggunakan PCA yang merupakan kumpulan dari perintah-perintah penghitungan dengan statistik. Tahapan dalam pengekstraksian pola iris mata dimulai dari menyiapkan sejumlah citra yang akan diproses. Citra yang akan diproses kemudin akan melalui beberapa lagkah. Gamar 3.6 merupakan langkah pengekstraksi citra.



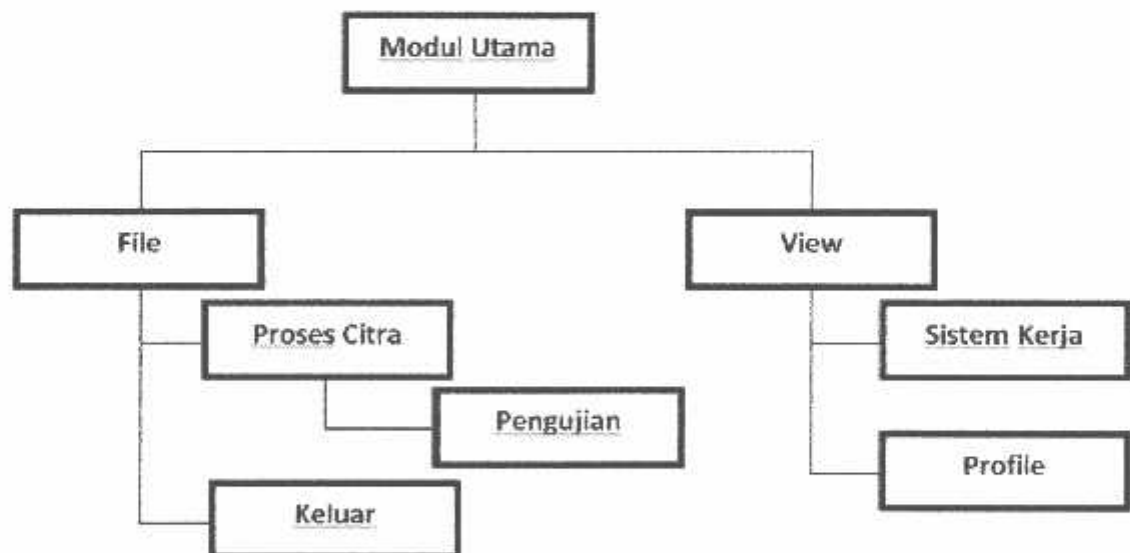
Gambar 3.6 Flowchart PCA

Proses *flowchart* gambar 3.6 melalui beberapa tahapan yaitu:

1. Memasukkan Citra
2. Pegalihan citra mejadi nilai matrik yag disimpan pada basis data
3. Mengurangi setiap vector iris mata (nilai x- nilai rata-rata) utuk meghasilkan matrik y
4. Hasil proses diatas dapat digunakan untuk mentukan nilai covariace
5. Bila nilai covariace sudah didapatkan maka dapat mentukan nilai eigen dan vector eigen
6. Megurutkan nilai vector eigen berdasarkan nilai ambang batas eigen
7. Maka degan bcrakhirnya proses diatas nilai PCA dihasilkan
8. Hasil nilai sebuah PCA akan disimpan pada .m
9. Proses selesai

3.12 Perancangan Menu Sistem

Pada perancangan menu sistem akan menjelaskan tentang menu-menu dan submenu-submenu yang terdapat pada sistem penerapan PCA dengan menggunakan iris mata. Gambar 3.7 Merupakan perancangan menu pada sistem.



Gambar 3.7 Perancangan menu pada sistem

Rancangan menu pada gambar 3.7 dapat dijelaskan sebagai berikut :

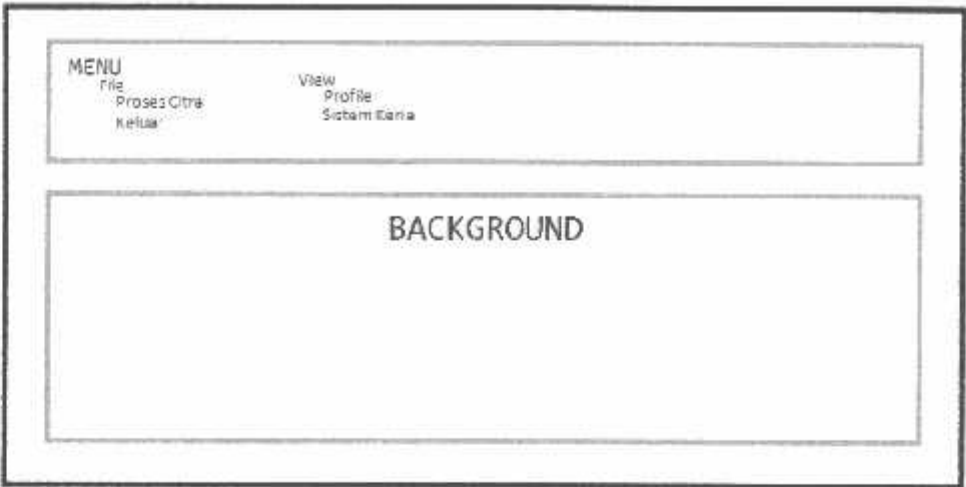
1. Menu *File* berfungsi untuk menampilkan submenu proses citra dan submenu keluar.
2. Submenu proses citra berfungsi untuk menampilkan halaman aplikasi untuk memproses citra.
3. Didalam sub menu proses citra terdapat menu pengujian yang berfungsi sebagai penguji kecocokan citra iris mata dengan ID.
4. Submenu keluar berfungsi untuk menampilkan *notification* keluar dari sistem.
5. Menu *View* berfungsi untuk menampilkan submenu profil dan sub menu sistem kerja
6. Submenu profil berfungsi menampilkan halaman profil penulis.
7. Submenu sistem kerja berfungsi sebagai penjelasan jalannya sistem aplikasi.

3.13 Perancangan Desain Aplikasi

Pada perancangan desain *interface* pada aplikasi akan dibuat menggunakan *tools* GUI pada Matlab R2008b yang diterapkan pada aplikasi Penerapan PCA untuk identifikasi iris mata. Dengan GUI sudah disediakan *tools-tools* untuk mendesain yang cara membuatnya hanya dengan melakukan *drag and drop tools* sesuai dengan kebutuhan kita. Perancangan desain aplikasi dari sistem ini terdiri dari desain tampilan halaman utama dapat dilihat pada Gambar 3.8, desain tampilan proses citra dapat dilihat pada Gambar 3.9, desain tampilan pengujian citra dapat dilihat pada Gambar 3.10, desain tampilan profile dapat dilihat pada Gambar 3.11 dan desain tampilan sistem kerja dapat dilihat pada Gambar 3.12.

3.13.1 Perancangan Tampilan Halaman Utama

Desain dibawah ini merupakan tampilan utama dari aplikasi pengenala iris mata. Halaman depan pada aplikasi ini adalah bagian yang paling penting, karena diletakkannya kontrol dalam pengaksesan menu lain. Gambar 3.8 ini adalah rancangan tampilan halaman depan.



Gambar 3.8 Rancangan halaman utama

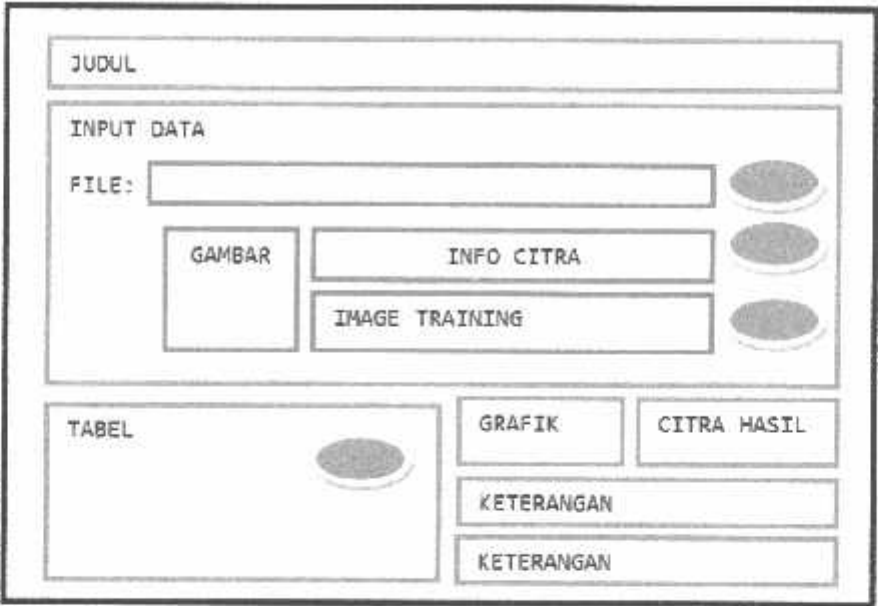
Pada tampilan utama terdapat komponen *menu editor* dan *axes*. *Menu editor* berfungsi untuk membuat menu, sedangkan *axes* berfungsi untuk menampilkan gambar yang digunakan sebagai *background*. Komponen pada tampilan utama dapat dilihat pada Tabel 3.1.

Tabel 3.1 Komponen tampilan halaman utama

Name	Properties	Value
Axes	Tag	Axes2
Menu editor	Label	File : proses citra dan keluar View : profil, sistem kerja

3.13.2 Perancangan Tampilan Proses Citra

Tampilan dibawah ini merupakan desain proses citra. Pada menu ini sebuah citra diproses utuk mengenali pola iris mata dan dilakukan pecocokan citra pada basis data yang sudah dsimpan. Selain itu juga akan dikeluarkan grafik pengenalan pola mata. Halama ini merupakan inti dari jalanya aplikasi yang dibuat.Gambar 3.9 ini adalah rancangan proses citra.



Gambar 3.9 Rancangan proses citra

Pada proses citra terdapat 6 komponen *panel*, 3 *axes*, 1 *table* dan 4 *button*. *Menu panel* berfungsi untuk mengelompokkan setiap bagian-bagiab komponen, sedangkan *axes* berfungsi untuk menampilkan gambar dan grafik. Komponen *button* berfungsi sebagai eksekusi perintah dan komponen *table* berfungsi sebagai penampil data citra. Komponen pada halaman dapat dilihat pada Tabel 3.2.

Tabel 3.2 Komponen tampilan proses citra

Name	Properties	Value
Panel 1	Tag	Uipnl3
	Tittle	Judul
Panel 2	Tag	pnl1
	Tittle	Input Data
Panel 3	Tag	uipanel4
	Tittle	priview
Panel 4	Tag	Uipanel5
	Tittle	informasi

Panel 5	Tag	pnl3
	Title	Grafik jarak PCA
Panel 6	Tag	pnl2
	Title	PCA
Panel 6	Tag	pnl4
	Title	Data Iris Hasil
Axes 1	Tag	axImage
Axes 2	Tag	axGrapPCA
Axes 3	Tag	axHasil
Table	Tag	tablePCA
Button 1	String	...
	Tag	btnOPen
Button 2	String	Pengujian
	Tag	btnUji
Button 3	String	...
	Tag	pushbutton5
Button 4	String	Proses PCA
	Tag	btnProses
Static Text 1	String	IRIS-PCA
	Tag	text1
Static Text 2	String	Algoritma iris mata PCA
	Tag	text3
Static Text 3	String	Nama File :
	Tag	text12
Static Text 4	String	W:
	Tag	text15
Static Text 5	String	H:
	Tag	text17

Tabel 3.3 Komponen tampilan pengujian

Name	Properties	Value
Panel 1	Tag	pnlUji
	Title	
Panel 2	Tag	uipanel16
	Title	Hasil:
Static Text 1	String	File:
	Tag	text18
Static Text 2	String	ID:
	Tag	Text19
Static Text 3	Tag	txtlHasilUji
Static Text 4	Tag	edUjiFile
Edit Text 1	Tag	edID
Edit Text 4	Tag	txtl1
Button 1	String	...
	Tag	btnUjiOpen
Button 2	String	Proses
	Tag	btnProsesUji
Button 3	String	Batal
	Tag	pushbutton9

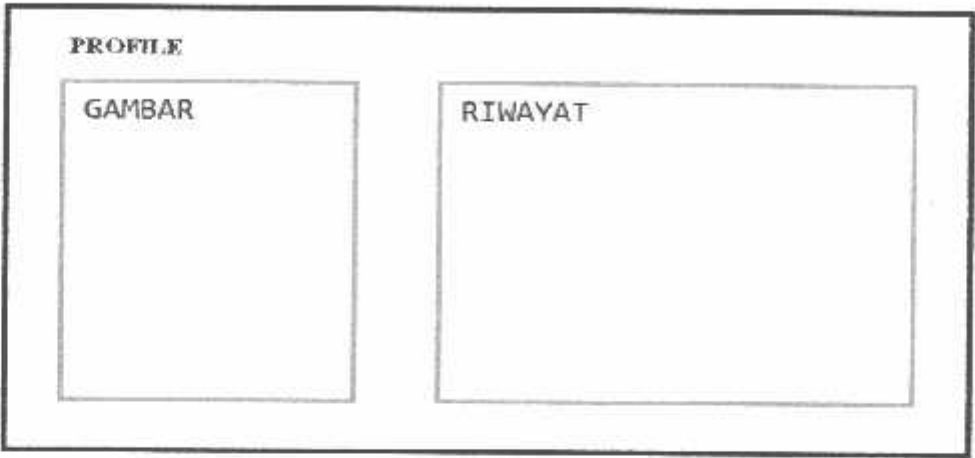
3.13.4 Perancangan Tampilan Profile

Tampilan profil menampilkan informasi mengenai identitas penulis. Gambar 3.11 menunjukkan rancangan tampilan profil pada sistem.

Pada rancangan tampilan profil terdapat sebuah komponen berupa *axes* yang berfungsi untuk menampilkan gambar yang digunakan sebagai *background*. Komponen pada tampilan profil dapat dilihat pada Tabel 3.3.

Tabel 3.4 Komponen tampilan profil

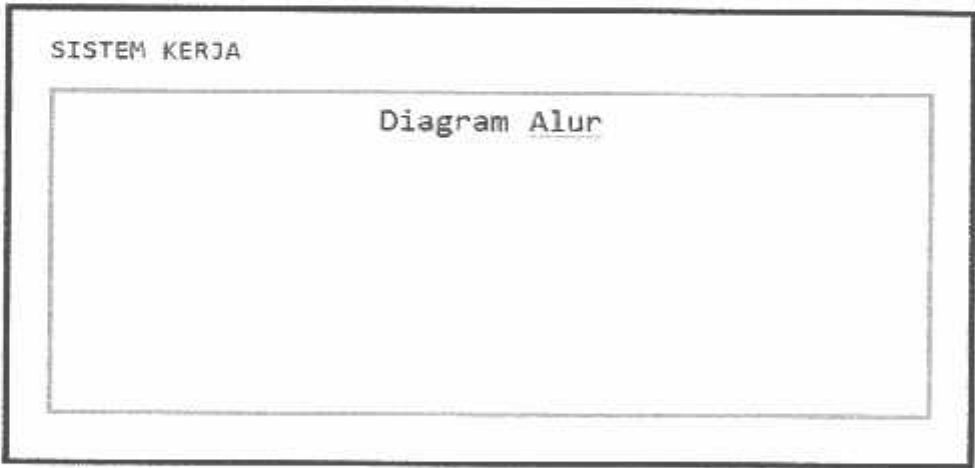
Name	Properties	Value
Axes	Tag	Axes5



Gambar 3.11 Rancangan profile

3.13.5 Perancangan Tampilan Sistem Kerja

Tampilan sistem Kerja menampilkan informasi penjelasan secara singkat aplikasi yang dibuat. Gambar 3.12 menunjukkan rancangan tampilan profil pada sistem.



Gambar 3.12 Rancangan sistem kerja

Pada rancangan tampilan Sistem Kerja terdapat sebuah komponen berupa *axes* yang berfungsi untuk menampilkan gambar yang digunakan sebagai *background*. Komponen pada tampilan profil dapat dilihat pada Tabel 3.5.

Tabel 3.5 Komponen tampilan sistem kerja

Name	Properties	Value
Axes	Tag	Axes4

BAB IV

IMPLEMENTASI DAN PENGUJIAN

Dalam bab ini akan dibahas tentang implementasi dan pengujian berdasarkan perencanaan dari sistem yang dibuat. Pengujian ini dilaksanakan untuk mengetahui kinerja dari suatu sistem yang dibuat apakah sudah sesuai dengan perencanaan yang diinginkan.

4.1 Implementasi

Pada tahapan ini akan membahas tentang bagaimana membuat penerapan dari perancangan sistem yang telah dibuat sebelumnya. Perancangan dan penerapan sistem yang dilakukan meliputi 2 aspek yaitu user interface atau tampilan antar muka sebagai penghubung dengan user dan implementasi program.

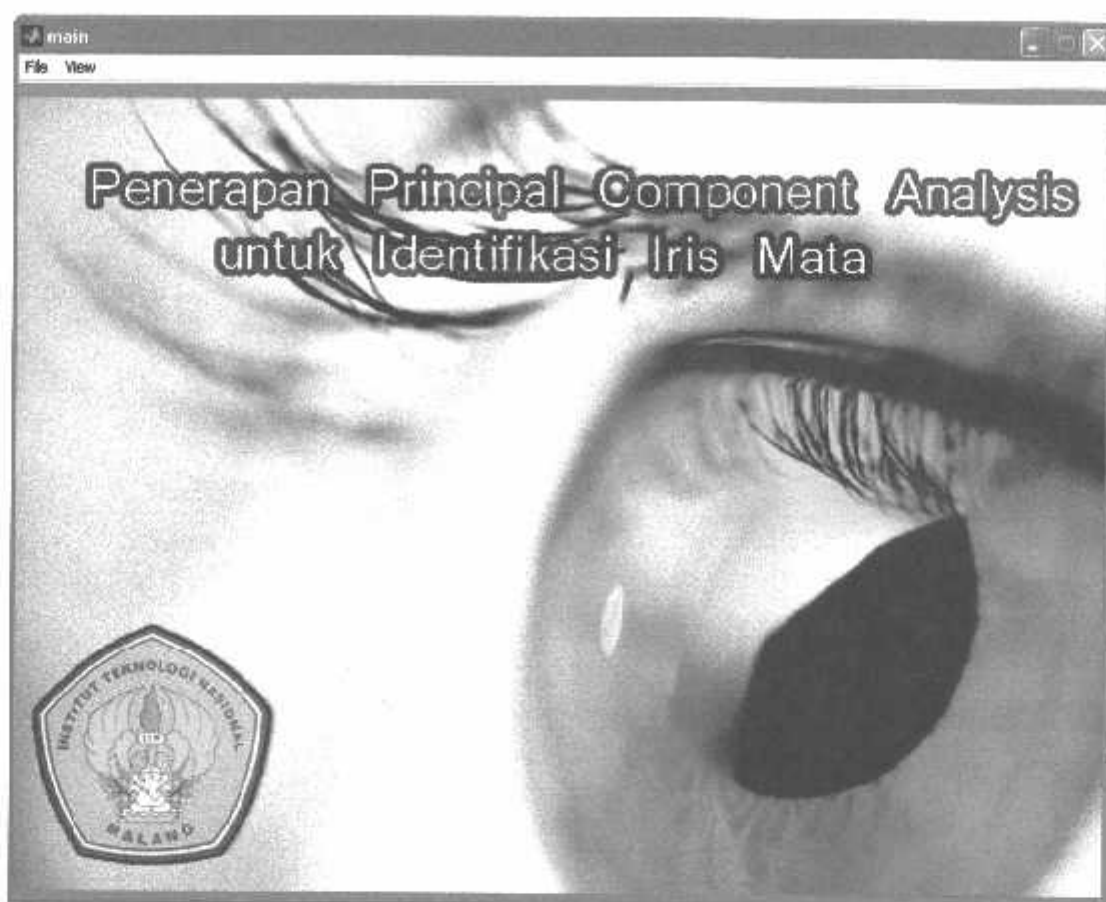
4.2 Implementasi *User Interface*

Implementasi *user interface* pada sistem penerapan principal component analysis untuk identifikasi iris mata mempunyai 5 (lima) tampilan *interface* yaitu:

1. Tampilan halaman utama
2. Tampilan halaman Proses Citra
3. Tampilan halaman Pengujian
4. Tampilan halaman Profile
5. Tampilan halaman Sistem Kerja

4.2.1 Implementasi Halaman Utama

Tampilan utama merupakan halaman yang tampil pada saat sistem pertama kali dijalankan. Halaman ini adalah bagian yang paling penting, karena merupakan kontrol dalam pengaksesan menu lain. Tampilan utama pada sistem ini dapat dilihat pada gambar 4.1



Gambar 4.1 Implementasi Halaman Utama

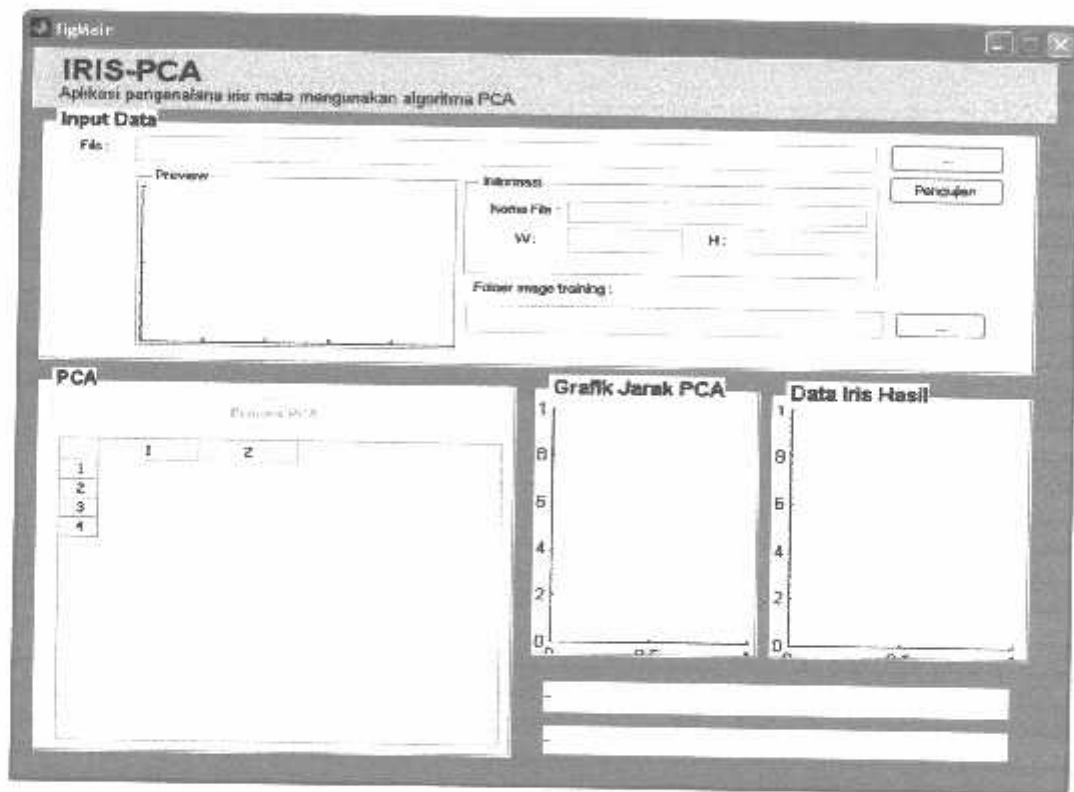
Pada halaman utama sistem PCA untuk identifikasi iris mata terdapat beberapa menu utama yang terdiri dari beberapa submenu, antara lain :

1. *File*, berfungsi untuk menampilkan beberapa halaman dari submenu yaitu, submenu proses citra dan submenu keluar.
2. *View*, berfungsi untuk menampilkan beberapa halaman dari submenu yaitu, submenu profil dan submenu sistem kerja.

4.2.2 Implementasi Proses Citra

Tampilan submenu proses citra ini berfungsi untuk memproses citra mulai dari pengambil citra awal dari komputer sampai memberikan informasi citra tersebut sama atau mirip sesuai dengan citra yang ada di dalam basis-data. Dalam halaman tersebut

terdapat beberapa bagian-bagian yaitu bagian pertama untuk mengambil citra awal yang diberi nama 'Input Data', didalam bagian 'Input Data' terdapat sub-sub bagian yaitu bagian yang menampilkan info citra yang diberi nama 'Informasi' dan bagian untuk memilih tempat tersimpnnya image training yang diberi nama ' Folder Image Training', bagian kedua untuk proses melakukan PCA dan menampilkan nilai eucliden antar citra diberi nama 'PCA', bagian ketiga untuk melihat grafik jarak PCA diberi nama 'Grafik Jarak PCA', dan bagian output berupa hasil iris mata yang keluar diberi nama 'Data Iris Hasil'. Tampilan proses citra dapat dilihat pada gambar 4.2.



Gambar 4.2 Implementasi Proses Citra

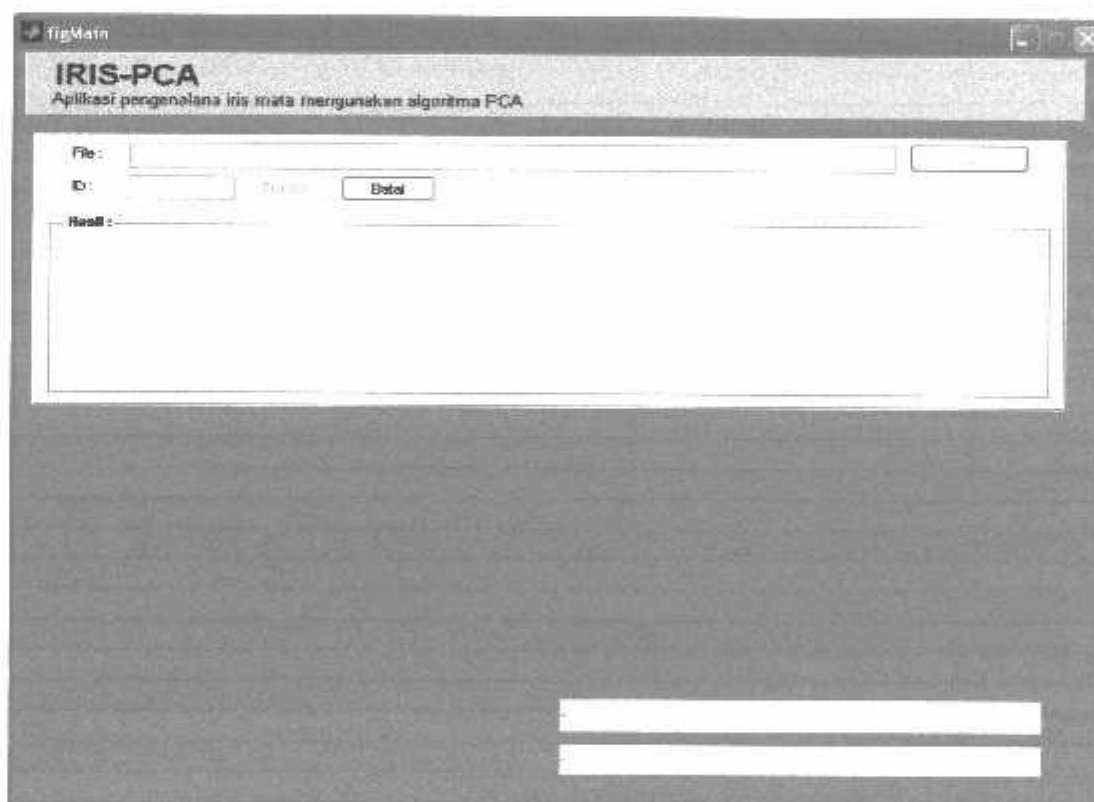
Pada tampilan proses citra terdapat 4 button yang memiliki fungsi sebagai berikut:

1. Button "Ambil Citra" berfungsi untuk membuka/mengambil citra asal dari komputer dan menampilkan informasi dari citra asal tersebut.
2. Button "Pengujian" berfungsi untuk menghubungkan dengan Form Pengujian.

3. Button “Pilih” berfungsi untuk menentukan letak folder citra yang sudah ditraining.
4. Button “Proses Citra” berfungsi untuk melakukan proses PCA dan proses jarak euclidean.

4.2.3 Implementasi Pengujian

Tampilan Pengujian berfungsi untuk menguji suatu citra yang akan dilakukan pencocokan citra dengan identitas iris mata yang telah disimpan. Iris mata yang digunakan merupakan iris yang telah dilakukan pelatihan. Dalam tampilan ‘Pengujian’ terdapat beberapa bagian yaitu bagian ‘File’ untuk memilih citra yang akan dilakukan pengujian, bagian ‘ID’ untuk memilih id mana yang akan dilakukan pencocokan dengan citra yang telah dipilih dan bagian ‘Hasil’ akan menampilkan output berupa identitas pemilik iris mata. Tampilan pengujian dapat dilihat pada gambar 4.3.



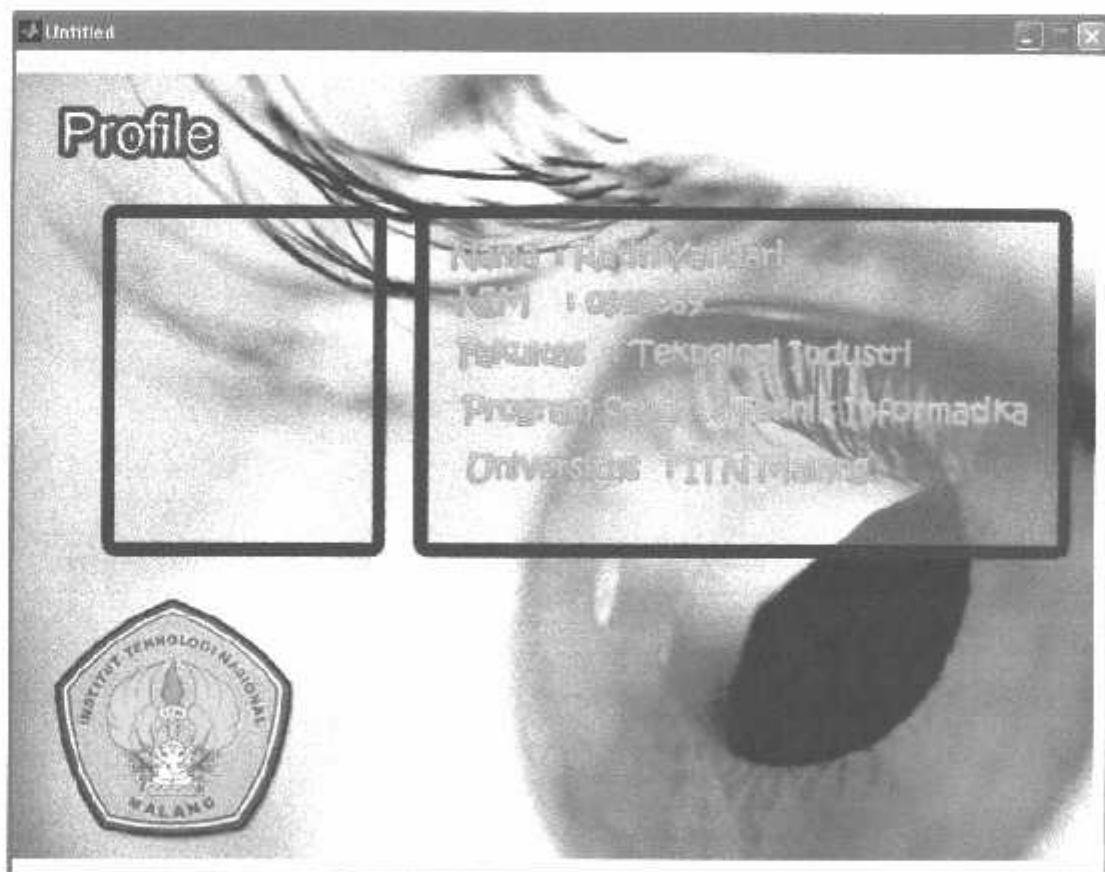
Gambar 4.3 Implementasi Pengujian

Pada tampilan pengujian terdapat 3 button yang masing-masing button berfungsi sebagai berikut:

1. Button “Open” berfungsi untuk membuka/mengambil citra asal dari komputer.
2. Button “Proses” berfungsi untuk menampilkan hasil kecocokan antara citra dengan ID yang dipilih.
3. Button “Batal” berfungsi untuk kembali ke ‘Proses Citra’.

4.2.4 Implementasi Profile

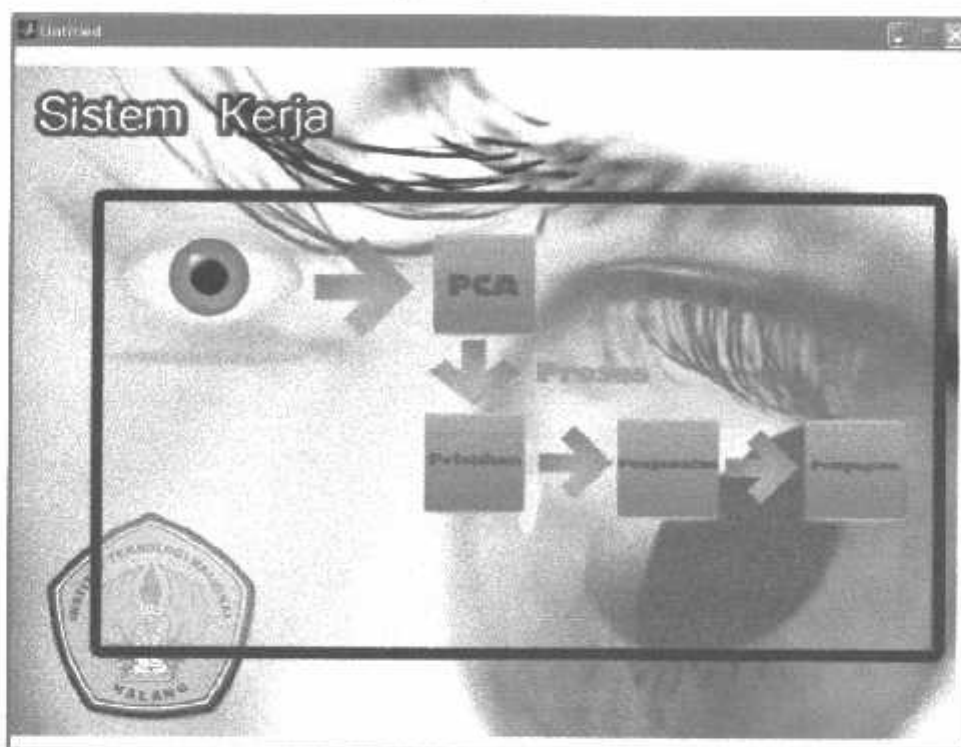
Tampilan *profile* terdapat informasi mengenai identitas dari penulis yaitu foto penulis, nama, nim, fakultas, program studi, dan universitas. Tampilan gambar 4.4 menunjukkan tampilan profil pada sistem.



Gambar 4.4 Implementasi Profile

4.2.5 Implementasi Sistem Kerja

Tampilan Sistem Kerja berisi tentang info alur kerja sistem penerapan PCA untuk identifikasi iris mata yang dibuat. Dalam gambar tersebut dijelaskan bagaimana tahapan awal mulai mengambil citra iris mata hingga proses pengenalnya. Tampilan sistem kerja dapat dilihat pada gambar 4.5.



Gambar 4.5 Implementasi sistem kerja

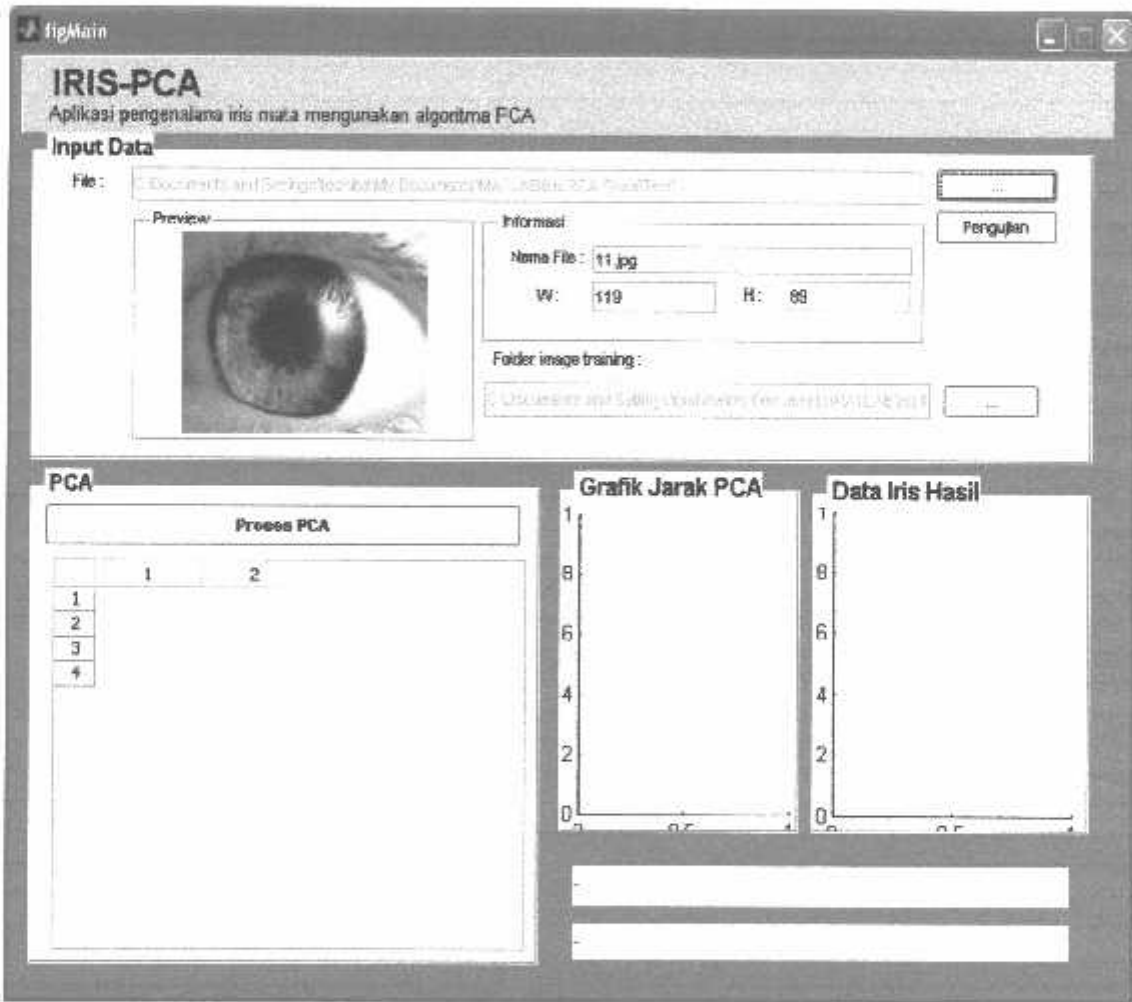
4.3 Implementasi Program

Implementasi program sistem penerapan PCA untuk identifikasi iris mata terdapat pada tampilan proses citra pada gambar 4.2. Dalam proses diawali dengan mengambil citra yang telah disimpan dalam komputer. Selanjutnya menentukan letak folder training yang sudah disimpan. Proses terakhir melakukan ekstraksi ciri dengan PCA sehingga informasi citra yang digunakan untuk identifikasi dapat diketahui. Proses PCA tersebut berkaitan dengan penentuan nilai jarak Euclidean antara citra latih dan citra uji yang berfungsi sebagai pembandingan antara vector ciri citra uji dan citra latih sehingga citra tersebut dapat dikenali sesuai dengan pola yang dimiliki.

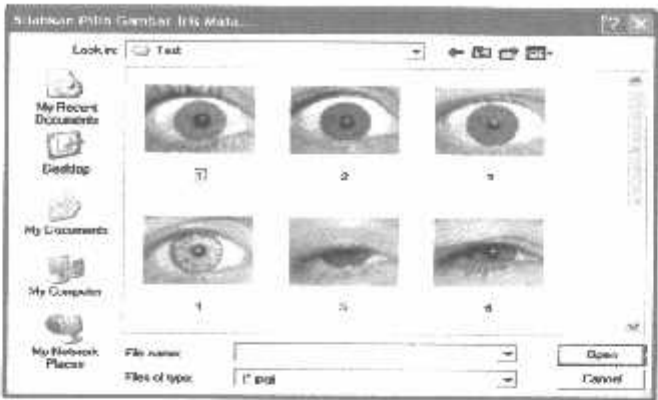
4.3.1 Implementasi Proses Ambil Citra

Proses mengambil citra adalah proses awal dalam sistem penerapan PCA untuk identifikasi iris mata. Dengan meng-klik tombol “Ambil Citra” pada tampilan proses citra gambar 4.2, maka citra yang ada dalam komputer dapat kita ambil dan akan diproses pada tahap berikutnya pada sistem.

“Tombol “Pilih” akan secara otomatis terhubung dengan folder training yang sudah kita tentukan sebelumnya. Gambar 4.6 proses pengambilan citra dari komputer. Gambar 4.7 adalah citra yang tersimpan dalam komputer yang akan diambil untuk pengujian sistem.



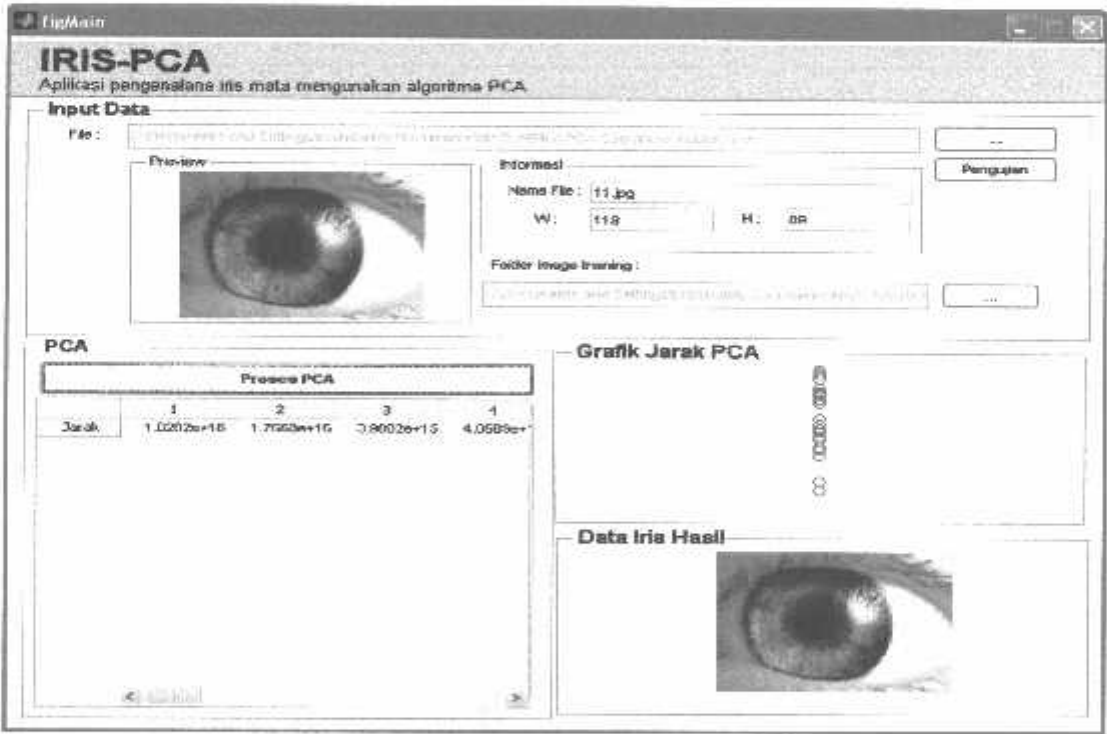
Gambar 4.6 Proses pengambilan citra pada ‘Proses Citra’



Gambar 4.7 Citra yang terdapat dalam komputer

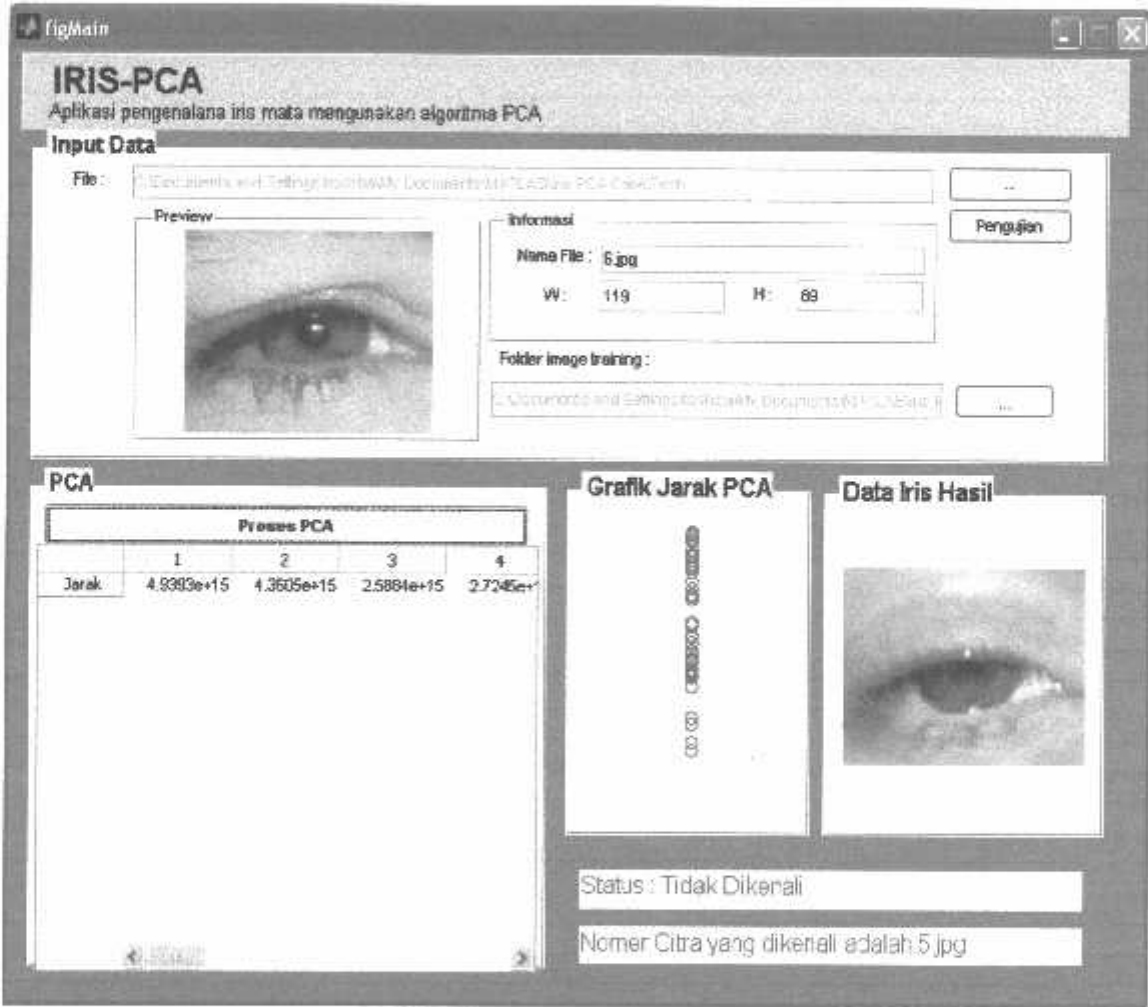
4.3.2 Implementasi Proses Citra

Untuk memproses citra awal yang telah diambil dari media komputer user dapat meng-klik tombol “Proses PCA” pada tampilan proses citra gambar 4.2 untuk memproses citra ke tahap berikutnya yaitu ekstraksi ciri dan menghasilkan nilai jarak euclidean. Pada gambar 4.8 merupakan implementasi dari proses citra.



Gambar 4.8 Hasil citra yang dikenali pada tampilan ‘Proses Citra’

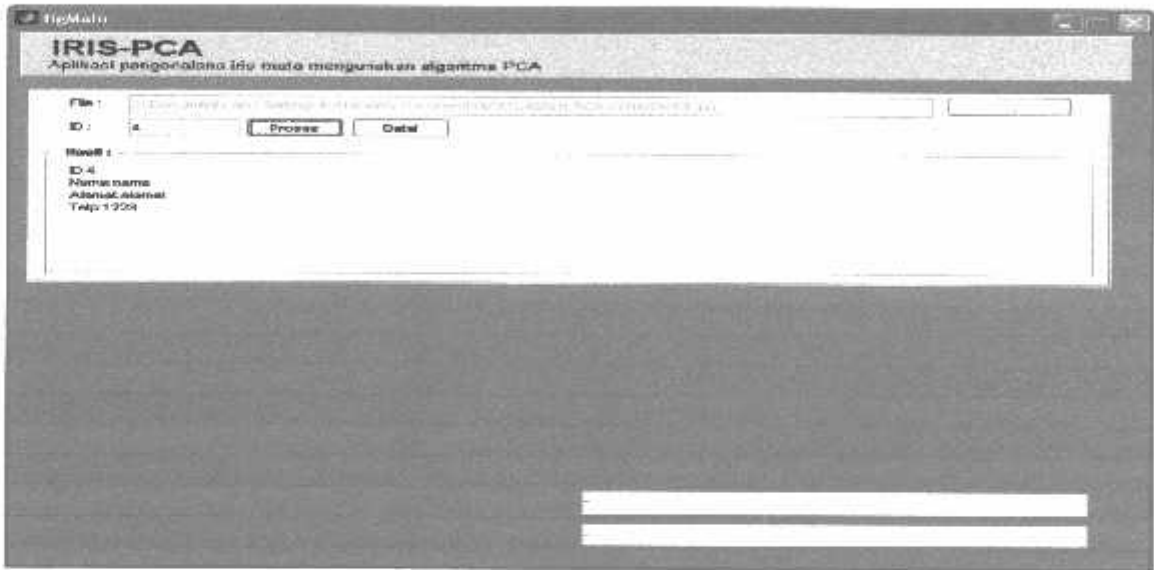
Tahap dalam ‘Proses Citra’ terdiri dari beberapa pra-proses citra yang meliputi proses mengubah citra warna menjadi grayscale untuk mempermudah pemrosesan citra. Proses selanjutnya melakukan standarisasi data dan covariance dari tahap tersebut. Hasil dari nilai covariance dapat menentukan proses selanjutnya yaitu menentukan nilai eigenface. Nilai eigenface inilah yang menjadi vector ciri sebuah citra. Dari hasil proses tersebut juga akan menampilkan jarak euclidean berupa tampilan table dan grafik. Maka pada output akan keluar data iris hasil yang sesuai dengan citra yang diuji. Bila nilai yang diproses tidak sesuai maka sistem akan mencari citra uji yang memiliki nilai terdekat. Adapun Gambar 4.9 adalah contoh proses citra yang tidak dikenali.



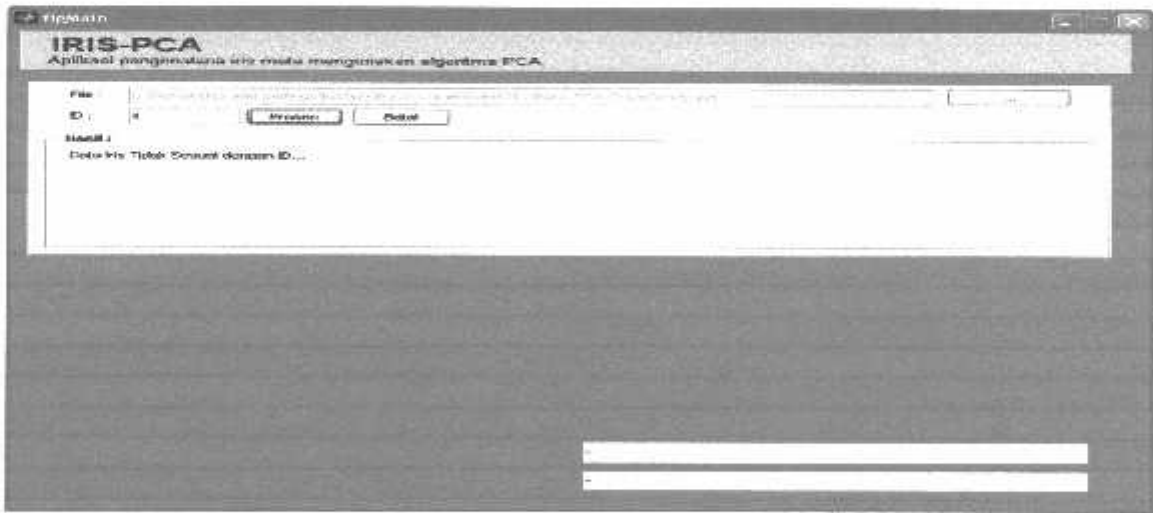
Gambar 4.9 Hasil citra yang tidak dikenali pada tampilan ‘Proses Citra’

4.3.3 Implementasi Proses Pengujian

Dari gambar 4.8 implementasi ‘Proses Citra’ terdapat sebuah tombol untuk menghubungkan dengan *form* pengujian yaitu tombol “Pengujian”. Didalam proses pengujian dilakukan pencocokan iris mata dengan identitas yang telah disimpan didalam komputer. Pada gambar 4.10 merupakan implementasi dari pengujian yang dapat dikenali. Adapun pada gambar 4.11 merupakan pengujian yang tidak bisa dikenali.



Gambar 4.10 Hasil citra yang dikenali pada tampilan ‘Pengujian’



Gambar 4.11 Hasil citra yang tidak dikenali pada tampilan ‘Pengujian’

4.4 Pengujian Aplikasi

Setelah implementasi *user interface* dan implementasi program tahap selanjutnya adalah tahap pengujian sistem. Tahap pengujian dimaksudkan untuk mengetahui keakuratan hasil pengenalan yang dihasilkan oleh sistem pengenalan iris mata dengan PCA.

Pada tahap pengujian akan memasukkan data citra iris mata yang terbagi menjadi 2 (dua) yaitu citra uji dan citra hasil *training* yang telah disimpan. Data-data tersebut diambil dari dataset dari CASIA versi 4.0 dan melalui dataset mata pada situs <http://phoenix.inf.upol.cz>. Data citra iris mata tersebut berupa citra RGB yang berjumlah 50 citra yang digunakan sebagai citra *training* dan citra uji. Dalam pengujian sistem diterapkan berbagai variasi keadaan pengujian sehingga membuat sistem lebih terlatih.

4.4.1 Pengaruh Jumlah Citra Latih

Dalam pengujian ini, sistem dilatih menggunakan 3macam pengujian. Dimana dalam pengujian pertama hanya menggunakan 10 citra latih dan 10 citra uji, pengujian kedua menggunakan 20 citra uji dan 20 citra latih dan pada pegujian ketiga menggunakan 25 citra latih dengan 25 citra uji. Hasil dari pengujian pertama dapat dilihat pada Tabel 4.1, Tabel 4.2 untuk hasil pengujian kedua dan Tabel 4.3 hasil pengujian ketiga.

Tabel 4.1 Tabel Hasil Pengujian 1

No.	Citra Uji	Hasil Pengenalan		Kesimpulan
		Jarak	Citra Terdekat	
1	Citra 1.jpeg	7,32 ^{^14}	Citra 1.jpeg	Sesuai
2	Citra 2.jpeg	1,27 ^{^15}	Citra 2.jpeg	Sesuai
3	Citra 3.jpeg	9,35 ^{^14}	Citra 3.jpeg	Sesuai
4	Citra 4.jpeg	1,10 ^{^14}	Citra 4.jpeg	Sesuai

8	Citra 11.jpeg	3,68 ¹⁴	Citra 11.jpeg	Sesuai
9	Citra 12.jpeg	3,42 ¹⁴	Citra 12.jpeg	Sesuai
10	Citra 13.jpeg	1,70 ¹⁵	Citra 13.jpeg	Sesuai
11	Citra 14.jpeg	1,15 ¹⁴	Citra 14.jpeg	Sesuai
12	Citra 15.jpeg	4,08 ¹⁴	Citra 15.jpeg	Sesuai
13	Citra 16.jpeg	3,12 ¹⁴	Citra 16.jpeg	Sesuai
14	Citra 17.jpeg	2,14 ¹⁵	Citra 24.jpeg	Tidak Sesuai
15	Citra 18.jpeg	1,54 ¹⁴	Citra 18.jpeg	Sesuai
16	Citra 19.jpeg	6,19 ¹⁴	Citra 19.jpeg	Sesuai
17	Citra 20.jpeg	3,84 ¹⁴	Citra 20.jpeg	Sesuai
18	Citra 21.jpeg	4,28 ¹⁴	Citra 21.jpeg	Sesuai
19	Citra 22.jpeg	9,97 ¹⁴	Citra 6.jpeg	Tidak Sesuai
20	Citra 23.jpeg	2,21 ¹⁴	Citra 23.jpeg	Sesuai

Hasil pengujian kedua diatas dapat diperoleh prosentase kecocokan sebagai berikut:

Prosentase Kecocokan = $\frac{\sum \text{citra uji yang cocok}}{\sum \text{citra uji}} \times 100\% = \frac{16}{20} \times 100\% = 80\%$

Tabel 4.3 Tabel Hasil Pengujian 3

No.	Citra Uji	Hasil Pengenalan		Kesimpulan
		Jarak	Citra Terdekat	
1	Citra 1.jpeg	1,56 ¹⁴	Citra 1.jpeg	Sesuai
2	Citra 2.jpeg	1,05 ¹⁵	Citra 2.jpeg	Sesuai

3	Citra 3.jpeg	1,30 ¹⁵	Citra 3.jpeg	Sesuai
4	Citra 4.jpeg	1,17 ¹⁵	Citra 4.jpeg	Sesuai
5	Citra 5.jpeg	2,09 ¹⁵	Citra 14.jpeg	Tidak Sesuai
6	Citra 6.jpeg	2,11 ¹⁵	Citra 6.jpeg	Sesuai
7	Citra 7.jpeg	2,27 ¹⁴	Citra 5.jpeg	Sesuai
8	Citra 8.jpeg	2,44 ¹⁴	Citra 8.jpeg	Sesuai
9	Citra 9.jpeg	3,17 ¹⁴	Citra 9.jpeg	Sesuai
10	Citra 10.jpeg	1,10 ¹⁵	Citra 10.jpeg	Sesuai
11	Citra 11.jpeg	8,34 ¹⁴	Citra 11.jpeg	Sesuai
12	Citra 12.jpeg	7,13 ¹⁴	Citra 12.jpeg	Sesuai
13	Citra 13.jpeg	1,10 ¹⁵	Citra 13.jpeg	Sesuai
14	Citra 14.jpeg	4,99 ¹⁴	Citra 14.jpeg	Sesuai
15	Citra 15.jpeg	6,06 ¹⁴	Citra 15.jpeg	Sesuai
16	Citra 16.jpeg	5,73 ¹⁴	Citra 16.jpeg	Sesuai
17	Citra 17.jpeg	2,91 ¹⁵	Citra 17.jpeg	Sesuai
18	Citra 18.jpeg	2,57 ¹⁴	Citra 18.jpeg	Sesuai
19	Citra 19.jpeg	6,82 ¹⁴	Citra 19.jpeg	Sesuai
20	Citra 20.jpeg	5,03 ¹⁴	Citra 20.jpeg	Sesuai
21	Citra 21.jpeg	4,83 ¹⁴	Citra 21.jpeg	Sesuai
22	Citra 22.jpeg	1,06 ¹⁵	Citra 22.jpeg	Sesuai
23	Citra 23.jpeg	3,72 ¹⁵	Citra 23.jpeg	Sesuai

24	Citra 24.jpeg	1,00^15	Citra 24.jpeg	Sesuai
25	Citra 25.jpeg	1,09^15	Citra 1.jpeg	Tidak Sesuai

Sedangkan hasil pengujian ketiga diperoleh kecocokan sebagai berikut:

Prosentase Kecocokan = $\frac{\sum \text{citra uji yang cocok}}{\sum \text{citra uji}} \times 100\% = \frac{23}{25} \times 100\% = 92\%$

Dari ketiga pengujian diatas dapat disimpulkan bahwa pada pengujian pertama memiliki prosentase kesalahan terbesar dikarenakan belum adanya variasi pada citra latih yang disajikan ke sistem, sehingga pada saat suatu iris mata dengan ekspresi yang berbeda dari citra latihnya disajikan ke sistem untuk dikenali, citra uji tersebut cenderung mengacu pada citra latih iris mata lain yng bentuk dan ekspresi irisnya lebih mirip. Sedangkan untuk pengujian kedua prosentase kesalahan lebih sedikit dibandingkan yang pertama dikarenakan variasinya yang sudah beragam.

Tingkat keberhasilan pengenalan pada pengujian ketiga cukup tinggi yaitu 92%. Pada pengujian ketiga ini sistem mampu mengurangi kesalahan pengenalan yang terjadi pada pengujian dengan jumlah citra latih yang lebih sedikit. Hal ini kemungkinan akibat variasi ekspresi iris mata dari citra uji, yang membutuhkan sejumlah citra latih dengan eksprei yang sesuai untuk dapat dikenali dengan baik oleh sistem pengenalan iris mata. Oleh karenanya dapat disimpulkan bahwa jumlah citra latih berbanding lurus dengan kinerja dari sistem pengenalan iris mata, dalam hal kemampuan melakukan pengenalan terhadap variasi ekspresi iris mata yang disajikan ke sistem. Artinya, semakin banyak variasi iris mata yang dilathkan, kemampuan sistem pengenalan iris juga akan semakin tinggi.

4.4.2 Pengujian Akurasi Identitas

Pada pengujian ini memiliki sebuah studi kasus. Setiap iris mata yang disimpan telah diberikan identitasnya masing-masing. Identitas tersebut berupa ID yang berbeda-beda. Bila ID yang dimasukkan pada pengujian merupakan milik iris

mata tertentu maka pada keterangan akan muncul identitas lengkap iris mata tersebut, jika bukan maka identitas tidak akan tampil karena tidak adanya kecocokan ID dengan iris mata yang sedang diuji. Berikut ini pengujian menggunakan 25 citra uji. Hasil dengan pengujian sebuah studi kasus ini dapat dilihat pada Tabel 4.4 dibawah ini.

Tabel 4.4 Tabel Hasil Pengujian Akurasi Identitas

No.	Citra Uji	Hasil Pengenalan		Kesimpulan
		ID	ID Teridentifikasi	
1	Citra 1.jpeg	1	1	Sesuai
2	Citra 2.jpeg	2	2	Sesuai
3	Citra 3.jpeg	3	3	Sesuai
4	Citra 8.jpeg	8	8	Sesuai
5	Citra 9.jpeg	9	9	Sesuai
6	Citra 10.jpeg	10	15	Tidak Sesuai
7	Citra 11.jpeg	11	11	Sesuai
8	Citra 12.jpeg	12	12	Sesuai
9	Citra 13.jpeg	13	15	Tidak Sesuai
10	Citra 14.jpeg	14	14	Sesuai
11	Citra 15.jpeg	15	15	Sesuai
12	Citra 16.jpeg	16	16	Sesuai
13	Citra 17.jpeg	17	13	Tidak Sesuai

14	Citra 18.jpeg	18	18	Sesuai
15	Citra 19.jpeg	19	15	Tidak Sesuai
16	Citra 20.jpeg	20	20	Sesuai
17	Citra 21.jpeg	21	21	Sesuai
18	Citra 22.jpeg	22	3	Tidak Sesuai
19	Citra 23.jpeg	23	23	Sesuai
20	Citra 24.jpeg	24	24	Sesuai

Hasil pengujian diperoleh kecocokan sebagai berikut:

Prosentase Kecocokan = $\frac{\sum \text{citra uji yang cocok}}{\sum \text{citra uji}} \times 100\% = \frac{15}{20} \times 100\% = 75\%$

Jadi berdasarkan perhitungan dengan studi kasus diatas tingkat keakurasian sistem untuk identifikasi iris mata tersebut sebesar 75%.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian dapat diambil beberapa kesimpulan sebagai berikut:

- 1) Metode PCA dapat digunakan untuk mengekstrak tekstur iris mata yang kemudian dapat dikenali dengan perhitungan jarak euclidean.
- 2) Pada pengujian berdasarkan jumlah citra latih menggunakan 10 citra latih terdapat kesalahan pengenalan sebanyak 3 kesalahan dengan prosentase keberhasilan 70%.
- 3) Pada pengujian berdasarkan jumlah citra latih menggunakan 20 citra latih terdapat kesalahan pengenalan sebanyak 4 kesalahan dengan prosentase keberhasilan 80%.
- 4) Pada pengujian berdasarkan jumlah citra latih menggunakan 25 citra latih terdapat kesalahan pengenalan sebanyak 2 kesalahan dengan prosentase keberhasilan 92%.
- 5) Jumlah citra latih yang digunakan pada metode PCA berbanding lurus dengan kinerja sistem pengenalan iris mata , artinya semakin banyak citra latih yang digunakan untuk pelatihan hasil pengenalan semakin bagus dan begitu juga sebaliknya semakin sedikit citra latih yang digunakan untuk pelatihan hasil pengenalan semakin buruk.
- 6) Keberhasilan sistem dipengaruhi oleh akuisisi citra dan proses pengolahan awal citra. Akuisisi citra dan proses pengolahan awal yang buruk menyebabkan sistem kurang mengenali dengan tepat.
- 7) Prosentase pengenalan iris mata terbaik ialah 92 % yaitu pada pengenalan dengan menggunakan 25 citra latih.

5.2 Saran

Adapun saran-saran yang dapat disampaikan yang berkaitan dengan aplikasi ini :

- 1) Sistem identifikasi iris mata ini dapat dikembangkan sekaigus dengan perangkat keras dalam memproses citra iris mata dapat didapatkan dari kamera secara langsung (realtime).
 - 2) Citra Masukan sebaiknya pada aplikasi ini dapat membaca ekstensi file selain .jpg (misalnya: .png, .bmp, dan lain-lain).
-

DAFTAR PUSTAKA

- Ahmad, U. 2005. *Pengolahan Citra Digital dan Teknik Pemrogramannya*. Graha Ilmu.
- Away, Gunaidi Abdia. 2010. *The Shortcut of Matlab Programming*. Bandung: Penerbit Informatika.
- Ch.Wijaya, M., Prijono, A. 2007. *Pengolahan Citra Digital Menggunakan MATLAB*. Informatika Bandung.
- Erwin, H., Unggul, Kurniawan, N. & Ilham, K. 2002. *Keamanan Iris*, (Online), (nurichsan.blog.unsoed.ac.id/files/.../Keamanan-Iris.ppt), diakses tanggal 23 Mei 2013).
- Fahmi, S. T, M. Sc. 2007. *Perancangan Algoritma Pengolahan Citra Mata Menjadi Citra Polar Iris Sebagai Bentuk Antara Sistem Biometrik*, (Online), (library.usu.ac.id/download/ft/07001529.pdf), diakses tanggal 18 Mei 2013).
- Majid, N. 2009. *Keystroke Biometric*. Ottawa: School of Information Technology and Engineering (STIE).
- Muntasa, A. & Purnomo, M.H. 2005. *Konsep Pengolahan Citra dan Ekstraksi Ciri*. Yogyakarta : Penerbit Graha Ilmu.
- Murni, A. 1992. *Pengantar Pengolahan Citra*". Jakarta: PT. Elex Media Komputindo.
- Prapita, N.L. 2013. *Langkah Umum Principal Component Analysis (PCA)*, (Online), (<http://putuprapita.com/2013/04/19/langkah-umum-principal-component-analysis/>), diakses tanggal 19 Juni 2013).
- Rizal, I.R., Hidayatno, A., & Dika, A.K. 2011. *Identifikasi Objek Berdasarkan Bentuk dan Ukuran*. (Online), (www.undip.ac.id), diakses tanggal 10 April 2013).
- Smith, L.I. 2002. *A Tutorial on Principal Component Analysis*. (Online), (www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf), diakses tanggal 10 April 2013).
-

- Sukamaïji, A. 2007. *Iris dan Vein Biometric*. Semarang: Universitas Diponegoro.
- Sutoyo, T., Mulyanto, E., Suhartono, V., Nurhayati, D.O., & Wijanarko. 2009. *Teori Pengenalan Citra Digital*. Semarang: Andi Yogyakarta dan UDINUS.
- Wibowo, B.B. 2005. *Pengenalan Wajah Menggunakan Analisis Komponen Utama*. Semarang: Universitas Diponegoro.
- , PHOENIX, *Dataset Iris Patterns*, <http://phoenix.inf.upol.cz>, April 2013.
- , *Image Processing Toolbox, for user's with MATLAB*, user's guide, The mathwork Inc.
-

LAMPIRAN



PROGRAM STUDI TEKNIK INFORMATIKA S-1
INSTITUT TEKNOLOGI NASIONAL MALANG
Jln. Raya Karanglo, Km2
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : Ratih Yanuari
NIM : 0918069
Masa Bimbingan : 11 Mei 2013 s.d 11 November 2013
Judul Skripsi : **Penerapan *Principal Component Analysis* Untuk Identifikasi Iris Mata**

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1.	18 Juni 2013	Revisi Halaman Hasil / Kesimpulan program	
2.	3 Juli 2013	Revisi Halaman awal / utama program	
3.	11 Juli 2013	Revisi Bab IV	
4.	12 Juli 2013	ACC Bab IV Revisi Bab IV	
5.	13 Juli 2013	ACC Bab IV Revisi Makalah Seminar Hasil	
6.	19 Juli 2013	ACC Makalah seminar Hasil	
7.	27 Juli 2013	Revisi Bab III, Bab II dan Bab I	
8.	1 Agustus 2013	ACC Bab III, Bab II dan Bab I	
9.			

Malang, 27 Agustus 2013
Dosen Pembimbing I

Joseph Dedy Irawan, ST, MT
NIP.197404162005011002



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

FORMULIR BIMBINGAN SKRIPSI

Nama : Ratih Yanuari
NIM : 0918069
Masa Bimbingan : 11 Mei 2013 s/d 11 November 2013
Judul Skripsi : **PENERAPAN PRINCIPAL COMPONENT ANALYSIS
UNTUK IDENTIFIKASI IRIS MATA**

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1.	30 Mei 2013	Revisi Bab I dan Bab II	
2.	24 Juli 2013	Flowchart Blok Diagram	
3.	26 Juli 2013	Revisi Bab III, Bab IV, Bab V ACC Bab I dan Bab II	
4.	29 Juli 2013	Revisi Bab III dan Bab IV ACC Bab V	
5.	30 Juli 2013	ACC Bab III & Bab IV Makalah seminar Hasil	
6.	31 Juli 2013	ACC Makalah seminar Hasil	
7.	12 Agustus 2013	ACC Kompre	
8.			
9.			

Malang, 13 Agustus 2013
Dosen Pembimbing II

Febriana Santi Wahyuni, S.Kom, M.Kom
NIP.P. 1031000425



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

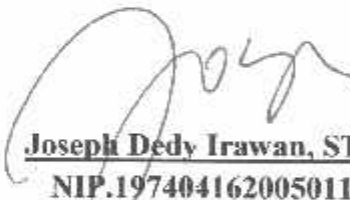
Nama : Ratih Yanuari
NIM : 09.18.069
Jurusan : Teknik Informatika S-1
Judul : **Penerapan *Principal Component Analysis* Untuk Identifikasi Iris Mata**

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1)
pada :

Hari : Selasa
Tanggal : 20 Agustus 2013
Nilai : 85.5 (A)


Panitia Ujian Skripsi :

Ketua Majelis Penguji


Joseph Dedy Irawan, ST, MT
NIP.197404162005011002

Anggota Penguji :

Penguji Pertama


Sandy Nataly Mantja, SKom.
NIP.1030800418

Penguji Kedua


Survo Adi Wibowo, ST, MT
NIP.P.1031000438



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

FORMULIR PERBAIKAN SKRIPSI

Nama : Ratih Yanuari
NIM : 0918069
Jurusan : Teknik Informatika S-1
Judul : Penerapan *Principal Component Analysis* Untuk Identifikasi Iris Mata

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	20 Agustus 2013	1. Cetak miring pada kata asing 2. Tujuan dan manfaat 3. Numbering/alphabet penulisan 4. Flowchart 5. Notifikasi pada program 6. Daftar Pustaka	
2.	Penguji II	20 Agustus 2013	1. Kesimpulan dan Abstraksi 2. Flowchart 3. Notifikasi pada program 4. Ambang batas jarak	

Penguji Pertama

Sandy Nataly Mantja, SKom.
NIP.1030800418

Penguji Kedua

Suryo Adi Wibowo, ST,MT
NIP.P. 1031000438

Dosen Pembimbing I

Joseph Dedy Irawan, ST, MT
NIP.197404162005011002

Dosen Pembimbing II

Febriana Santi Wahyuni,SKom,MKom
NIP. 1031000425



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor : ITN-78/T.INF/TA/2013
Lampiran : -
Perihal : Bimbingan Skripsi

11 Mei 2013

Kepada : Yth. Sdr. Joseph Dedy Irawan, ST, MT
Dosen Pembimbing Program Studi Teknik Informatika S1
Institut Teknologi Nasional
M a l a n g

Dengan hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : RATIH YANUARI
Nim : 0918069
Prodi : Teknik Informatika S1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu 6 (enam) bulan, terhitung mulai tanggal ;

11 Mei 2013 – 11 Nopember 2013

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik, Program Studi Teknik Informatika S1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S1
Kl. 100
Joseph Dedy Irawan, ST, MT
NIP : 197404162005021002
Form S-4a



FORMULIR PERBAIKAN UJIAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Informatika, maka perlu adanya perbaikan untuk mahasiswa :

Nama : PATIH YANUARI
NIM : 0918069
Perbaikan Meliputi : _____

1. KATA ASING DICETAK MIRING

2. TUJUAN, MANFAAT


3. TIDAK PERCH MENGUNAKAN BOTTON DAN PENOLITAN
ILAIH. SUMBER NUMERIK / ALFABETH

4. FLOWCHART /

5. TAMBAH KOTAK KERTAS 2-FOL SEJUA DI PERCH

6. DAFTAR PUSTAKA

Malang, 20.8.13


(RANDY NATALY)



FORMULIR PERBAIKAN UJIAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Informatika, maka perlu adanya perbaikan untuk mahasiswa :

Nama : Rehli Yauari
NIM : _____
Perbaikan Meliputi : _____

- Penyajian citra yg benar sesuai
- kesimpulan dibayangi
- Flowchart 3.4
- Flowchart 3.3
- Penyajian ttd
- Notasi ttd yg tidak sesuai dan benar
- kesimpulan dan abstrak
- Abstrak ditulis sesuai dan tidak sesuai

Malang , _____

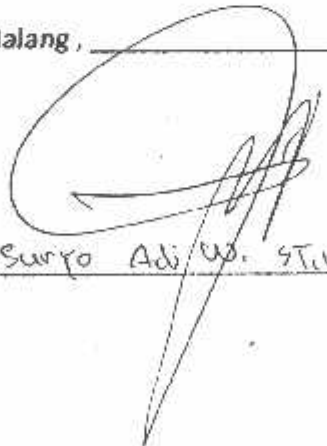

(Suryo Adi W. ST, MT)

FIG MAIN

Button Open

```
function btnOpen_Callback(hObject, eventdata, handles)
[FileName, PathName] = uigetfile('*.jpg', 'Silahkan Pilih Gambar Iris
Mata...');

    frm = guidata(gcbo);
    namaFile = strcat(PathName , FileName);
    if isequal(FileName,0)
        return;
    end;
    imgFile = imread(namaFile);
    set(frm.figMain, 'CurrentAxes', frm.axImages);
    set(imshow(namaFile));

    imgData=imfinfo(namaFile);
    set(frm.figMain, 'userdata', imgFile);
    set(frm.axImages, 'userdata', imgFile );

    set(frm.txtFile, 'String', strcat('', PathName));
    set(frm.txtNama, 'String', strcat('', FileName ));
    set(frm.txtW, 'String', ( imgData.Width ));
    set(frm.txtH, 'String', ( imgData.Height ));

    set(frm.btnProses, 'Enable', 'on');
    set (frm.txtTraining, 'String', strcat('', PathName));

function btnProses_Callback(hObject, eventdata, handles)
    frm = guidata(gcbo);
    TestDatabasePath = get(frm.txtFile, 'String' );
    TrainDatabasePath = get(frm.txtTraining, 'String' );
    disp (TrainDatabasePath);

    TestImage = strcat(TestDatabasePath, '\', get(frm.txtNama, 'String' ));
    im = imread(TestImage);

    I = training(TrainDatabasePath);
    [m, A, Eigenfaces] = eigenface(I);

    Train_Number = size(Eigenfaces,2);

    InputImage = imread(TestImage);
    temp = InputImage(:,:,1);

    [irow icol] = size(temp);
    xmean=mean(irow);
    ymean=mean(icol);

    xnew =irow-xmean*ones(Train_Number,1);
    ynew =icol-ymean*ones(Train_Number,1);

    set(frm.figMain, 'CurrentAxes', frm.axGrappCA);
```

```

subplot(xnew,ynew, 'o');

OutputName = recognition(TestImage, m, A, Eigenfaces,1);
set(frm.figMain, 'CurrentAxes', frm.axGrpPCA);

SelectedImage = strcat(TrainDatabasePath, '\', OutputName);
SelectedImage = imread(SelectedImage);

str = strcat('Nomer Citra yang dikenali adalah: ', OutputName);
imshow(im);
set(frm.figMain, 'CurrentAxes', frm.axHasil);
set(imshow(SelectedImage));
set(frm.figMain, 'CurrentAxes', frm.axHasil);
set(frm.btnSimpan, 'Enable', 'on');
disp(str)

function pushbutton5_Callback(hObject, eventdata, handles)
    TrainDatabasePath = uigetdir('D:\Program Files\MATLAB\R2006a\work\Iris
PCA\Train\','Silahkan Pilih Folder untuk Training Set ');
    frm = guidata(gcbo);
    set(frm.txtTraining, 'String', strcat('', TrainDatabasePath));

function btnUji_Callback(hObject, eventdata, handles)
    frm = guidata(gcbo);
    set(frm.pnUji, 'Visible', 'on');
    set(frm.pn1, 'Visible', 'off');
    set(frm.pn2, 'Visible', 'off');
    set(frm.pn3, 'Visible', 'off');
    set(frm.pn4, 'Visible', 'off');

function btnUjiOpen_Callback(hObject, eventdata, handles)
[FileName, PathName] = uigetfile('*.jpg','Silahkan Pilih Gambar Iris
Mata...');

    frm = guidata(gcbo);
    namaFile = strcat(PathName, FileName);
    if isequal(FileName,0)
        return;
    end;
    imgFile = imread(namaFile);
    set(frm.figMain, 'CurrentAxes', frm.axImages);
    set(imshow(namaFile));

    imgData=imsize(imgFile);
    set(frm.figMain, 'userdata', imgFile);
    set(frm.axImages, 'userdata', imgFile );

    set(frm.txtFile, 'String', strcat('', PathName));
    set(frm.txtNama, 'String', strcat('', FileName));
    set(frm.edUjiFile, 'String', strcat('', namaFile));

    set(frm.txtW, 'String', ( imgData.Width ));
    set(frm.txtH, 'String', ( imgData.Height ));

```

```

set(frm.btnProsesUji, 'Enable', 'on');

set(frm.txtTraining, 'String', strcat('', PathName));

function pushbutton9_Callback(hObject, eventdata, handles)
    frm = guidata(gcbo);
    set(frm.pnlUji, 'Visible', 'off');
    set(frm.pnl1, 'Visible', 'on');
    set(frm.pnl2, 'Visible', 'on');
    set(frm.pnl3, 'Visible', 'on');
    set(frm.pnl4, 'Visible', 'on');

function btnProsesUji_Callback(hObject, eventdata, handles)
    frm = guidata(gcbo);
    TestDatabasePath = get(frm.txtFile, 'String') ;
    TrainDatabasePath = get(frm.txtTraining, 'String') ;
    TestImage = strcat(TestDatabasePath, '\', get(frm.txtName, 'String') );
    im = imread(TestImage);

    T = training(TrainDatabasePath);
    [m, A, Eigenfaces] = eigentface(T);

    Train_Number = size(Eigenfaces, 2);

    InputImage = imread(TestImage);
    temp = InputImage(:,:,1);

    [irow icol] = size(temp);
    xmean=mean(irow);
    ymean=mean(icol);

    xnew=irow-xmean*ones(Train_Number,1);
    ynew=icol-ymean*ones(Train_Number,1);

    set(frm.figMain, 'CurrentAxes', frm.axGrafPCA);
    %plot(xnew,ynew, 'o');

    OutputName = recognition(TestImage, m, A, Eigenfaces, 0);
    set(frm.figMain, 'CurrentAxes', frm.axGrafPCA);

    SelectedImage = strcat(TrainDatabasePath, '\', OutputName);
    SelectedImage = imread(SelectedImage);

    str = strcat('Nomer Citra yang dikenali adalah: ', OutputName);
    imshow(im);
    set(frm.figMain, 'CurrentAxes', frm.axHasil);
    set(handles.SelectedImage);
    set(frm.figMain, 'CurrentAxes', frm.axHasil);
    set(frm.btnSimpan, 'Enable', 'on');

    tab = (strcat(TrainDatabasePath, '\', strcat(OutputName, '-.jpg', '.txt')));
    texts = readtextfile(tab);

```

```

txtID = get(frm.ediD, 'String' ) ;
if (strcmp([strcat(txtID, '.jpg'), OutputName])
    set(frm.txtHasilUji, 'String', texts);
else
    set(frm.txtHasilUji, 'String', 'Data Iris Tidak Sesuai dengan ID...');
end;
disp(texts);

```

MAIN

```

function Untitled_6_Callback(hObject, eventdata, handles)
guidata(profile)

```

```

function Untitled_7_Callback(hObject, eventdata, handles)
guidata(sistem)

```

```

function Untitled_2_Callback(hObject, eventdata, handles)
guidata(figmain)

```

```

function Untitled_3_Callback(hObject, eventdata, handles)
respon = input('Titled', 'Konfirmasi Keluar');
switch lower(respon)
case 'tidak'
    %tidak ada aksi
case 'ya'
    close
end

```

```

function axes2_CreateFcn(hObject, eventdata, handles)
imshow('awal.jpg')

```

SISTEM

```

function axes4_CreateFcn(hObject, eventdata, handles)
imshow('sistem.jpg')

```

PROFILE

```

function axes5_CreateFcn(hObject, eventdata, handles)
imshow('profile.jpg')

```

EIGENFACE

```

function [m, A, Eigenfaces] = eigenface(T)

% mean image
m = mean(T, 2);
% average m = (1/P)*sum(T(:,s))    (j = 1 : P)
Train_Number = size(T, 2);

% deviation images
A = [];
for i = 1 : Train_Number
    temp = double(T(:,i)) - m; % difference image    training set Ai = Ti - m
    A = [A temp]; % Merging centered images
end
L = A'*A; % covariance matrix C=A*A'.
[V D] = eig(L); % Diagonal eigenvalues    I=A'*A    C=A*A'.

```

```

% Sorting eigenvalues
L_eig_vec = [];
for i = 1 : size(V,2)
    if( D(i,i)>1 )
        L_eig_vec = [L_eig_vec V(:,i)];
    end
end

% eigenvectors    covariance matrix 'C'

Eigenfaces = A * L_eig_vec; % A: centered image

```

EXIT

```

function exit_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = 'Yes';

guidata(hObject, handles);
if(nargin > 3)
    for index = 1:2:(nargin-3);
        if nargin-3==index, break, end
        switch lower(varargin{index})
            case 'title'
                set(hObject, 'Name', varargin{index+1});
            case 'string'
                set(handles.text1, 'String', varargin{index+1});
            end
        end
    end
end

% Determine the position of the dialog - centered on the callback figure
% if available, else, centered on the screen
FigPos=get(0,'DefaultFigurePosition');
OldUnits = get(hObject, 'Units');
set(hObject, 'Units', 'pixels');
OldPos = get(hObject, 'Position');
FigWidth = OldPos(3);
FigHeight = OldPos(4);
if isempty(gcf)
    ScreenUnits=get(0,'Units');
    set(0,'Units','pixels');
    ScreenSize=get(0,'ScreenSize');
    set(0,'Units',ScreenUnits);

    FigPos(1)=1/2*(ScreenSize(3)-FigWidth);
    FigPos(2)=2/3*(ScreenSize(4)-FigHeight);
else
    GCBFOldUnits = get(gcf, 'Units');
    set(gcf, 'Units', 'pixels');
    GCBFPos = get(gcf, 'Position');
    set(gcf, 'Units', GCBFOldUnits);
    FigPos(1:2) = [(GCBFPos(1) + GCBFPos(3) / 2) - FigWidth / 2, ...
        (GCBFPos(2) + GCBFPos(4) / 2) - FigHeight / 2];
end
FigPos(3:4)=[FigWidth FigHeight];

```



```

set(hObject, 'Position', FigPos);
set(hObject, 'Units', OldUnits);

% Show a question icon from dialogicons.mat - variables questIconData
% and questIconMap
load dialogicons.mat

IconData=questIconData;
questIconMap(256,:) = get(handles.figure1, 'Color');
IconCMap=questIconMap;

Img=image(IconData, 'Parent', handles.axes1);
set(handles.figure1, 'Colormap', IconCMap);

set(handles.axes1, ...
    'Visible', 'off', ...
    'YDir'    , 'reverse' , ...
    'XLim'    , get(Img,'XData'), ...
    'YLim'    , get(Img,'YData') ...
    );

% Make the GUI modal
set(handles.figure1,'WindowStyle','modal')

% UIWAIT makes exit wait for user response (see UIRESUME)
uiwait(handles.figure1);

```

READ TEXT FILE

```

function tab=readTextFile(filename)

% Read a text file into a matrix with one row per input line
% and with a fixed number of columns, set by the longest line.
% Each string is padded with NUL (ASCII 0) characters
%
% open the file for reading

ip = fopen(filename,'rt');          % 'rt' means read text
if (ip < 0)
    error('could not open file');    % just abort if error
end;
% find length of longest line
max=0;                               % record length of longest string
cnt=0;                               % record number of strings
s = fgetl(ip);                       % get a line

while (ischar(s))                   % while not end of file
    cnt = cnt+1;
    if (length(s) > max)             % keep record of longest
        max = length(s);
    end;
    s = fgetl(ip);                  % get next line
end;

```

```

% rewind the file to the beginning
frewind(ip);
% create an empty matrix of appropriate size
tab=char(zeros(cnt,max)); % fill with ASCII zeros
% load the strings for real
cnt=0;
s = fgetl(ip);

while (ischar(s))

    cnt = cnt+1;
    tab(cnt,1:length(s)) = s; % slot into table
    s = fgetl(ip);

end;
% close the file and return
fclose(ip);

return;

```

REGOCNATION

```

function OutputName = recognition(TestImage, m, A, Eigenfaces,mode)
TrainingImages = [];
Train_Number = size(Eigenfaces,2);
for i = 1 : Train_Number
    temp = Eigenfaces'*A(:,i);
    TrainingImages = [TrainingImages temp];
end

% Extracting the PCA features
InputImage = imread(TestImage);
temp = InputImage(:,:,1);

[irow icol] = size(temp);
InImage = reshape(temp',irow*icol,1);
Difference = double(InImage)-m; % Centered test image
ProjectedTestImage = Eigenfaces'*Difference; % test image feature vector
disp( InputImage );
% Euclidean distances

Euc_dist = [] ;
T_dist = [] ;
hold on;
for i = 1 : Train_Number
    q = TrainingImages(:,i) ;
    disp( q );
    temp = ( norm( ProjectedTestImage - q )) ^ 2;
    Euc_dist = [Euc_dist temp] ;
    T_dist = [temp T_dist] ;
    plot ( Temp , 'o' ) ;
end

[Euc_dist_min , Recognized index] = min(Euc_dist);

```
