

16.21314/02

TUGAS AKHIR

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK PENGENALAN WAJAH MENGGUNAKAN METODE FISHERFACE



RSIF
005.1
Duu
P-1

2001

Disusun oleh :

Fitri Damayanti

2695100069

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2001**

PERPUSTAKAAN ITS	
Tgl. Terima	03/01/02
Terima Oleh	H
No. Agenda Frp.	21-4298



PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK PENGENALAN WAJAH MENGGUNAKAN METODE FISHERFACE

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Komputer

Pada

Jurusan Teknik Informatika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya

Mengetahui / Menyetujui,

Dosen Pembimbing I



Dr. Ir. Arif Djunaidy, M.Sc.

NIP. 131 633 403

Dosen Pembimbing II



Rully Soelaiman, S.Kom

NIP. 132 085 802

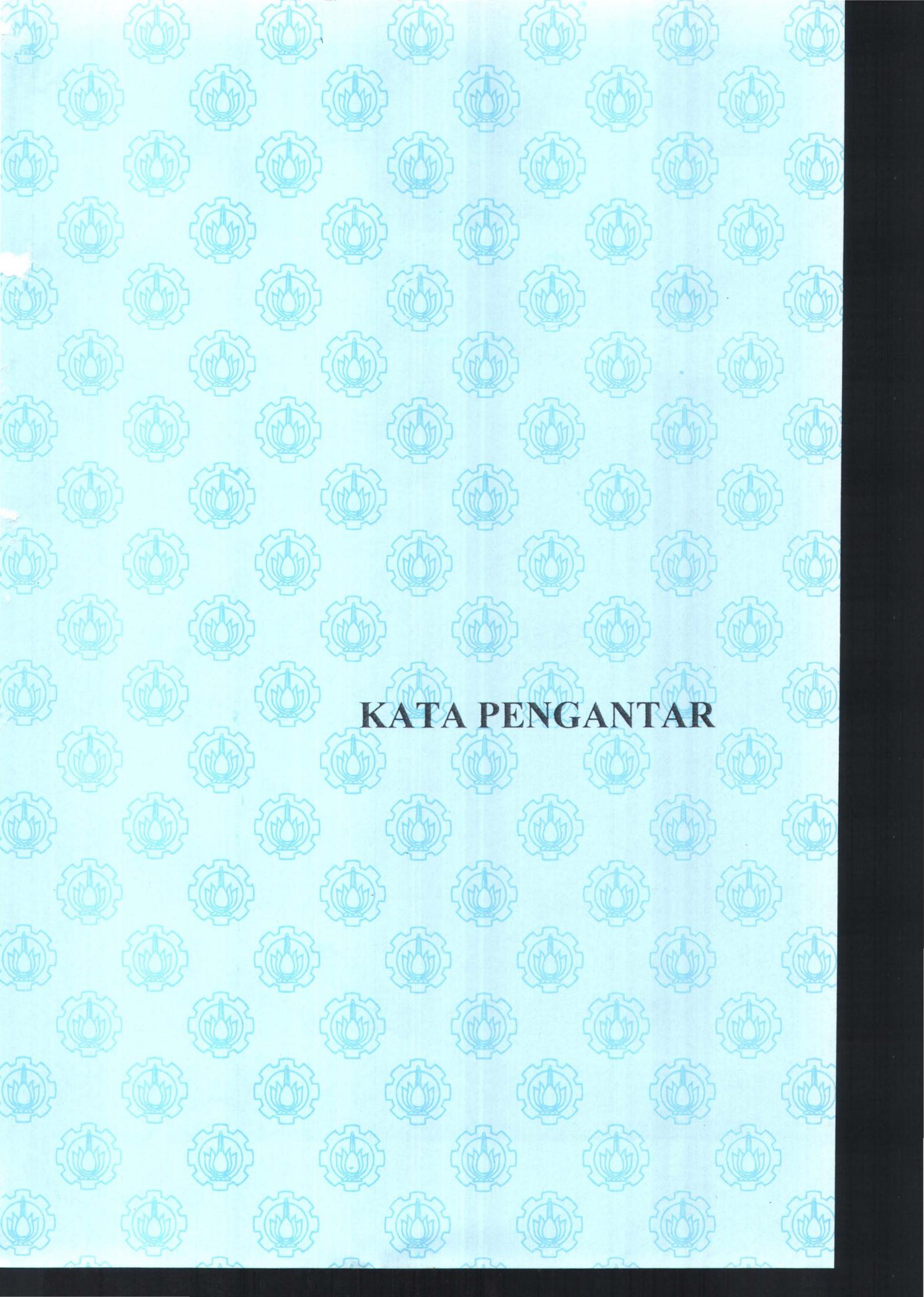
Surabaya
Agustus, 2001

Bismillahirrahmanirrahim

"Hai orang-orang yang beriman, bertakwalah kepada Allah sebenar-benar takwa kepada-Nya, dan janganlah sekali-kali kamu mati melainkan dalam keadaan beragama Islam (QS:3.102)"

"Dan hendaklah ada diantara kamu segolongan umat yang menyeru kepada kebajikan, menyuruh kepada yang ma'ruf dan mencegah dari yang munkar, merekalah orang-orang yang beruntung (QS:3.104)"

*Kupersembahkan Untuk
Ayahanda, Ibunda dan Suami Tercinta*



KATA PENGANTAR

KATA PENGANTAR

Alhamdulillah Rabbil'aalamin segala puji bagi Allah serta Sholawat dan salam bagi Nabi Muhammad atas berkat dan rahmat Allah SWT telah diselesaikan Tugas Akhir

Tugas Akhir ini tidak akan mampu diselesaikan tanpa dukungan dari semua pihak baik moril maupun materiil. Pada kesempatan ini saya menyampaikan terima kasih pada semua pihak yang telah memberikan dukungan dan dorongan.

- Bapak Dr.Ir.Arif Djunaidy,M.Sc selaku dosen pembimbing dan ketua jurusan Teknik Informatika, yang selalu meluangkan waktu untuk memberikan bimbingan serta dorongan dan motivasi dalam menyelesaikan tugas akhir ini.
- Bapak Rully Soelaiman,S.Kom selaku dosen pembimbing yang telah memberikan ide, bimbingan dan bantuan selama pengerjaan tugas akhir ini.
- Ibu Nanik Suciati,M.Kom selaku dosen wali.
- Seluruh staf pengajar Teknik Informatika atas ilmu yang diberikan selama masa perkuliahan.
- Staf tata usaha jurusan Teknik Informatika atas bantuannya.
- Ayahanda tersayang yang selalu memberikan kasih sayang, bimbingan, dorongan, dan do'a.
- Ibunda tercinta yang tidak henti-hentinya menyayangi, mendo'akan dan memberi semangat serta nasehat dengan ikhlas tanpa pamrih.
- Sang suami tersayang dan tercinta yang dengan setia mendengarkan keluh-kesah dan omelanku,yang memberikan motivasi dan do'a, dan selalu setia dalam suka maupun duka.
- Bapak dan Ibu Muin atas kasih sayang , nasehat dan bimbingannya.
- Adik-adikku Ahmad dan Wawan atas bantuan dan do'anya. Rajin-rajinlah beribadah dan belajar semoga menjadi anak yang Sholeh. Amin.

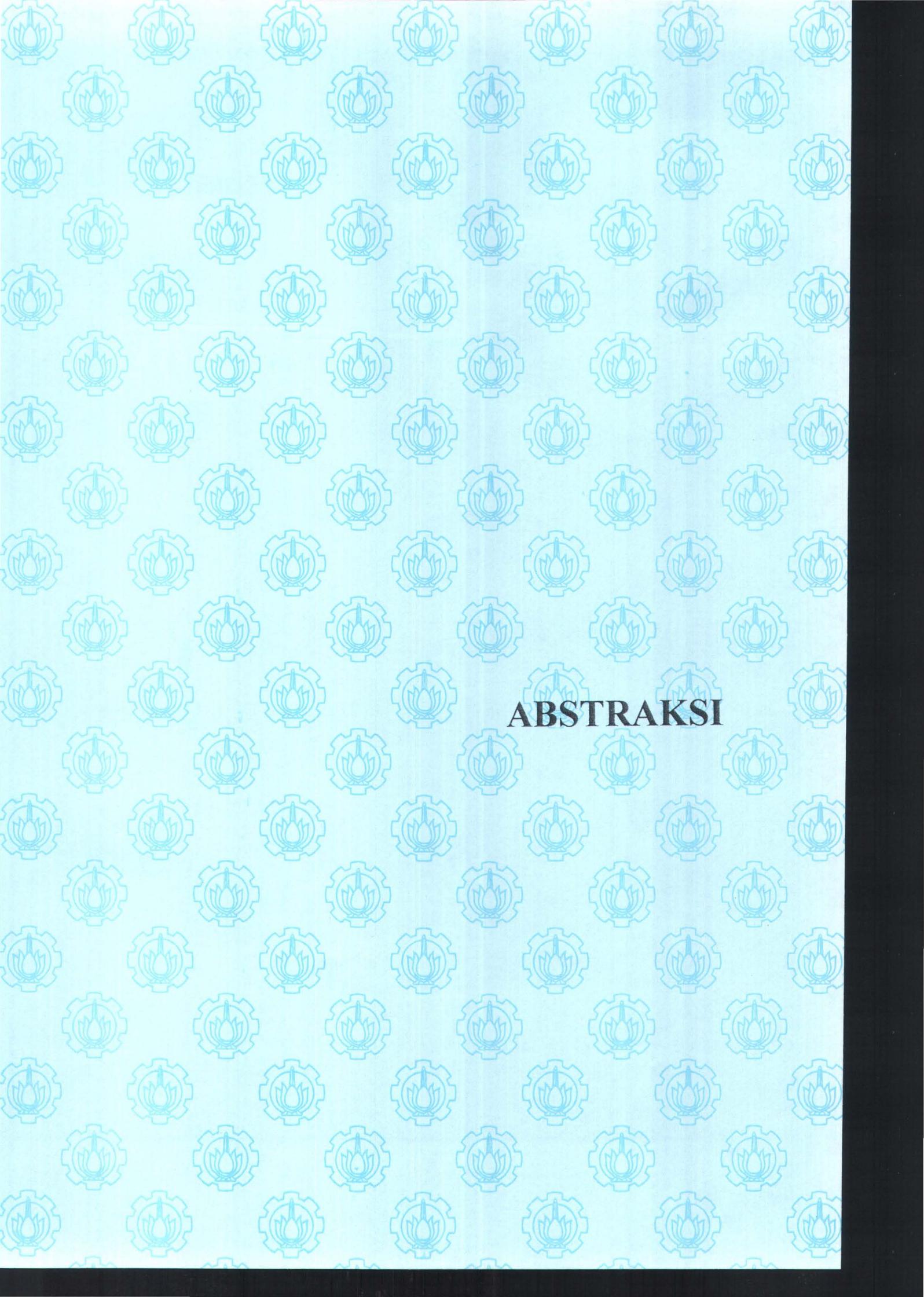
- Naim dan Yuli atas bantuan dan motivasinya. Cepetan Tugas Akhirnya dikerjakan. Semoga cepet nyusul (lulus dan mendapat jodoh).
- Anak-anakku Lyha dan Ita yang setia mendengarkan keluh-kesahku dan selalu memberi motivasi, semoga cepet lulus.
- Mbah Imam atas ilmu yang diberikan, serta bantuan dan motivasinya.
- Mukib, Yuni, Eny, Isty, Hida, Izzah, Saroh, Elok atas bantuan dan kebersamaan kita selama ini.
- Mbak Zaenab, Mbak Uswah, Mbak Niswah, Mbak Nurul yang telah membimbing dan mengarahkan untuk menuju jalan yang lurus dan benar.
- Pram, Eko, Fauzi, Anib, Salim, Hardi, Gandi dan teman-temanku C-0B lainnya atas bantuannya.
- Teman-teman kostku GK-70 atas bantuan dan motivasinya .
- Pihak-pihak yang ikut membantu terwujudnya tugas akhir ini.

Tugas akhir ini belum sampai pada tahap sempurna, maka saran dan kritik dari pembaca sangat diharapkan. Semoga karya ini dapat memberikan manfaat dan dicatat sebagai ilmu yang bermanfaat.

Amin ya Robbal Alamiin.

Surabaya, 7 Agustus 2001

Fitri Damayanti



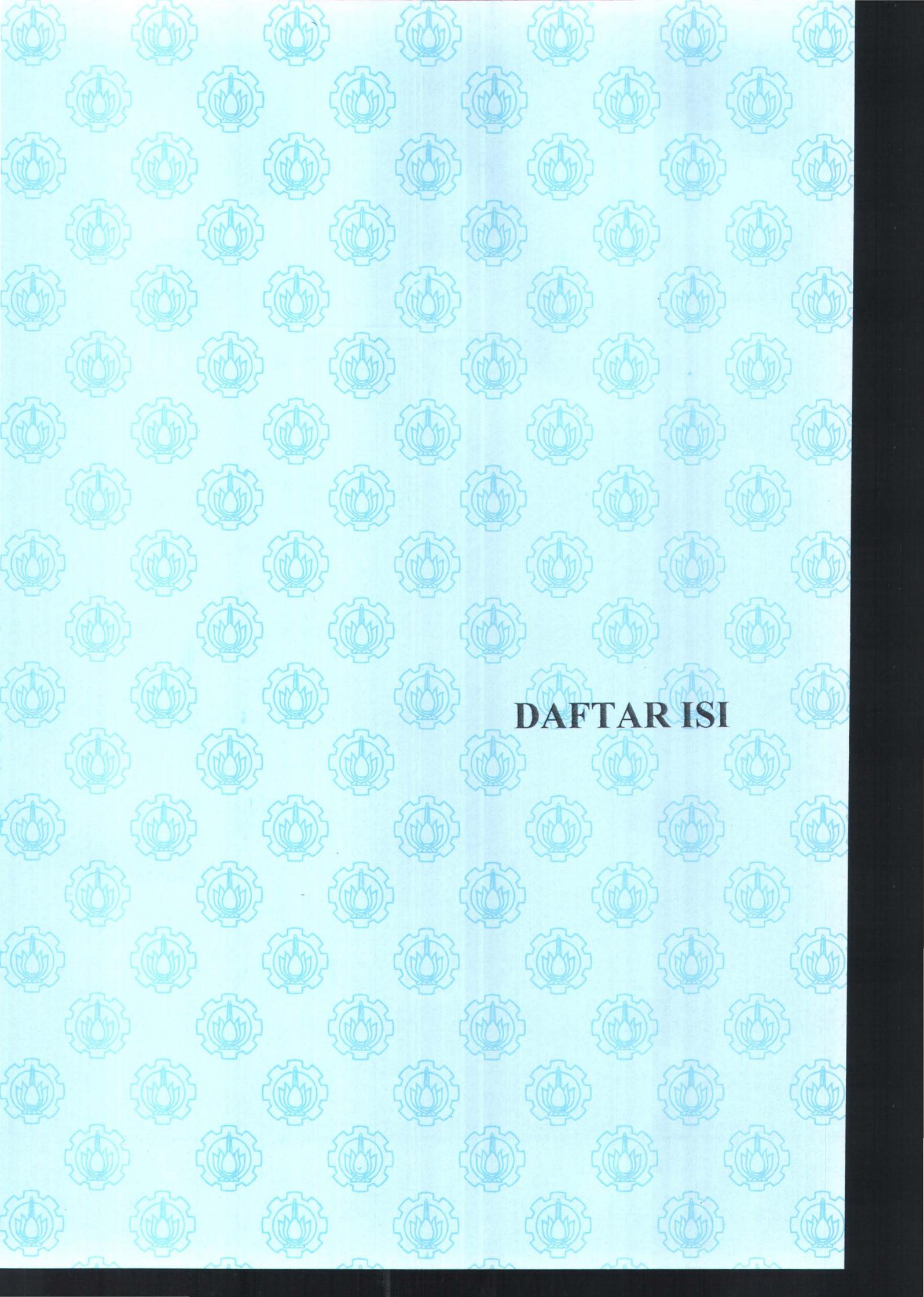
ABSTRAKSI

ABSTRAKSI

Dalam perkembangan teknologi komputer dewasa ini, banyak aplikasi yang menggunakan citra sebagai sumber informasi maupun dalam hal komunikasi data seringkali menggunakan citra sebagai sarana komunikasi, secara khusus adalah citra wajah. Sistem pengenalan wajah merupakan sistem yang mudah digunakan dan lebih berpotensi untuk tidak dapat ditembus.

Dalam tugas akhir ini, dirancang dan dibuat satu sistem pengenalan wajah dengan masukan berupa citra wajah yang berukuran tertentu, dengan menggunakan metode Fisherface. Sistem ini terdiri dari dua tahapan utama, yaitu tahap pelatihan dan tahap pengenalan. Tahap pelatihan dimaksudkan untuk melatih citra wajah yang telah ada didalam basis data untuk dicari nilai Transformasi Fisher's Linear Discriminant (FLD) dan bobot wajah pelatihan. Sedangkan tahap pengenalan akan mengenali wajah berdasarkan informasi hasil pelatihan. Uji coba dilakukan dengan menggunakan dua tipe data pelatihan, yaitu: data pelatihan terurut dan data pelatihan terkendali. Data pelatihan terurut diambil dari data pelatihan sesuai dengan nomor urutan datanya tanpa memandang variasi data yang ada. Data pelatihan terkendali dipilih secara manual dan mempertimbangkan variasi data yang ada tanpa memandang urutan indeks data.

Hasil uji coba dengan data pelatihan terurut menunjukkan 80% sedangkan dengan data pelatihan terkendali menunjukkan 88%. Kemampuan sistem pengenalan wajah ini sangat tergantung dari jumlah variasi data pelatihan yang digunakan dan juga tergantung pada kemiripan wajah tiap-tiap orang. Jumlah variasi yang memadai akan memberikan tingkat keberhasilan pengenalan wajah tinggi bila dibandingkan dengan variasi data pelatihan yang kurang memadai. Sistem yang telah berhasil dibuat dapat dimanfaatkan sebagai dasar bagi pengembangan pengaman elektronik yang berbasis pada pengenalan wajah.



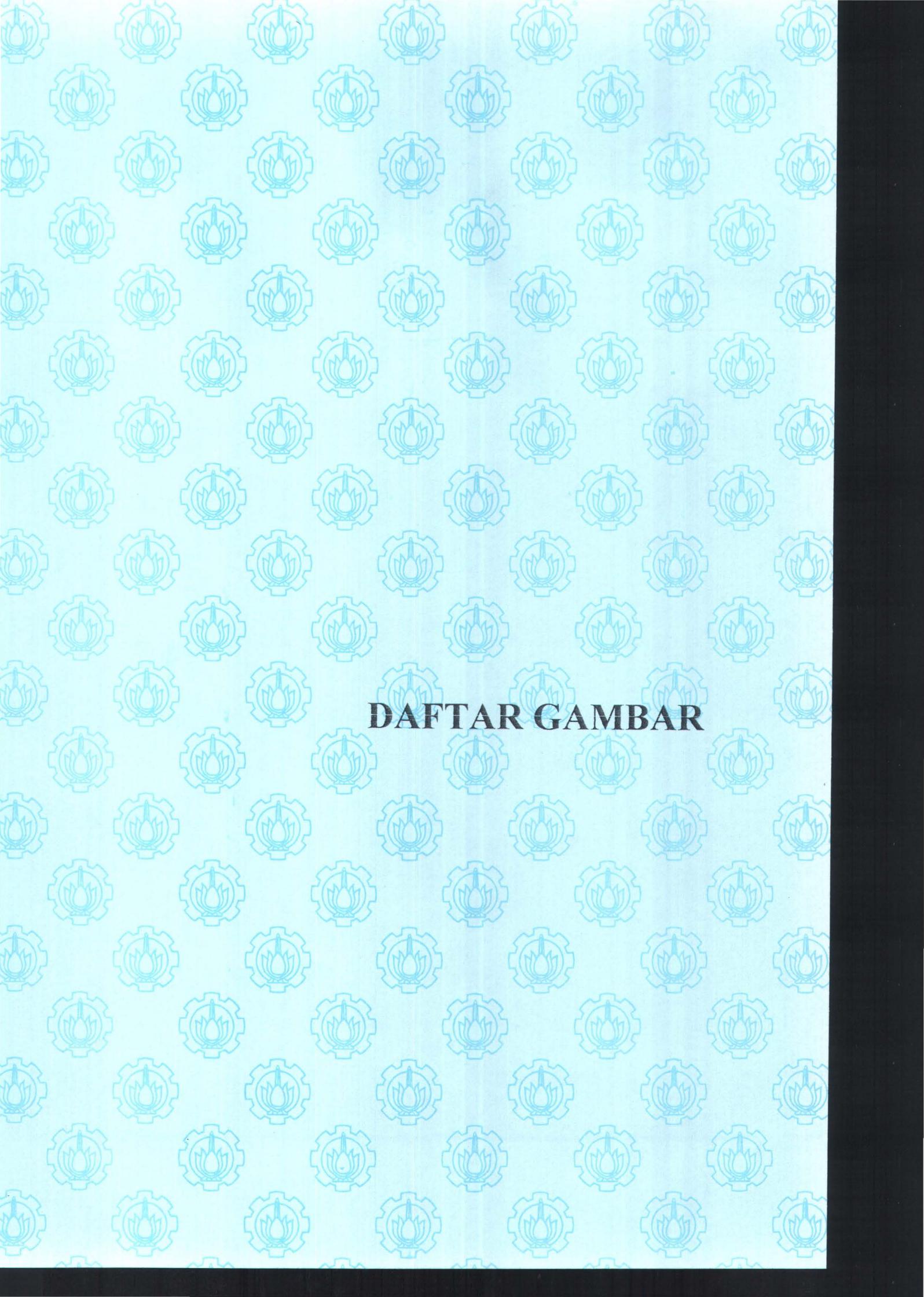
DAFTAR ISI

DAFTAR ISI

Kata Pengantar	i
Abstraksi	iii
Daftar Isi	iv
Daftar Gambar	vii
Daftar Tabel	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan dan Manfaat	3
1.4 Batasan Masalah	3
1.5 Metodologi Pengerjaan Tugas Akhir	3
1.6 Sistematika Penulisan	5
BAB II PENGENALAN WAJAH MENGGUNAKAN METODE FISHERFACE	6
2.1 Teori Dasar	6
2.1.1 Wajah Dipandang Sebagai Sebuah Vektor	6
2.1.2 Ruang Lingkup Gambar	7
2.2 Transformasi Matriks	8
2.2.1 Matriks Normalisasi	9
2.2.2 Matriks Kovariansi	9
2.3 Nilai Karakteristik dan Vektor Karakteristik	10
2.3.1 Definisi Nilai Karakteristik dan Vektor Karakteristik	11
2.3.2 Penentuan Nilai Karakteristik dan Vektor Karakteristik	11
2.3.3 Transformasi Kemiripan	12
2.3.4 Metode Cyclic Jacobi untuk Matriks Simetris	14
2.4 Metode Fisherface	15

2.4.1	Algoritma Metode Fisherface	18
2.4.2	Matriks Bobot Wajah	23
2.4.2.1	Matriks Bobot Wajah Pelatihan	23
2.4.2.2	Matriks Bobot Wajah Uji Coba	24
BAB III PERANCANGAN DAN PEMBUATAN		
PERANGKAT LUNAK		25
3.1	Tujuan dan Sasaran Sistem	25
3.2	Perancangan Perangkat Lunak	25
3.2.1	Perancangan Data	26
3.2.1.1	Data Masukan	26
3.2.1.2	Data Proses	26
3.2.1.3	Data Keluaran	29
3.2.2	Perancangan Proses	29
3.2.3	Perancangan Antarmuka	34
3.3	Pembuatan Perangkat Lunak	35
3.3.1	Implementasi Data	35
3.3.2	Implementasi Proses	37
3.3.3	Implementasi Antarmuka	41
BAB IV UJI COBA DAN EVALUASI SISTEM		43
4.1	Lingkungan Uji Coba	43
4.2	Data Uji Coba	44
4.3	Uji Coba dan Evaluasi Hasil	44
4.3.1	Uji Coba dengan Data Pelatihan terurut	46
4.3.2	Uji Coba dengan Data Pelatihan Terkendali	48
4.3.3	Evaluasi Hasil Uji Coba	50
4.3.4	Perbandingan Dengan Sistem Lain	51
BAB V KESIMPILAN		53

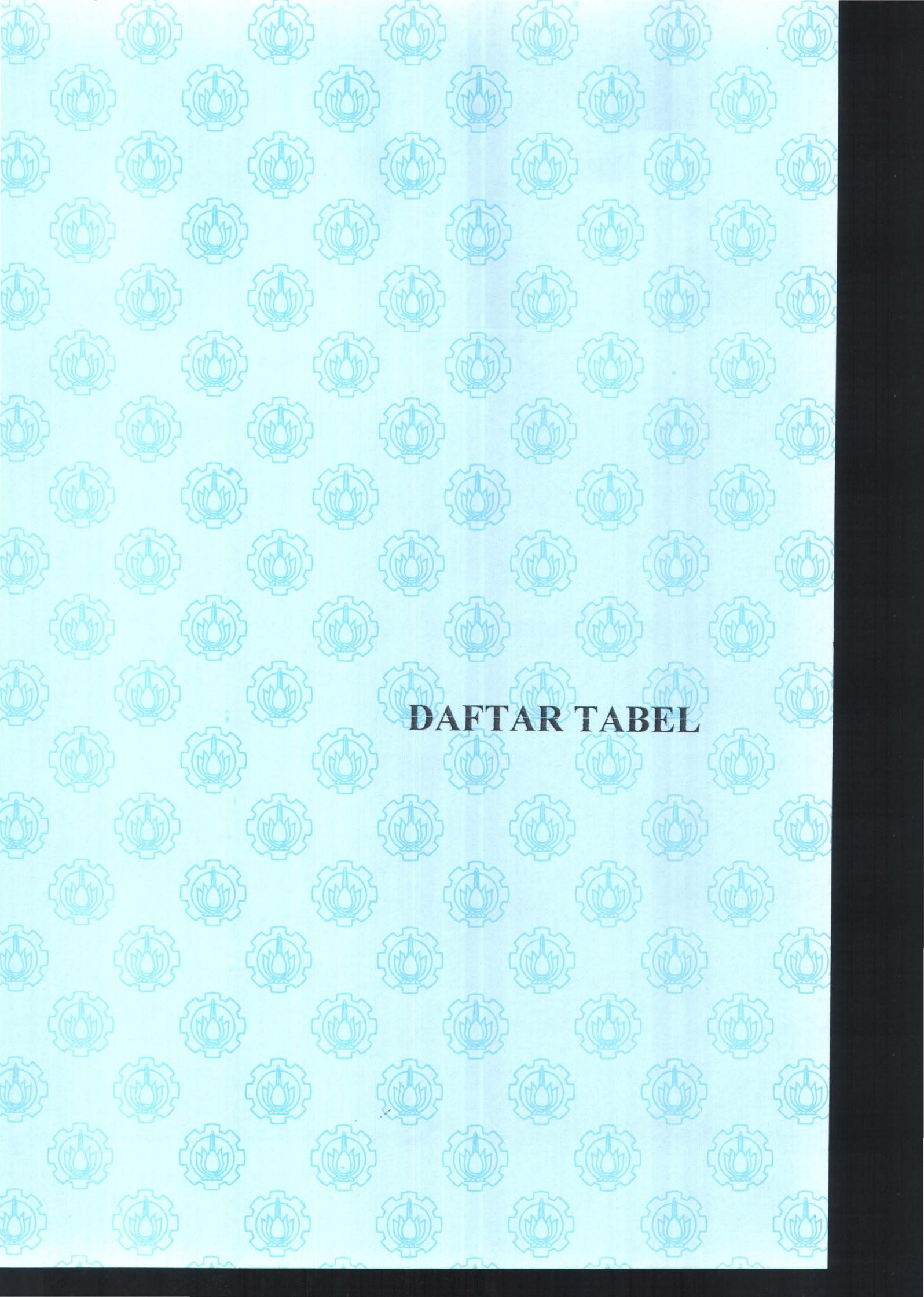
DAFTAR PUSTAKA	54
LAMPIRAN A : DATA PELATIHAN	A-1
LAMPIRAN B : PETUNJUK PEMAKAIAN PERANGKAT LUNAK	B-1



DAFTAR GAMBAR

DAFTAR GAMBAR

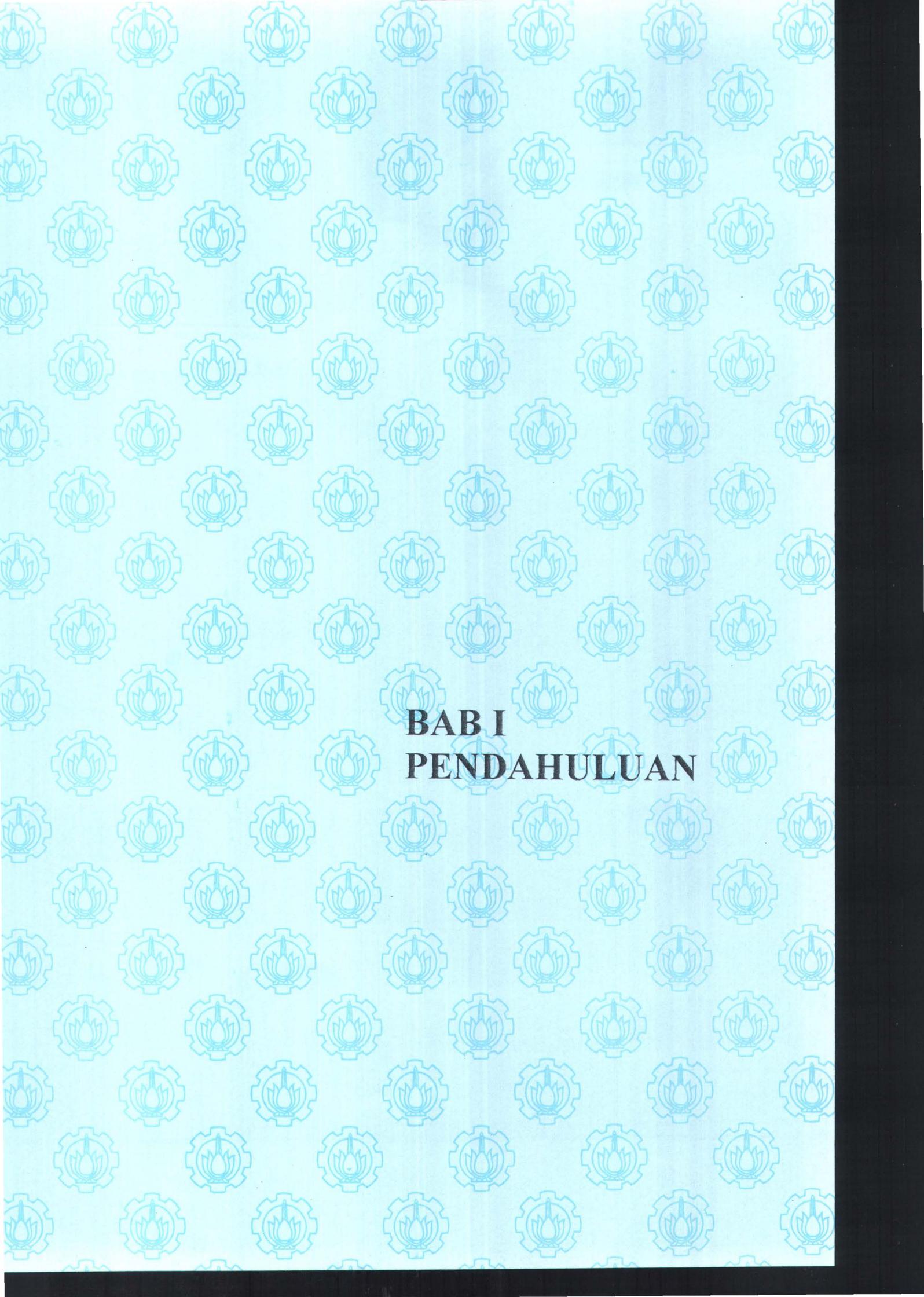
Gambar 2.1	Formasi vektor dari sebuah gambar
Gambar 2.2	Basis ruang lingkup gambar
Gambar 3.1	DFD level 0 subsistem pelatihan
Gambar 3.2	DFD level 1 subsistem pelatihan
Gambar 3.3	DFD level 2 subsistem pelatihan Metode PCA
Gambar 3.4	DFD level 2 subsistem pelatihan Metode FLD
Gambar 3.5	DFD level 0 subsistem pengenalan wajah
Gambar 3.6	DFD level 1 subsistem pengenalan wajah
Gambar 3.7	Menu utama
Gambar 3.8	Tampilan sebagian wajah pelatihan
Gambar B.1	Tampilan dari sebagian wajah training
Gambar B.2	Tampilan matriks hasil dari proses training
Gambar B.3	Tampilan hasil dari testing
Gambar B.4	Tampilan matriks dari hasil proses testing



DAFTAR TABEL

DAFTAR TABEL

Tabel 4.1	Hasil uji coba pengenalan wajah dengan data pelatihan terurut
Tabel 4.2	Hasil uji coba pengenalan wajah dengan data pelatihan terkendali
Tabel 4.3	Perbandingan keberhasilan pengenalan berdasarkan metode pemilihan data pelatihan yang digunakan
Tabel A.1	Data pelatihan



**BAB I
PENDAHULUAN**

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Sistem pengaman elektronik mengalami kemajuan pesat dalam beberapa dekade terakhir ini. Perkembangan ini didukung oleh kemajuan teknologi dibidang hardware, khususnya komputer. Dukungan yang diberikan komputer seperti kemudahan dibidang perancangan, simulasi, dan implementasi sistem pengaman elektronik tersebut.

Aplikasi sistem pengenalan biometrik (sistem pengenalan dengan penggunaan ciri khas pada diri manusia, seperti wajah, sidik jari, dan telapak tangan) untuk sistem pengaman mendapatkan perhatian yang serius akhir-akhir ini. Perhatian ini muncul karena ciri-ciri yang digunakan tersebut merupakan ciri yang unik untuk setiap manusia dan tidak dapat dipalsukan. Diantara ketiga ciri yang disebutkan diatas, ciri wajah merupakan ciri yang lebih kompleks karena banyaknya informasi yang terdapat padanya. Informasi-informasi itu meliputi bentuk wajah, jarak kedua mata, warna kulit, panjang dan lebar alis mata, dan informasi lainnya. Dengan demikian, penggunaan wajah sebagai kode akses untuk sistem pengaman elektronik lebih berpotensi untuk tidak dapat ditembus dibandingkan dengan penggunaan ciri yang lain.

Tugas akhir ini berisi tentang sistem pengenalan wajah dengan menggunakan metode Fisherface. Metode ini mampu menghasilkan presentase

total tingkat pengenalan yang lebih tinggi. Hal ini dapat terjadi karena dengan menerapkan metode Fisherface untuk mereduksi dimensi citra wajah ke dimensi ciri sebagai data masukan sistem, data-data pada citra wajah yang dapat mengganggu tidak ikut diproses lagi. Dengan demikian hanya ciri-ciri wajah yang penting saja yang disimpan dan diolah sistem untuk melakukan pengenalan terhadap objek wajah. Selain itu bila terdapat variasi ciri wajah yang akan dikenali, seperti ekspresi, tetap mampu dikenal dengan baik. Dapat dikatakan bahwa metode Fisherface yang mereduksi dimensi data masukan sistem dapat meningkatkan kinerja sistem, terutama dengan mempercepat proses pelatihan dan meningkatkan prosentase tingkat pengenalan objek.

1.2 PERUMUSAN MASALAH

Masalah utama dari pengenalan wajah adalah diberikan sebuah gambar atau citra yang berisi satu atau banyak pas foto wajah orang, kemudian gambar tersebut diidentifikasi masing-masing individu yang ada dengan menggunakan basis data dari wajah yang sudah ada. Kesulitan yang muncul dalam pengenalan pola adalah besarnya variasi pada input sistem. Perbedaan-perbedaan posisi dan arah pandangan pada input yang ada dari wajah orang yang sama harus tetap dikenali sebagai wajah orang yang sama pula. Metode Fisherface akan mereduksi data masukan sistem yang berupa dimensi citra wajah ke dimensi ciri, data yang mengganggu tidak ikut diproses. Setelah proses pelatihan, kemudian proses uji coba dilakukan sebuah foto yang tidak dikenal diinputkan ke dalam program dan dibandingkan dengan semua basis data.

1.3 TUJUAN DAN MANFAAT

Tujuan dari tugas akhir ini adalah mencoba mengenali suatu pas foto yang diinputkan dalam program dari pas foto yang tidak dikenal sebelumnya berdasarkan basis data yang ada dengan menggunakan metode Fisherface.

Manfaat yang bisa diperoleh dari pembuatan sistem ini adalah dasar bagi pembuatan sistem yang lebih besar dalam bidang pembuatan sistem keamanan di berbagai bidang.

1.4 BATASAN MASALAH

Agar pembahasan menjadi terfokus, dibuat ruang lingkup pembahasan sebagai berikut :

- Foto yang diinputkan harus merupakan foto oval dari wajah saja (pas foto)
- Foto merupakan gambar dengan skala keabuan (gray scale) dan mempunyai suatu ukuran tertentu, yaitu : 92 * 112 piksel.
- Foto yang dipergunakan berekstensi BMP.

1.5 METODOLOGI Pengerjaan Tugas Akhir

Pembuatan Tugas Akhir ini akan dilaksanakan dengan metodologi sebagai berikut :

◆ Studi Literatur

Studi literatur dilakukan dengan mencari dan mempelajari literatur-literatur yang berkaitan dengan *metode Fisherface* , teori-teori yang berhubungan

dengan sistem yang akan dibangun, desain sistemnya, dan bahasa pemrogramannya.

◆ Perancangan Sistem Perangkat Lunak

Melakukan perancangan sistem mulai dari Input/output, *pre-processing* data, teknik pengolahan data, perancangan struktur data, dan perancangan struktur pemrogramannya

◆ Pembuatan Perangkat Lunak

Setelah semua rancangan aplikasi tersedia maka selanjutnya diimplementasikan kedalam *source code* sesuai bahasa pemrograman yang dipilih.

◆ Uji Coba dan Evaluasi

Pengujian suatu perangkat lunak dilakukan guna mengetahui permasalahan yang terjadi setelah perangkat lunak tersebut selesai dikerjakan.

◆ Perbaikan dan Penyempurnaan Perangkat Lunak

Apabila dalam proses pengujian ditemukan kesalahan tau kekurangan maka perlu dilakukan perbaikan dan penyempurnaan. Setelah itu dilakukan lagi pengujian sampai aplikasi bisa benar-benar sempurna dan sudah layak dipakai sesuai dengan tujuannya.

◆ Penulisan Tugas Akhir

Pada tahap akhir dibuat suatu dokumentasi lengkap tentang aplikasi yang telah dibuat, dalam hal ini dibuat dalam bentuk Buku Tugas Akhir.

1.6 SISTEMATIKA PENULISAN

Uraian sistematika penulisan pada tugas akhir ini bertujuan supaya perancangan dan pembuatan perangkat lunak yang dibahas menjadi mudah dipahami, jelas dan sistematis untuk tiap-tiap bab atau sub bahasan. Secara kronologis uraian dalam tugas akhir ini disusun sebagai berikut :

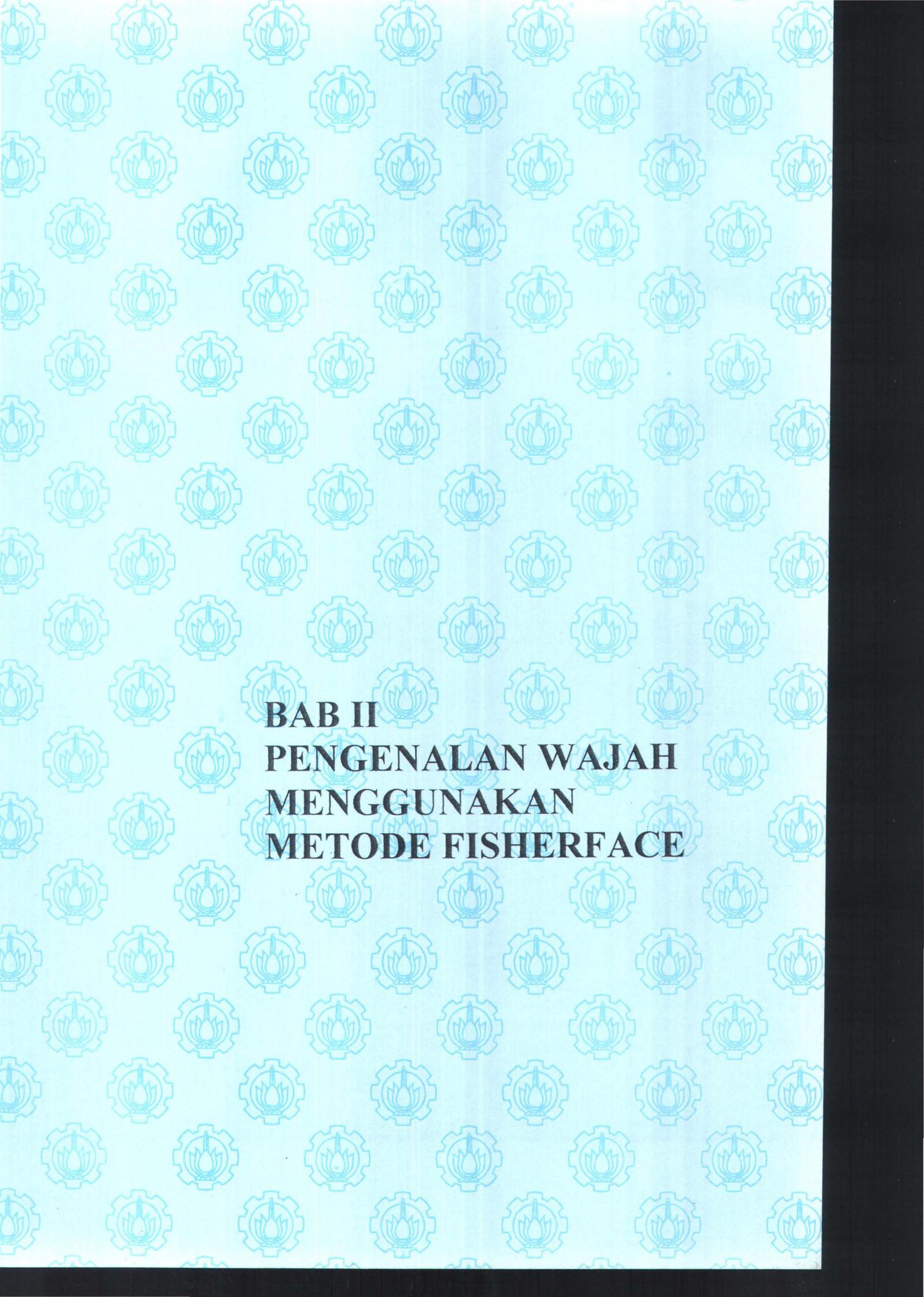
BAB I berisi pendahuluan, menguraikan tentang latar belakang, permasalahan, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan Tugas Akhir.

BAB II berisi pengenalan wajah menggunakan metode Fisherface, yang menguraikan teori-teori dan konsep-konsep dasar mengenai pengenalan wajah dari sebuah pas foto yang diinputkan dalam program berdasarkan basis data yang ada dengan menggunakan metode Fisherface.

BAB III menjelaskan tentang modul-modul perancangan dan pembuatan perangkat lunak yang meliputi prosedur-prosedur atau fungsi-fungsi program.

BAB IV menjelaskan tentang analisis hasil dari implementasi program.

BAB V menjelaskan tentang penutup yang berisi kesimpulan dan kemungkinan pengembangan lebih lanjut.



BAB II
Pengenalan Wajah
Menggunakan
Metode Fisherface

BAB II

Pengenalan Wajah Menggunakan Metode Fisherface

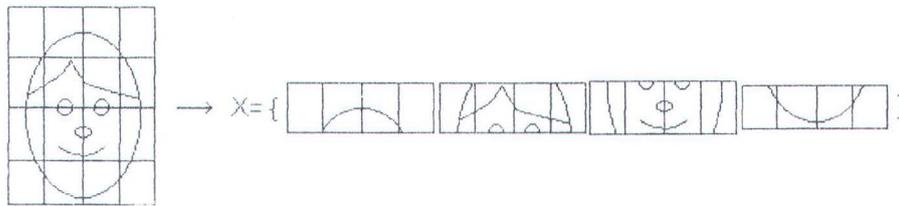
Dalam bab ini akan dibahas konsep-konsep dan teori-teori dasar tentang pengenalan wajah dengan menggunakan metode Fisherface yang berhubungan erat dengan pembuatan sistem perangkat lunak dan implementasi program. Dalam bab ini juga dibahas metode serta algoritma dari metode Fisherface

2.1 TEORI DASAR

Gambar merupakan kumpulan dari angka-angka, yang angka-angka tersebut membentuk sebuah matriks. Sehingga sebelum membahas lebih lanjut akan dikenalkan dulu tentang vektor dan matriks serta rumus-rumus untuk menghitungnya.

2.1.1 Wajah Dipandang Sebagai Sebuah Vektor

Sebuah wajah, yang juga merupakan sebuah gambar, dapat dipandang sebagai sebuah vektor. Apabila lebar dan tinggi gambar adalah w dan h piksel, maka banyaknya komponen dari vektor ini adalah $w \cdot h$. Setiap piksel dikodekan oleh satu komponen vektor. Konstruksi vektor dari sebuah gambar dibentuk oleh penggabungan sederhana, yaitu baris dari sebuah gambar diletakkan saling bersebelahan dengan baris-baris yang lain, seperti yang terlihat pada gambar 2.1.



Gambar 2.1 Formasi Vektor dari Sebuah Gambar

2.1.2 Ruang Lingkup Gambar

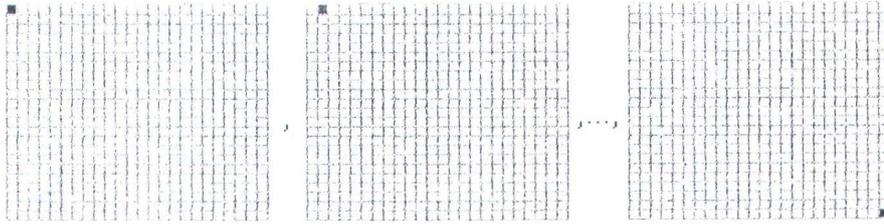
Vektor wajah yang telah dideskripsikan sebelumnya merupakan bagian dari sebuah ruang. Ruang ini adalah sebuah ruang lingkup gambar (*image space*), yaitu ruang dari semua gambar keseluruhan yang mempunyai dimensi $w \times h$ piksel. Basis dari ruang lingkup gambar dikomposisikan oleh vektor-vektor berikut, seperti terlihat pada gambar 2.2.

Semua wajah mirip satu sama lain. Mereka semua mempunyai dua mata, satu hidung, satu mulut, dua telinga, dan lain sebagainya yang terletak pada tempat yang sama. Akibatnya semua vektor wajah terletak pada tempat-tempat yang amat berdekatan dalam ruang lingkup gambar.

Oleh karena itu, sebuah ruang lingkup gambar yang penuh bukanlah sebuah ruang yang bagus untuk mendeskripsikan sebuah wajah. Tugas yang akan dipresentasikan di sini adalah bertujuan untuk membangun sebuah ruang lingkup wajah (*face space*) yang lebih dapat mendeskripsikan wajah. Vektor basis dari ruang lingkup wajah disebut sebagai komponen utama (*principal component*).

Dimensi dari ruang lingkup gambar adalah $w \times h$. Tentu saja semua piksel dari sebuah wajah tidak relevan, dan setiap piksel bergantung pada tetangganya.

Sehingga dimensi dari ruang lingkup wajah adalah kurang dari dimensi ruang lingkup gambarnya. Dimensi dari ruang lingkup wajah tidak dapat ditentukan, tetapi hal ini sudah dapat dipastikan bahwa dimensi dari ruang lingkup wajah akan jauh lebih kecil daripada ruang lingkup gambarnya.



Gambar 2.2 Basis Ruang Lingkup Gambar

2.2 TRANSFORMASI MATRIKS

Diberikan sebuah basis data wajah pelatihan yang terdiri dari K gambar. Ukuran gambar diasumsikan terdiri dari $N \times N$ piksel, dimana setiap piksel dikodekan dengan menggunakan 8 bit atau 256 tingkat keabuan (*gray level*). Setiap gambar dapat dipandang sebagai sebuah vektor kolom dengan ukuran $(N \times N) \times 1$.

Misalkan setiap gambar diberi notasi $\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_K$ yang merupakan vektor kolom $N^2 \times 1$. Bila sebuah gambar dalam wajah pelatihan dinotasikan dalam bentuk matriks, maka akan diperoleh matriks yang berordo $N^2 \times K$.

$\Gamma_n = (\Gamma_1 \ \Gamma_2 \ \Gamma_3 \ \dots \ \Gamma_K)^T$, dimana T adalah matriks transpose. Sedang isi dari tiap vektor kolom adalah

$$\Gamma_n = \begin{bmatrix} \gamma_{1,1} & \gamma_{1,2} & K & \gamma_{1,K} \\ \gamma_{2,1} & \gamma_{2,2} & K & \gamma_{2,K} \\ M & M & & M \\ \gamma_{N \times N,1} & \gamma_{N \times N,2} & K & \gamma_{N \times N,K} \end{bmatrix}$$

2.2.1 Matriks Normalisasi

Sebuah gambar harus sudah dinormalisasi terlebih dahulu untuk mendefinisikan sebuah lokasi penting dalam gambar tersebut. Normalisasi berpengaruh langsung pada dasar pengenalan wajah apabila semua wajah memiliki latar belakang sama dan variansi antar gambar sebagian besar merupakan sebuah variansi pengukuran fitur. Misalkan :

- $\Gamma = [\gamma_{ik}]$ adalah matriks berordo $N^2 \times K$ yang mendefinisikan himpunan wajah pelatihan.
- Vektor kolom ke-k dari matriks ini, γ_k , berkorespondensi pada wajah ke-k dari wajah pelatihan.
- Vektor γ_k dinormalisasi dengan rumus :

$$|\gamma_k| = \sqrt{\gamma_k^T \cdot \gamma_k}$$

2.2.2 Matriks Kovariansi

Matriks kovariansi adalah sebuah matriks simetris berordo $N^2 \times N^2$ dan merupakan sebuah pengukuran yang tidak hanya pada sebuah variansi, tetapi *kovariansi* dari vektor-vektor kolom.

- Wajah rata-rata, m_x , merupakan operasi rata-rata sederhana dari K kolom yang merepresentasikan sebuah wajah :

$$m_x = \frac{1}{K} \sum_{n=1}^K \Gamma_n$$

- Deviasi merupakan penyimpangan pengamatan terhadap rata-rata wajah :

$$\Phi_n = \Gamma_n - m_x$$

- Misalkan $\Phi_n = \Gamma_n - m_x$ dan $A = (\Phi_1 \ \Phi_2 \ \dots \ \Phi_K)$ adalah matriks wajah ternormalisasi, maka matriks kovariansi di atas dapat ditulis menjadi :

$C_x = AA^T$ adalah matriks *outer product* berordo $N^2 \times N^2$, atau

$L = A^T A$ adalah matriks *inner product* berordo $K \times K$, $K \ll N^2$.

- Matriks C_x dan L adalah matriks *simetris*, sebab unsur-unsur yang berada pada posisi simetris relatif terhadap diagonal utama adalah sama, sehingga dapat diilustrasikan bahwa $\Phi_p^T \Phi_q = \Phi_q^T \Phi_p$, untuk $p, q \in [1, K]$. Jadi dapat disimpulkan bahwa, matriks A $n \times n$ dikatakan simetris apabila $A^T = A$.

2.3 NILAI KARAKTERISTIK DAN VEKTOR KARAKTERISTIK

Dalam metode Fisherface dihitung juga nilai karakteristik dan vektor karakteristik, sehingga subbab ini akan membahas definisi vektor karakteristik dan nilai karakteristik sekaligus cara menghitungnya.



2.3.1 Definisi Nilai Karakteristik dan Vektor Karakteristik

Definisi 1 : Misalkan A adalah sebuah matriks $n \times n$. Sebuah matriks bukan nol, P , yang berukuran $n \times 1$ sedemikian rupa sehingga $AP = \lambda P$ dinamakan vektor karakteristik bagi A , sedangkan skalar λ dinamakan nilai karakteristik bagi A yang bersesuaian dengan P .

Persamaan pendefinisinya, yaitu :

$$AP = \lambda P \quad \text{ekivalen dengan}$$

$$\lambda P = AP \quad \text{atau} \quad (\lambda I - A)P = 0.$$

Persamaan terakhir akan mempunyai solusi bukan nol jika dan hanya jika

$$\det(\lambda I - A) = 0.$$

Hal ini memberi petunjuk tentang bagaimana cara memperoleh nilai karakteristik dan vektor karakteristik. Akan tetapi, apabila matriks A berukuran besar, maka cukup banyak memerlukan perhitungan sehingga cara demikian sangatlah rumit dan tidak efisien.

2.3.2 Penentuan Nilai Karakteristik dan Vektor Karakteristik

Untuk mendapatkan nilai karakteristik dan vektor karakteristik sekaligus, digunakan metode *Cyclic Jacobi* atau *Rotation Jacobi* yang telah dikembangkan oleh Jacobi [SAM-97]. Metode ini hanya bisa dikerjakan untuk matriks simetris.

Untuk mempermudah pencarian vektor karakteristik P yang bersesuaian dengan nilai karakteristik λ pada matriks berukuran besar, maka sebelum membahas metode *Cyclic Jacobi*, akan dibahas dulu definisi-definisi matematika

dasar dan aljabar linier. Sebab tanpa dasar ini, maka kita akan kesulitan untuk mendapatkan pengertian vektor karakteristik yang bersesuaian dengan nilai karakteristik dalam metode Cyclic Jacobi.

2.3.3 Transformasi Kemiripan

Transformasi Kemiripan (*Similarity Transform*) adalah untuk mentransformasikan matriks A ke matriks B yang *similar* dengan matriks A yang telah dinormalisasi, dimana B adalah matriks diagonal yang dikomposisikan dari nilai karakteristik $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n$. Tujuannya adalah untuk menunjukkan bahwa nilai karakteristik dari matriks diagonal B adalah nilai karakteristik dari A.

Definisi 2 : Kemiripan (*Similarity*)

A dan B adalah dua matriks $n \times n$ dikatakan *similar* bila $B = P^{-1}AP$.

Bukti bahwa nilai karakteristik dari matriks diagonal B adalah nilai karakteristik dari A dapat diilustrasikan sebagai berikut :

$$\text{Bukti : } B = P^{-1}AP = \begin{bmatrix} \lambda_1 & 0 & 0 & \Lambda & 0 \\ 0 & \lambda_2 & 0 & \Lambda & 0 \\ 0 & 0 & 0 & & M \\ M & M & & 0 & M \\ 0 & 0 & \Lambda & \Lambda & \lambda_n \end{bmatrix}$$

Misalkan P_1, P_2, \dots, P_n adalah kolom dari P, atau $P = [P_1 P_2 \dots P_n]$ adalah vektor karakteristik yang bersesuaian dengan nilai karakteristik, maka

$$A[P_1 P_2 P_3 \dots P_n] = [P_1 P_2 P_3 \dots P_n] \begin{bmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

Dengan menulis kembali dalam istilah kolom-kolom, maka

$$[AP_1 \ AP_2 \ AP_3 \ \dots \ AP_n] = [\lambda_1 P_1 \ \lambda_2 P_2 \ \lambda_3 P_3 \ \dots \ \lambda_n P_n]$$

Jadi diperoleh $AP_i = \lambda_i P_i$, $i = 1, 2, 3, \dots, n$.

Yang berarti bahwa vektor P_i adalah vektor karakteristik yang bersesuaian dengan nilai karakteristik λ_i . Jadi terbukti bahwa, jika A dan B adalah 2 matriks *similar*, maka mereka mempunyai nilai karakteristik yang sama.

Dalam implementasi program, untuk mendapatkan nilai karakteristik dan vektor karakteristik dengan menggunakan matriks P *invers*, P^{-1} , akan mendapatkan kesulitan, sebab nilai karakteristik di sini harus tidak boleh nol, sehingga harus diusahakan agar P adalah *matriks ortogonal*.

Agar P *ortogonal*, maka A harus *simetris*. Pada sub bab sebelumnya sudah dibuktikan bahwa A adalah matriks simetris. Oleh karena A matriks $n \times n$ adalah matriks *simetris*, maka matriks $A^T = A$.

Dari uraian di atas, maka terbukti bahwa P adalah matriks yang dapat didiagonalkan secara ortogonal. Oleh karena P adalah *matriks ortogonal*, maka matriks $P^{-1} = P^T$. Akibatnya, matriks diagonal B pada uraian sebelumnya yang semula ditulis :

$$B = P^{-1}AP \text{ ekuivalen dengan } B = P^TAP$$

Persamaan yang ditulis terakhir ini adalah persamaan untuk mendapatkan kemudahan dalam menentukan nilai karakteristik dan vektor karakteristik dengan menggunakan *Metode Cyclic Jacobi*.

2.3.4 Metode Cyclic Jacobi untuk Matriks Simetris

Setelah kita mendapatkan gambaran tentang aljabar linier yang erat kaitannya dengan matriks, maka sekarang ini kita akan membahas metode Cyclic Jacobi.

Matriks rotasi bidang R_{pq} , yang mana elemen-elemen matriksnya adalah :

$$\begin{aligned} r_{pp} &= \cos \theta & r_{pq} &= \sin \theta \\ r_{qp} &= \sin \theta & r_{qq} &= -\cos \theta \end{aligned}$$

$$R_{pq} = \begin{bmatrix} 1 & & & & & & & & \\ & \Lambda & & & & & & & \\ & & \cos \theta & \Lambda & \sin \theta & & & & \\ & & & \Lambda & & & & & \\ & & \sin \theta & \Lambda & -\cos \theta & & & & \\ & & & & & & \Lambda & & \\ & & & & & & & & 1 \end{bmatrix}$$

Semua elemen-elemen matriks yang tidak dispesifikasikan adalah 0 dan semua entris diagonal adalah 1, sin, atau cos. Sudut rotasi tidak masuk dalam pembicaraan kita. Matriks ini digunakan untuk mentransformasikan matriks A satu iterasi pada satu saat. Misalkan A matriks simetris dan kenyataannya bahwa matriks di atas berpengaruh hanya pada baris dan kolom p dan q.

Pada subbab sebelumnya telah dijelaskan bahwa terdapat ketentuan nilai karakteristik dari matriks diagonal B adalah nilai karakteristik dari matriks simetri A.

Dengan menggunakan rumus pada metode cyclic Jacobi yang bersesuaian dengan ketentuan similaritas di atas, maka diperoleh persamaan

$$D = R_{pq}^T A R_{pq}$$

dimana R adalah perkalian semua matriks rotasi R_{pq} dan R matriks ortogonal, sehingga dari persamaan di atas, dapat ditulis kembali sebagai :

$$AR = RD$$

dimana kolom R memberikan vektor karakteristik dari A yang bersesuaian dengan nilai karakteristiknya

2.4 METODE FISHERFACE

Metode Fisherface adalah metode untuk mentransformasikan vektor citra dari ruang citra dimensi-n ke ruang ciri dimensi-m[BEL-97]. Dengan demikian jika pada tugas akhir ini digunakan citra wajah dengan ukuran 92x112 piksel maka dimensi ruang citra sebesar 10304 piksel. Dimensi citra ini akan direduksi dengan Fisherface hingga memiliki dimensi sebesar m, dengan $m < n$.

Salah satu metode yang digunakan di sini adalah metode Principal Component Analys (PCA), yang juga diketahui sebagai transformasi Karhunen Loeve. Metode PCA ini bertujuan untuk mengurangi dimensi dari sebuah ruang sehingga menghasilkan basis baru yang lebih baik dalam mendeskripsikan berbagai kumpulan "model".

Sedangkan tujuan dari metode Fisherface adalah mereduksi dimensi sekaligus memperbesar rasio jarak antar kelas (between-class scatter) dengan jarak intra kelas (within-class scatter) dari vektor ciri dengan anggapan bahwa semakin besar rasio, vektor ciri yang dihasilkan semakin tidak sensitif baik terhadap perubahan ekspresi maupun perubahan cahaya[BEL-97].

Metode Fisherface dikembangkan untuk citra dalam berbagai variasi cahaya dan ekspresi wajah. Dasar metode Fisherface ini adalah Fisher's Linear Discriminant (FLD). Metode ini ditemukan oleh Robert Fisher pada tahun 1936 untuk klasifikasi taksonomi dan menjadi salah satu teknik yang banyak digunakan dalam pengenalan pola (pattern recognition)[BEL-97].

FLD merupakan salah satu contoh metode class specific, karena metode ini berusaha untuk membentuk jarak (scatter) antar kelas dan intra kelas sehingga dapat menghasilkan klasifikasi yang lebih baik. FLD ini membuat matriks transformasi W dapat memaksimalkan rasio antara determinan between-class scatter (S_B) dengan within-class scatter (S_W) dari vektor-vektor ciri melalui fungsi:

$$W_{\text{opt}} = \arg \max_w \frac{|WS_B W^T|}{|WS_W W^T|}$$

$$= [w_1; w_2; \dots; w_m]$$

dimana $[w_1; w_2; \dots; w_m]$ merupakan m buah vektor eigen (dalam bentuk vektor baris) dari rasio antara S_B dengan S_W , yang bersesuaian dengan m buah nilai eigen terbesar. Jika w_i adalah vektor eigen dari rasio antara matriks S_B dengan matriks S_W dan d_i merupakan nilai eigen yang bersesuaian, maka :

$$S_B w_i^T = d_i S_W w_i^T$$

dimana $i : 1 \dots m$ dan $d_1 > d_2 > \dots > d_m$.

Jika $x_i, i = 1 \dots N$, adalah vektor citra dimensi- n dan masing-masing vektor citra merupakan anggota salah satu dari C kelas citra wajah $\{X_1, X_2, \dots, X_C\}$ dan vektor μ adalah rata-rata vektor citra yang dapat diperoleh dari persamaan

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

maka matriks S_B dan matriks S_W dapat diperoleh melalui persamaan berikut :

$$S_B = \sum_{i=1}^C N_i (\mu_i - \mu)^T (\mu_i - \mu)$$

$$S_W = \sum_{i=1}^C \sum_{j=1, x_j \in X_i}^{N_i} (x_j - \mu_i)^T (x_j - \mu_i)$$

dimana N_i adalah jumlah anggota kelas X_i dan μ_i adalah rata-rata citra anggota kelas $X_i, i = 1 \dots C$.

Suatu citra wajah dengan lebar dan tinggi masing-masing l dan t piksel memiliki jumlah piksel sebanyak $l \times t$. Tiap-tiap piksel dikodekan dengan nilai 0-255 sesuai dengan nilai tingkat keabuannya. Maka dapat dibentuk vektor citra wajah berdasarkan nilai keabuan tersebut, yaitu :

$$g_i = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1l} \\ a_{21} & a_{22} & \dots & a_{2l} \\ \dots & \dots & \dots & \dots \\ a_{t1} & a_{t2} & \dots & a_{tl} \end{bmatrix}$$

dimana $i = 1 \dots N$ (banyaknya citra wajah)

Setiap g_i adalah anggota salah satu kelas wajah X . Jika terdapat C buah kelas wajah X maka terdapat X_j , dimana $j = 1 \dots C$. Untuk setiap kelas wajah X_j terdapat N_j buah citra wajah, dimana $j = 1 \dots C$. Dengan demikian jumlah semua citra wajah adalah $N_1 + N_2 + \dots + N_C = N$.

Dari vektor citra wajah di atas dapat dibentuk suatu vektor baris citra wajah, yaitu: $x_i = [a_{i1} \ a_{i2} \ \dots \ a_{in}]$ $(1 * n)$

Dengan demikian vektor citra dikatakan berada dalam ruang citra dimensi- n , dimana $i = 1 \dots N$. Selanjutnya adalah membentuk matriks input berdimensi $N * n$ yang berisi kumpulan vektor baris citra yang akan digunakan dalam pelatihan dan pengujian.

$$\text{input} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_N \end{bmatrix} \quad (N * n)$$

2.4.1 Algoritma Metode Fisherface

Pada dasarnya metode Fisherface terdiri dari 4 langkah utama, yaitu Metode Principal Component Analys (PCA), Transformasi Principal Component Analys (PCA), Metode Fisher's Linear Discriminant (FLD), dan Transformasi Fisher's Linear Discriminant (FLD). Detail langkah dapat dijelaskan sebagai berikut :

Metode PCA

1. Menormalisasi matriks dari wajah-wajah pelatihan

$$X_i = \frac{X_i}{\sqrt{\sum_{i=1}^n X_i^2}}$$

Dimana n adalah dimensi dari ruang citra.

2. Mencari rata-rata vektor citra

$$\mu_{PCA} = \frac{1}{N} \sum_{i=1}^N X_i \quad (1*n)$$

Dimana N adalah banyaknya data wajah yang akan mengalami pelatihan dan n adalah dimensi dari ruang citra.

3. Untuk $i = 1 \dots N$, mencari selisih vektor citra dengan rata-rata vektor citra,

μ_{PCA}

$$\Phi_i = X_i - \mu_{PCA} \quad (1*n)$$

Menyimpan Φ dalam matriks A, sehingga

$$A = [\Phi_1; \Phi_2; \dots; \Phi_N] \quad (N*n)$$

4. Mencari matriks total scatter, S_T

$$S_T = A A^T \quad (N*n)*(n*N) = (N*N)$$

5. Mencari vektor eigen dan nilai eigen dari matriks S_T , kemudian mengurutkan berdasarkan nilai eigen, dari yang terbesar.

$$[v,d]=\text{eig}(S_T)$$

$$\text{Nilai eigen dari } S_T : d=[d_1 \ d_2 \ \dots \ d_n] \quad (1*N)$$

Dimana $d_1 > d_2 > \dots > d_n$

Vektor eigen yang bersesuaian dengan nilai eigen (dalam bentuk vektor kolom):

$$v = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & w_{nn} \end{bmatrix} \quad (N \times N)$$

6. Mereduksi vektor eigen dari S_T sesuai dengan nilai (N-C)

$$v' = X_i^T V D$$

$$(n \times N) \times (N \times N) \times (N \times (N-C)) = (n \times (N-C))$$

dimana :

- X_i adalah matriks wajah yang telah dinormalisasi dengan ordo $(N \times n)$
- V adalah matriks yang berisi vektor karakteristik dengan ordo $(N \times N)$
- D adalah matriks diagonal yang elemen diagonalnya adalah $(\lambda_i)^{-1/2}$ dengan ordo $(N \times (N-C))$

Matriks transformasi W_{PCA} adalah :

$$W_{PCA} = v'^T \quad ((N-C) \times n)$$

Transformasi PCA

7. Transformasi vektor citra dimensi-n menjadi vektor input dimensi-(N-C), untuk input masukan metode FLD

$$\text{InputFLD} = A W_{\text{PCA}}^T \quad (N \times n) \times (n \times (N-C)) = (N \times (N-C))$$

Atau untuk $i = 1 \dots N$:

$$X'_i = \Phi_i W_{\text{PCA}}^T \quad (1 \times n) \times (n \times (N-C)) = (1 \times (N-C))$$

Jadi :

$$\text{input} = \begin{bmatrix} x'_1 \\ x'_2 \\ \dots \\ x'_N \end{bmatrix}$$

Metode FLD

8. Mencari rata-rata input FLD

$$\mu_{\text{FLD}} = \frac{1}{N} \sum_{i=1}^N X'_i \quad (1 \times (N-C))$$

9. Untuk $I = 1..C$, mencari rata-rata kelas X_i

$$\mu_i = \frac{1}{N_i} \sum_{i=1}^{N_i} X'_i \quad (1 \times (N-C))$$

dimana N_i adalah jumlah anggota kelas X_i

menyimpan μ_i , $i = 1..N$, dalam matriks μ_k , sehingga :

$$\mu_k = \begin{bmatrix} x'_1 \\ x'_2 \\ \dots \\ x'_C \end{bmatrix} \quad (C \times (N-C))$$

10. Mencari matriks *between-class scatter*, S_B

$$S_B = \sum_{i=1}^C N_i (\mu_i - \mu_{\text{FLD}})^T (\mu_i - \mu_{\text{FLD}})$$

$$((N-C) \times C) \times (C \times (N-C)) = ((N-C) \times (N-C))$$

11. Mencari matriks *within-class scatter*, S_W

$$S_W = \sum_{i=1}^C \sum_{j=1, x_j \in X_i}^{N_i} (x_j - \mu_i)^T (x_j - \mu_i)$$

$$((N-C)*N)*(N*(N-C)) = ((N-C)*(N-C))$$

12. Mencari rasio antara S_B dengan S_W

$$\text{Rasio} = \frac{S_B}{S_W} \quad ((N-C)*(N-C))$$

13. Mencari vektor eigen dan nilai eigen dari matriks rasio, kemudian

mengurutkan berdasarkan nilai eigen, dari yang terbesar.

$$[v, d] = \text{eig}(\text{rasio})$$

$$\text{Nilai eigen dari rasio : } d = [d_1 \quad d_2 \quad \dots \quad d_{(N-C)}] \quad (1*(N-C))$$

$$\text{Dimana } d_1 > d_2 > \dots > d_{(N-C)}$$

Vektor eigen yang bersesuaian dengan nilai eigen (dalam bentuk vektor kolom):

$$v = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1(N-C)} \\ w_{21} & w_{22} & \dots & w_{2(N-C)} \\ \dots & \dots & \dots & \dots \\ w_{(N-C)1} & w_{(N-C)2} & \dots & w_{(N-C)(N-C)} \end{bmatrix} \quad ((N-C)*(N-C))$$

14. Mereduksi vektor eigen dari rasio sesuai dengan nilai $m = (C - 1)$

$$v' = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \dots & \dots & \dots & \dots \\ w_{(N-C)1} & w_{(N-C)2} & \dots & w_{(N-C)m} \end{bmatrix} \quad ((N-C)*m)$$

Matriks transformasi W_{FLD} adalah :

$$W_{FLD} = v'^T \quad (m*(N-C))$$

Transformasi FLD

15. Transformasi vektor citra dimensi-(N-C) menjadi vektor ciri dimensi-m

$$W_{opt} = W_{PCA}^T W_{FLD}^T$$

2.4.2 Matriks Bobot Wajah

Matriks bobot wajah (weight face) adalah matriks yang berisi bobot dari tiap-tiap gambar yang digunakan untuk tahap pengenalan wajah. Matriks bobot wajah terdiri dari matriks bobot wajah pelatihan dan matriks bobot wajah uji coba.

2.4.2.1 Matriks Bobot Wajah Pelatihan

Misalkan matriks bobot wajah pelatihan, dinotasikan sebagai Ω , maka matriks bobot pelatihan untuk wajah pelatihan dari sejumlah n gambar adalah :

$$\Omega = X W_{opt}$$

$$(Nxm) = (Nxn) \times (nxm)$$

dimana :

- Ω adalah matriks bobot wajah pelatihan dengan ordo (Nxm).
- X adalah matriks wajah pelatihan yang telah dinormalisasi dengan ordo (Nxn).
- W_{opt} adalah matriks hasil dari transformasi FLD.

2.4.2.2 Matriks Bobot Wajah Uji Coba

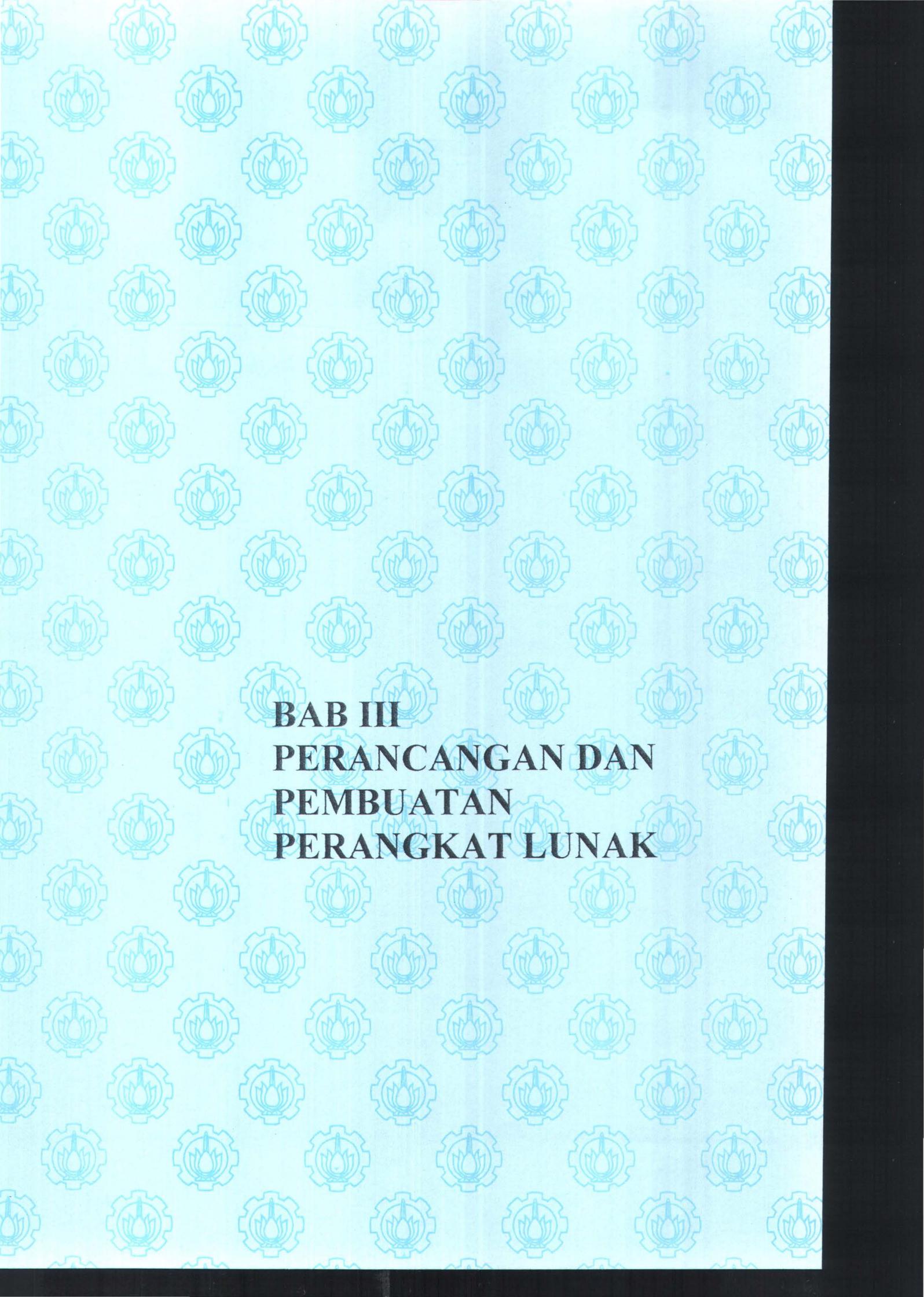
Misalkan sebuah wajah dalam tahap uji coba (testing) akan dihitung bobotnya, maka matriks bobot wajah uji coba, dinotasikan sebagai Ω_t , dengan persamaan sebagai berikut :

$$\Omega_t = X_t W_{opt}$$

$$(1 \times m) = (1 \times n) \times (n \times m)$$

dimana :

- Ω_t adalah matriks bobot wajah ujicoba dengan ordo (1xm).
- X_t adalah matriks dari wajah uji coba yang telah dinormalisasi dengan ordo (1xn).
- W_{opt} adalah matriks hasil dari transformasi FLD.



BAB III
PERANCANGAN DAN
PEMBUATAN
PERANGKAT LUNAK

BAB III

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK

Dalam bab ini dibahas tentang perancangan input/output sistem, perancangan diagram alir data, perancangan struktur data, dan perancangan proses.

3.1 TUJUAN DAN SASARAN SISTEM

Sistem pengenalan wajah ini dibuat dengan tujuan untuk mengimplementasikan metode Fisherface untuk pengenalan pola wajah kedalam bentuk perangkat lunak. Sasaran yang ingin dicapai adalah menguji kemampuan metode tersebut untuk mengenali sejumlah pola wajah yang bervariasi. Untuk memenuhi tujuan dan sasaran sistem diatas, maka ditempuh langkah-langkah seperti yang akan dijelaskan pada sejumlah sub bab dibawah.

3.2 PERANCANGAN PERANGKAT LUNAK

Perancangan perangkat lunak yang dilakukan meliputi perancangan data, perancangan proses, dan perancangan antarmuka. Perancangan data terdiri dari data masukan, data proses, dan data keluaran. Perancangan proses meliputi dari perancangan diagram alir data (DAD). Perancangan antarmuka terdiri form masukan, form proses, form keluaran.

3.2.1 Perancangan Data

Perancangan data bertujuan untuk mengetahui kebutuhan perangkat lunak yang akan dibangun. Pada bagian ini ditentukan data-data yang akan terlibat langsung dalam sistem yang akan dibangun. Perancangan data dapat digolongkan dalam tiga bagian, diantaranya adalah :

3.2.1.1 Data Masukan

Data Masukan dibagi menjadi dua bagian, yaitu : data masukan untuk proses pelatihan dan data masukan untuk proses uji coba. Data masukan untuk proses pelatihan berupa kelompok foto yang ukurannya sudah dijelaskan dalam bab I. Kelompok foto direpresentasikan dalam suatu tabel dengan format no record gambar, gambar. Sedangkan data masukan untuk proses uji coba berupa foto yang tidak terdapat dalam data pelatihan.

3.2.1.2 Data Proses

Data proses adalah data yang digunakan oleh sistem selama proses berlangsung. Data-data ini dapat dibagi lagi menjadi beberapa bagian, yaitu : data proses pelatihan dan data proses uji coba. Proses uji coba baru bisa dilakukan setelah proses pelatihan selesai, karena hasil dari proses pelatihan akan digunakan dalam proses uji coba.

□ *Data Proses Pelatihan, terdiri dari :*

□ *Data Hasil Transformasi PCA*

□ *Data Hasil Normalisasi*

Data ini diperoleh dari normalisasi matriks pelatihan. Kemudian dipakai untuk menghitung rata-rata vektor citra.

□ *Data Hasil Perhitungan Matriks Kovarian*

Dari hasil rata-rata vektor citra, kemudian dicari selisih vektor citra dengan rata-rata vektor citra. Selanjutnya dijadikan matriks kovarian (S_T).

□ *Data Hasil Perhitungan Vektor Eigen dan Nilai Eigen dari Matriks Kovarian (S_T)*

Vektor eigen dan nilai eigen diperoleh dengan menggunakan algoritma Jacobi. Kemudian mengurutkannya berdasarkan nilai eigen, dari yang terbesar. Selanjutnya vektor eigen direduksi sesuai dengan nilai pengurangan antara banyaknya foto dengan banyaknya kelas. Perkalian antara matriks wajah pelatihan yang dinormalisasi dengan transpose vektor eigen yang telah direduksi nantinya dipakai sebagai inputan untuk transformasi FLD(W_{PCA}).

□ *Data Hasil Transformasi FLD*

□ *Data Hasil Rata-rata Input FLD*

Data ini diperoleh dari perhitungan rata-rata input FLD, yang selanjutnya data ini dipakai untuk menghitung matriks between-class scatter (S_B).

□ *Data Hasil Rata-rata Kelas*

Data ini diperoleh dengan menghitung rata-rata untuk masing-masing kelas, yang selanjutnya data ini dipakai untuk menghitung matriks between-class scatter (S_B) maupun matriks within-class scatter (S_W).

□ *Data Hasil Perhitungan Matriks Between-class scatter (S_B)*

Data ini diperoleh dari pengurangan antara data hasil dari rata-rata kelas dengan data hasil dari rata-rata input FLD.

□ *Data Hasil Perhitungan Matriks Within-class scatter (S_W)*

Data ini diperoleh dari pengurangan antara data hasil dari metode PCA dengan data hasil dari rata-rata kelas.

□ *Data Hasil Perhitungan Vektor Eigen dan Nilai Eigen dari Rasio antara S_B dengan S_W*

Mencari vektor eigen dan nilai eigen dari perbandingan antara S_B dengan S_W dengan menggunakan algoritma Jacobi. Kemudian mengurutkannya berdasarkan nilai eigen, dari yang terbesar. Selanjutnya vektor eigen direduksi sesuai dengan nilai banyaknya kelas dikurangi 1. Hasil tersebut selanjutnya disebut (W_{FLD}). Perkalian antara transpose data hasil dari transformasi PCA (W_{PCA}) dengan transpose data hasil dari transformasi FLD (W_{FLD}) atau yang disebut W_{opt} ini yang nantinya akan digunakan untuk menghitung matriks bobot wajah pelatihan maupun matriks bobot wajah uji coba.

□ *Data Proses Uji Coba, terdiri dari :*

□ *Data Hasil Perhitungan Matriks Bobot Wajah Uji Coba*

Data ini didapat dari proses perkalian antara matriks wajah uji coba yang telah dinormalisasi dengan data dari proses pelatihan (W_{opt}). Data ini selanjutnya dibandingkan dengan matriks bobot pelatihan.

□ *Data Hasil Perhitungan Jarak Euclidean*

Data ini diperoleh dengan menghitung perbandingan bobot wajah uji coba dengan bobot wajah pelatihan. Rumus dari jarak Euclidean adalah $|x_{1i} - x_{2i}| = (\sum (x_{1i} - x_{2i})^2)^{1/2}$.

3.2.1.3 Data Keluaran

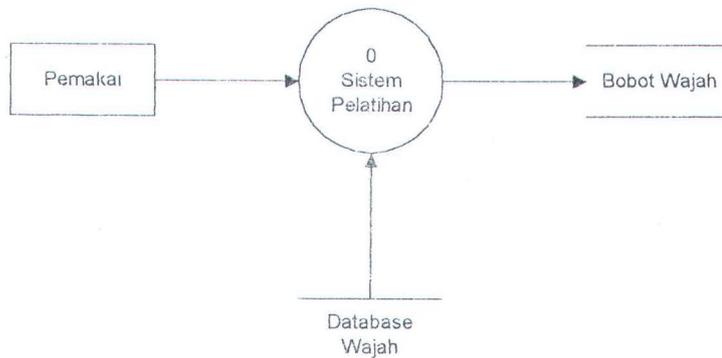
Data keluaran diperoleh dari pengurutan data hasil perhitungan jarak euclidean. Dari jarak terpendek akhirnya didapat foto dari data pelatihan yang mirip dengan foto dari data uji coba yang dimasukkan.

3.2.2 Perancangan Proses

Sistem pengenalan wajah ini terdiri dari dua subsistem yaitu subsistem proses pelatihan wajah-wajah yang ada didalam basis data dan subsistem proses uji coba atau pengenalan. Pembagian sistem pengenalan wajah ini menjadi dua subsistem didasarkan pada kondisi bahwa proses uji coba hanya bisa dilakukan setelah subsistem pelatihan dijalankan.

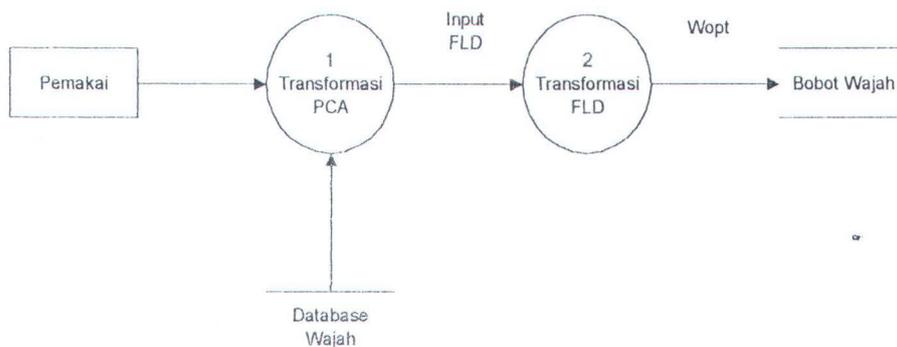
Bagian ini juga akan membahas perancangan sistem pengenalan wajah yang akan direpresentasikan dalam gambar, yaitu menggunakan Diagram Aliran Data (DAD) atau Data Flow Diagram (DFD). Dengan menggunakan diagram aliran data, dapat memperjelas semua proses dan aliran data yang terjadi dalam sistem pengenalan wajah ini, dari diagram aliran data juga dapat diketahui data yang diolah dan dihasilkan oleh suatu proses, dengan demikian aliran data mulai awal proses sampai akhir proses dapat diketahui dengan jelas.

Subsistem pelatihan merupakan bagian dari sistem pengenalan wajah dengan menggunakan metode Fisherface, subsistem pelatihan ini harus dilakukan dahulu sebelum subsistem uji coba atau pengenalan wajah dilakukan. Secara keseluruhan subsistem pelatihan dapat dilihat pada gambar 3.1 dibawah ini, yang menggambarkan kontek diagram atau DFD level 0 dari subsistem pelatihan



Gambar 3.1 DFD level 0 dari subsistem pelatihan

Data flow diagram diatas menggambarkan sistem pelatihan yang mengambil beberapa input citra dari database wajah kedalam proses pelatihan dan menghasilkan bobot wajah dari masing-masing wajah dari database wajah yang digunakan. Selanjutnya proses pelatihan ini dipecah-pecah menjadi beberapa subproses seperti gambar 3.2 yang menggambarkan DFD level 1.

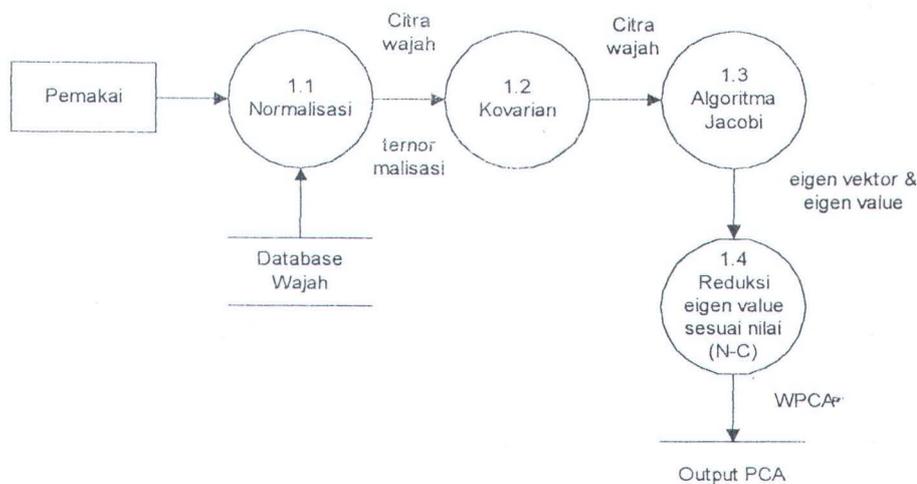


Gambar 3.2 DFD level 1 dari subsistem pelatihan

Proses pelatihan terdiri dari transformasi PCA dan transformasi FLD. Hasil atau output dari transformasi PCA tersebut menjadi inputan untuk melakukan transformasi FLD. Dalam transformasi PCA dipecah-pecah menjadi beberapa proses seperti gambar 3.3.

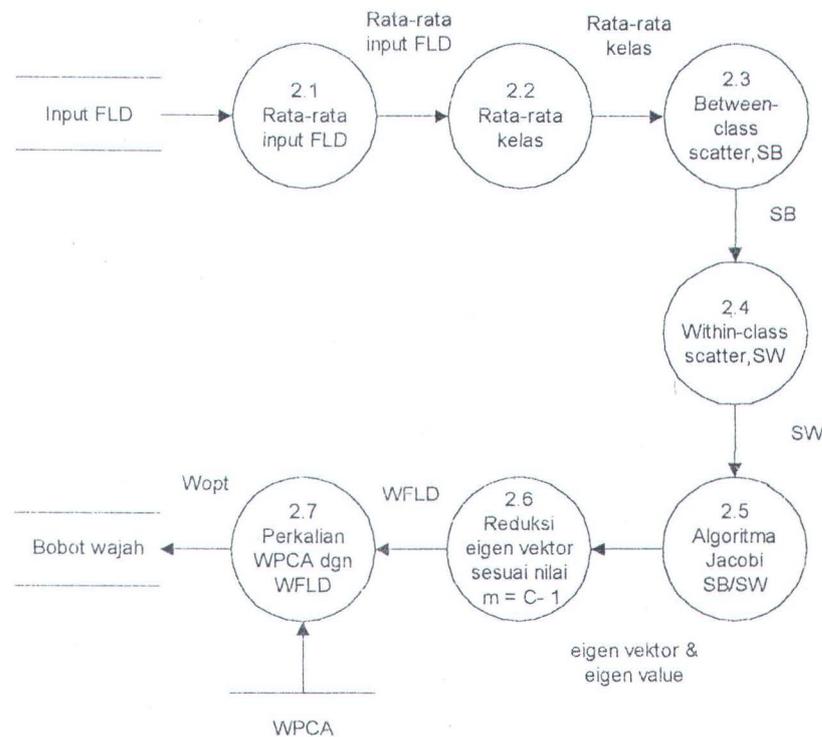
Proses dalam tranformasi PCA terdiri dari proses-proses sebagai berikut :

- Menormalisasi wajah-wajah pelatihan.
- Menghitung matriks kovariansi dengan menggunakan *outer product*.
- Menghitung vektor karakteristik dan nilai karakteristik untuk matriks kovariansi.
- Mengurutkan vektor karakteristik yang bersesuaian dengan nilai karakteristik dengan urutan *descending(naik)*.
- Mereduksi vektor karakteristik dengan nilai $N-C(W_{PCA})$, dimana N adalah banyak wajah atau data dalam basis data wajah, C merupakan jumlah kelas dalam basis data tersebut.



Gambar 3.3 DFD level 2 dari subsistem pelatihan Metode PCA

Setelah proses transformasi PCA selesai dijalankan hasil dari proses tersebut menjadi inputan untuk proses FLD. Output atau hasil dari transformasi FLD tersebut yang digunakan untuk menghitung bobot wajah-wajah pelatihan dan bobot wajah-wajah uji coba. Secara lebih lengkap proses transformasi FLD dapat dilihat pada gambar 3.4.

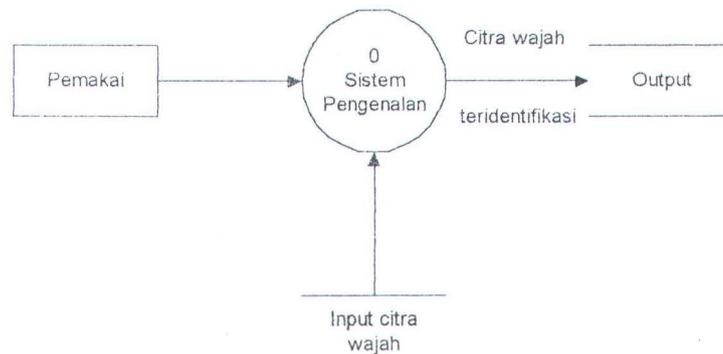


Gambar 3.4 DFD level 2 dari subsistem pelatihan Metode FLD

Proses dalam transformasi FLD terdiri dari proses-proses sebagai berikut :

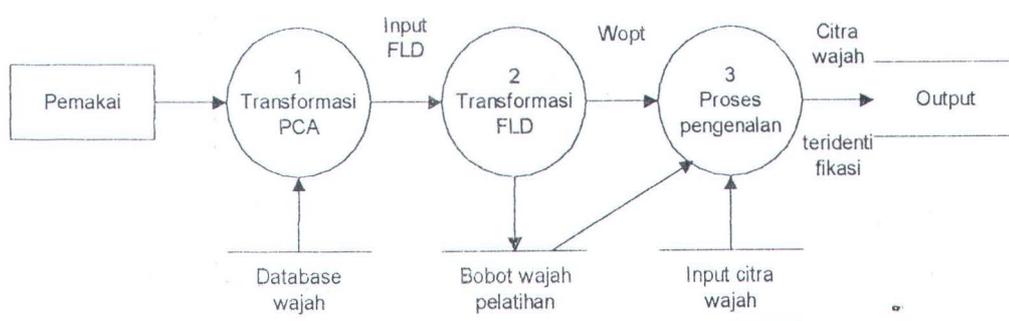
- Mencari rata-rata inputFLD.
- Mencari rata-rata kelas dari inputFLD.
- Mencari matriks *between-class scatter*, S_B .
- Mencari matriks *within-class scatter*, S_W .
- Mencari rasio antara S_B dengan S_W

- Menghitung vektor karakteristik dan nilai karakteristik untuk matriks rasio.
- Mengurutkan vektor karakteristik yang bersesuaian dengan nilai karakteristik dengan urutan *descending*.
- Mereduksi vektor karakteristik dengan nilai $C-1$. (W_{FLD}).



Gambar 3.5 DFD level 0 dari subsistem pengenalan wajah

Gambaran dari subsistem pengenalan wajah secara keseluruhan dapat digambarkan dengan Diagram Aliran Data level 0 (Data Flow Diagram). Dari gambar DFD level 0 dapat dilihat cara kerja sistem pengenalan wajah secara umum menerima suatu input citra yang akan dikenali, dan kemudian menghasilkan output suatu citra yang teridentifikasi.



Gambar 3.6 DFD level 1 dari subsistem pengenalan wajah

Seperti dijelaskan dalam dekripsi sistem diatas bahwa proses pengenalan atau uji coba baru bisa dijalankan setelah proses training selesai dijalankan. Secara lebih lengkap proses pengenalan atau uji coba dapat dilihat pada gambar 3.6.

Proses dalam pengenalan atau uji coba terdiri dari proses-proses sebagai berikut :

- Mengambil sebuah wajah uji coba.
- Menormalisasi wajah uji coba tersebut.
- Menghitung bobot wajah uji coba.
- Membandingkan bobot wajah uji coba dengan bobot wajah pelatihan dengan menggunakan rumus jarak Euclidean.
- Mengurutkan jarak yang dihasilkan dengan urutan ascending, kemudian mengambil jarak minimum terdekat.

3.2.3 Perancangan Antarmuka

Pada bagian ini akan dijelaskan struktur rancangan antarmuka dari sistem. Adapun hirarki menunya adalah sebagai berikut :

□ Menu Open Database

Menu ini digunakan untuk membuka tabel atau database yang berisi wajah-wajah pelatihan yang akan diproses.

□ Menu Inisialisasi

Menu ini digunakan untuk membaca foto kedalam bentuk matriks atau angka.

- Menu Training

Menu ini digunakan untuk melakukan proses pelatihan dari wajah-wajah pelatihan yang diinputkan tersebut.

- Menu Open File

Pada menu ini wajah-wajah uji coba dapat dimasukkan.

- Menu Testing

Menu ini akan melakukan proses pengenalan dari wajah-wajah uji coba yang dimasukkan.

3.3 PEMBUATAN PERANGKAT LUNAK

Subbab ini dibagi menjadi tiga bagian, yaitu implementasi data, implementasi proses, dan implementasi antarmuka.

3.3.1 Implementasi Data

Data berupa foto yang kemudian dirubah dalam bentuk matriks atau angka. Disini bisa menambah kelas dan menghapus kelas. Setiap kelas harus mempunyai jumlah data yang sama.

Prosedur dibawah ini digunakan untuk menambah kelas.

```
procedure TFMain.btnAddClick(Sender: TObject);
var Pic: TPicture;
begin
  if dlgOpenImg.Execute then
  try
    Pic := TPicture.Create;
    Pic.LoadFromFile(dlgOpenImg.FileName);
    tblData.Append;
    dbImg.Picture := Pic;
    tblData.Post;
    AddData(Pic);
    LoadData;
```

```

        ShowData;
    finally
        Pic.Free;
    end;
end;

procedure TFMMain.AddData(Pic: TPicture);
var len: integer;
begin
    len := Length(ImgData);
    SetLength(ImgData, len+1);
    ImgData[len] := TImage.Create(Self);
    ImgData[len].Picture := Pic;
    ShowData;
end;

```

Prosedur dibawah ini digunakan untuk menghapus kelas.

```

procedure TFMMain.btnDeleteClick(Sender: TObject);
var idx: integer;
begin
    tblData.Delete;
    LoadData;
    ShowData;
end;

procedure TFMMain.LoadData;
var JmlRec, i: integer;
begin
    FreeData;
    SetLength(ImgData, 0);
    scbData.Refresh;
    With tblData do
    begin
        JmlRec := RecordCount;
        SetLength(ImgData, JmlRec);
        i := 0;
        tblData.First;
        While not tblData.Eof do
        begin
            ImgData[i] := TImage.Create(Self);
            ImgData[i].Picture := dbImg.Picture;
            Inc(i);
            Next;
        end;
    end;
end;

```

Prosedur untuk menampilkan foto-foto yang digunakan sebagai masukan dari sistem

```

procedure TFMMain.ShowData();
var Len, i, a, b: integer;
begin
    With tblData do
    begin
        a := 0; b := 0;

```

```

Len := Length(ImgData);
for i:=0 to Len-1 do
begin
  ImgData[i].Parent := scbData;
  ImgData[i].Width := 92;
  ImgData[i].Height := 112;
  ImgData[i].Left := 5 + (i*100);
  if a > Ni-1 then
  begin
    a := 0;
    Inc(b);
  end;
  ImgData[i].Top := 5 + (b*120);
  ImgData[i].Left := a * 100;
  Inc(a);
  Next;
end;
end;
stBar.SimpleText := ' Jumlah Item :' +
IntToStr(Length(ImgData));
end;

```

Prosedur untuk membaca foto ke dalam bentuk matriks atau angka

```

procedure TFMMain.Inisialisasi;
var i,j,k :integer;
begin
  setlength(imgdata,tbldata.recordcount);
  for k:=0 to tbldata.recordcount-1 do
  begin
    for i:=1 to 112 do
      for j:=1 to 92 do
        D[k+1,(i-1)*92+j]:=
          getrvalue(imgdata[k].picture.bitmap.
            canvas.pixels[i-1,j-1]);
      end;
    end;
  end;
end;

```

3.3.2 Implementasi Proses

Dalam proses terdapat prosedur jacobi, prosedur normalisasi matriks, prosedur untuk mencari rata-rata kelas, prosedur mencari between-class scatter, prosedur mencari within-class scatter, prosedur membandingkan bobot wajah pelatihan dengan bobot wajah uji coba.

Prosedur Pencarian Matriks Eigen dengan Algoritma Jacobi.

Prosedur ini digunakan untuk mencari vektor eigen dan nilai eigen suatu matriks,

baik untuk matriks kovarian maupun matriks rasio antara between-class scatter

dengan within-class scatter

```

procedure jacobi(var a: glnpnp;n:integer;var d:glnp; var v : glnpnp; var
nrot : integer);
label 99;
const
  nmax = 200;
var
  j,iq,ip,i:integer;
  tresh,theta,tau,t,sm,s,h,g,c : real;
  b,z : array [1..nmax] of real;
begin
  for ip := 1 to n do
    begin
      for iq := 1 to n do
        begin
          v[ip,iq] := trunc(0.0)
        end;
      v[ip,ip] := trunc(1.0)
    end;
  for ip := 1 to n do
    begin
      b[ip] := a[ip,ip];
      d[ip] := b[ip];
      z[ip] := trunc(0.0)
    end;
  nrot := 0;
  for i:= 1 to 50 do
    begin
      sm := trunc(0.0);
      for ip := 1 to n-1 do
        begin
          for iq := ip+1 to n do
            begin
              sm := sm + abs(a[ip,iq])
            end;
          end;
      if (sm = trunc(0.0)) then goto 99;
      if (i < 4) then tresh := 0.2*sm/sqr(n)
      else
        tresh := trunc(0.0);
      for ip := 1 to n-1 do
        begin
          for iq := ip + 1 to n do
            begin
              g := trunc(100.0)* abs(a[ip,iq]);
              if ((i>4) and ((abs(d[ip]+g))=
                abs(d[ip])) and ((abs(d[iq]+g)) =
                abs(d[iq])))) then a[ip,iq] := trunc(0.0)
              else
                if (abs(a[ip,iq])>tresh) then

```

```

begin
  h := d[iq]-d[ip];
  if ((abs(h)+g) = abs(h)) then
    begin
      t:=a[ip,iq]/h
    end
  else
    begin
      theta := 0.5*h/a[ip,iq];
      t:=trunc(1.0)/(abs(theta)+
        sqrt(trunc(1.0)+sqr(theta)));
      if (theta<trunc(0.0)) then t:=-t
    end;
  c:=trunc(1.0)/sqrt(1+sqr(t));
  s:=t*c;
  tau:=s/(trunc(1.0)+c);
  h:=t*a[ip,iq];
  z[ip]:=z[ip]-h;
  z[iq]:=z[iq]+h;
  d[ip]:=d[ip]-h;
  d[iq]:=d[iq]+h;
  a[ip,iq]:=trunc(0.0);
  for j:=1 to ip-1 do
    begin
      g:=a[j,ip];
      h:=a[j,iq];
      a[j,ip]:=g-s*(h+g*tau);
      a[j,iq]:=h+s*(g-h*tau)
    end;
  for j := ip+1 to iq - 1 do
    begin
      g:= a[ip,j];
      h:= a[j,iq];
      a[ip,j]:= g-s*(h+g*tau);
      a[j,iq]:= h+s*(g-h*tau)
    end;
  for j:= iq +1 to n do
    begin
      g:= a[ip,j];
      h:= a[iq,j];
      a[ip,j]:= g-s*(h+g*tau);
      a[iq,j]:= h+s*(g-h*tau)
    end;
  for j:= 1 to n do
    begin
      g:= v[j,ip];
      h:= v[j,iq];
      v[j,ip]:= g-s*(h+g*tau);
      v[j,iq]:= h+s*(g-h*tau)
    end;
  nrot := nrot+1
end
end
end
end;
for ip := 1 to n do
begin
  b[ip]:=b[ip]+z[ip];
  d[ip]:=b[ip];
  z[ip]:=trunc(0.0)
end
end
end
end

```

Prosedur normalisasi matriks

Normalisasi matrik menggunakan rumus :
$$\frac{X_i}{\sqrt{\sum_{i=1}^n X_i^2}}$$

Normalisasi untuk matriks wajah pelatihan maupun untuk matriks wajah uji coba.

X_i adalah matriks wajah, n besar dari dimensi matriks wajah.

```
for i:= 1 to N do
  begin
    temp := 0;
    for j:= 1 to Dim do
      temp := temp + D[i,j]*D[i,j];
    temp:= sqrt(temp);
    for j:= 1 to Dim do
      D[i,j]:= D[i,j]/temp;
    end;
  end;
```

{Mencari Rata-rata Kelas menggunakan rumus $\mu_i = \frac{1}{N_i} \sum_{i=1}^{N_i} X'_i$ }

Dalam proses ini mencari rata-rata kelas masing-masing kelas, dimana N_i adalah jumlah wajah atau anggota setiap kelas.

```
for k:= 1 to N-C do
  begin
    for i:= 1 to C do
      begin
        temp3:=0;
        for j:=(i-1)*Ni+1 to (Ni*i) do
          begin
            temp3:=temp3+H[j,k];
          end;
        R[i,k]:=temp3/Ni;
      end;
    end;
  end;
```

{Mencari $S_B = \sum_{i=1}^C N_i (\mu_i - \mu_{FLD})^T (\mu_i - \mu_{FLD})$ }

Mencari matriks jarak antar kelas (between-class scatter), diperoleh dari rata-rata kelas dikurangi dengan rata-rata matriks hasil dari transformasi PCA

```
for i:=1 to N-C do
  begin
    for j:= 1 to C do
      begin
```

```

        F[j,i]:=R[j,i]-E[i];
    end;
end;

```

$$\{\text{Mencari } S_w = \sum_{i=1}^C \sum_{j=1, x_j \in X_i}^{N_i} (x_j - \mu_i)^T (x_j - \mu_i)\}$$

Mencari matriks jarak intra kelas (within-class scetter), diperoleh dari matriks hasil dari transformasi PCA dikurangi dengan matriks rata-rata kelas.

```

for k:=1 to N-C do
    begin
        for i:=1 to C do
            begin
                for j:=(i-1)*Ni+1 to (Ni*i) do
                    begin
                        H[j,k]:=H[j,k]-R[i,k];
                    end;
                end;
            end;
        end;
    end;

```

{Jarak Euclidean menggunakan rumus $|x_{1i} - x_{2i}| = (\sum (x_{1i} - x_{2i})^2)^{1/2}$ }

Proses ini digunakan untuk menghitung perbandingan bobot wajah uji coba dengan wajah pelatihan.

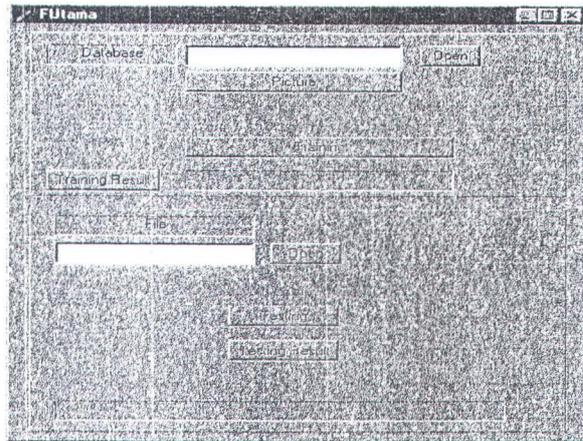
```

for i:=1 to N do
    begin
        templ:=0;
        for j:= 1 to C-1 do
            begin
                templ:=templ+(sqr(L[j]-H[i,j]));
            end;
        E[i]:=sqrt(templ);
        z1[i]:=E[i];
    end;

```

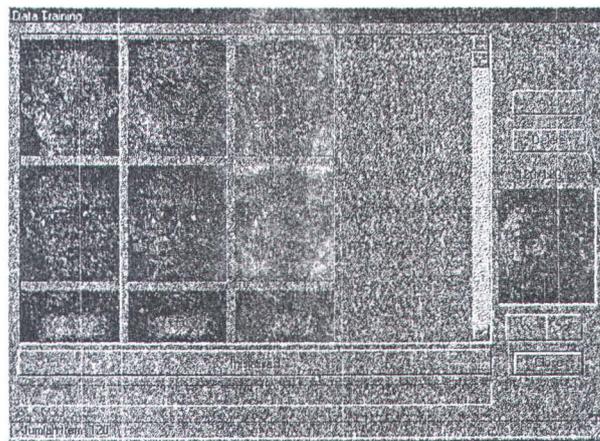
3.3.3 Implementasi Antarmuka

Pada bagian ini akan dijelaskan implementasi rancangan antarmuka seperti yang dijelaskan sebelumnya. Adapun prototype menu utama seperti ditunjukkan oleh gambar 3.7 berikut ini :



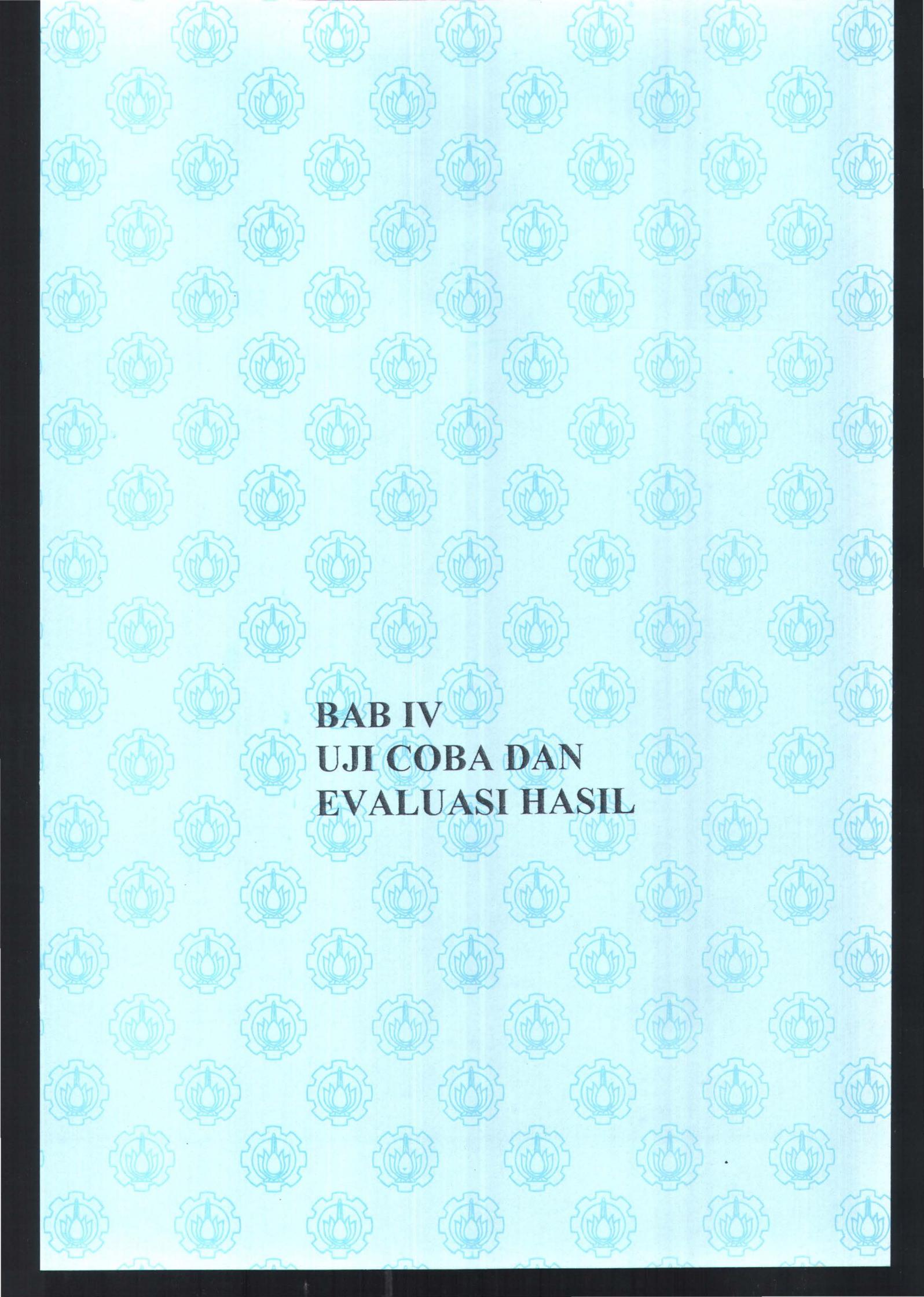
Gambar 3.7 Menu Utama

Setelah mengisi Open Database akan tampil form seperti gambar 3.8



Gambar 3.8 Tampilan sebagian wajah pelatihan

Add dan *Delete* digunakan untuk menambah dan mengurangi kelas pelatihan. Tombol *Inisialisasi* digunakan untuk merubah wajah pelatihan kedalam bentuk matriks. Setelah proses inisialisasi selesai, kembali ke menu utama. Tombol *Training* digunakan untuk melakukan proses pelatihan dari wajah-wajah pelatihan. Open file wajah yang akan diuji coba dilakukan setelah proses training selesai. Wajah yang mirip dengan wajah uji coba akan tampil setelah proses testing selesai.



BAB IV
UJI COBA DAN
EVALUASI HASIL

BAB IV

UJI COBA DAN EVALUASI PERANGKAT LUNAK

Bab ini menjelaskan tentang hasil uji coba dan evaluasi yang dilakukan terhadap perangkat lunak sistem pengenalan wajah dengan menggunakan metode Fosherface. Uji coba yang dilakukan terhadap sistem ini dimaksudkan untuk mengetahui kemampuan dan kelemahan sistem. Hasil pengujian akan dijadikan dasar bagi evaluasi terhadap sistem ini. Hasil evaluasi terhadap sistem ini bisa digunakan untuk pengembangan sistem lebih lanjut, sehingga bisa dijadikan sistem pengenalan yang lebih baik dibandingkan dengan sistem ini. Kriteria yang digunakan untuk evaluasi sistem ini adalah sebagai berikut :

- Waktu yang digunakan selama proses pelatihan berlangsung.
- Keberhasilan sistem dalam mengenali citra wajah dengan mengukur prosentase keberhasilan pengenalan terhadap jumlah citra wajah keseluruhan.

4.1 LINGKUNGAN UJI COBA

Perangkat keras yang dipakai untuk uji coba perangkat lunak ini adalah komputer dengan prosesor Intel Pentium II 450 Mhz dengan memori utama sebesar 128 MB.

4.2 DATA UJI COBA

Uji coba terhadap sistem ini dilakkan dengan menggunakan citra wajah yang berukuran 92 x 112 piksel. Citra wajah tersebut berada dalam bentuk grey level dengan tingkat keabuan 256. Data uji coba diambil dari basis data wajah yang dimiliki oleh laboratorium riset Olivetti. Basis data ini terdiri dari 400 obyek wajah dari 40 orang dengan 10 variasi untuk setiap orang. Data ini diindeks dengan angka 1 sampai 10. Sebagian dari data ini digunakan untuk proses pelatihan dan sisanya sebagai data uji coba.

4.3 UJI COBA DAN EVALUASI HASIL

Subbagian ini menjelaskan tentang uji coba dan evaluasi hasil yang dilakukan terhadap sistem pengenalan wajah ini. Pengujian yang dilakukan terhadap sistem pengenalan wajah ini menggunakan dua tipe data pelatihan, yaitu: data pelatihan terurut dan data pelatihan terkendali.

a. Data pelatihan terurut yang terdiri dari :

Data pelatihan terurut diambil dari data pelatihan sesuai dengan nomor urutan datanya tanpa memandang variasi data yang ada. Data pelatihan terurut terdiri

- Data pelatihan terurut 1

Indeks nomor 1 sampai dengan 3 dari 10 variasi data yang ada digunakan sebagai data pelatihan, sedangkan sisanya untuk pengujian.

- Data pelatihan terurut 2.

Indeks nomor 1 sampai dengan 4 dari 10 variasi data yang ada digunakan sebagai data pelatihan, sedangkan sisanya untuk pengujian

- Data pelatihan terurut 3.

Indeks nomor 1 sampai dengan 5 dari 10 variasi data yang ada digunakan sebagai data pelatihan, sedangkan sisanya untuk pengujian

b. Data pelatihan terkendali.

Data pelatihan ini diambil dengan memilih dan mempertimbangkan variasi data yang ada tanpa memandang urutan indeks data. Data pelatihan terkendali ini diambil secara manual. Data pelatihan terkendali terdiri dari :

- Data pelatihan terkendali 1.

3 data dipilih dari 10 variasi data dengan pertimbangan variasi data yang ada tanpa memandang urutan indeks data. Sedangkan untuk pengujiannya digunakan tujuh data sisanya. Data yang terpilih untuk tipe ini sebagai data pelatihan dan data uji coba bisa dilihat pada lampiran.

- Data pelatihan terkendali 2.

4 data dipilih dari 10 variasi data dengan pertimbangan variasi data yang ada tanpa memandang urutan indeks data. Sedangkan untuk pengujiannya digunakan enam data sisanya. Data yang terpilih untuk tipe ini sebagai data pelatihan dan data uji coba bisa dilihat pada lampiran.

- Data pelatihan terkendali 3.

5 data dipilih dari 10 variasi data dengan pertimbangan variasi data yang ada tanpa memandang urutan indeks data. Sedangkan untuk pengujiannya digunakan lima data sisanya. Data yang terpilih untuk tipe ini sebagai data pelatihan dan data uji coba bisa dilihat pada lampiran.

4.3.4 Perbandingan Dengan Sistem

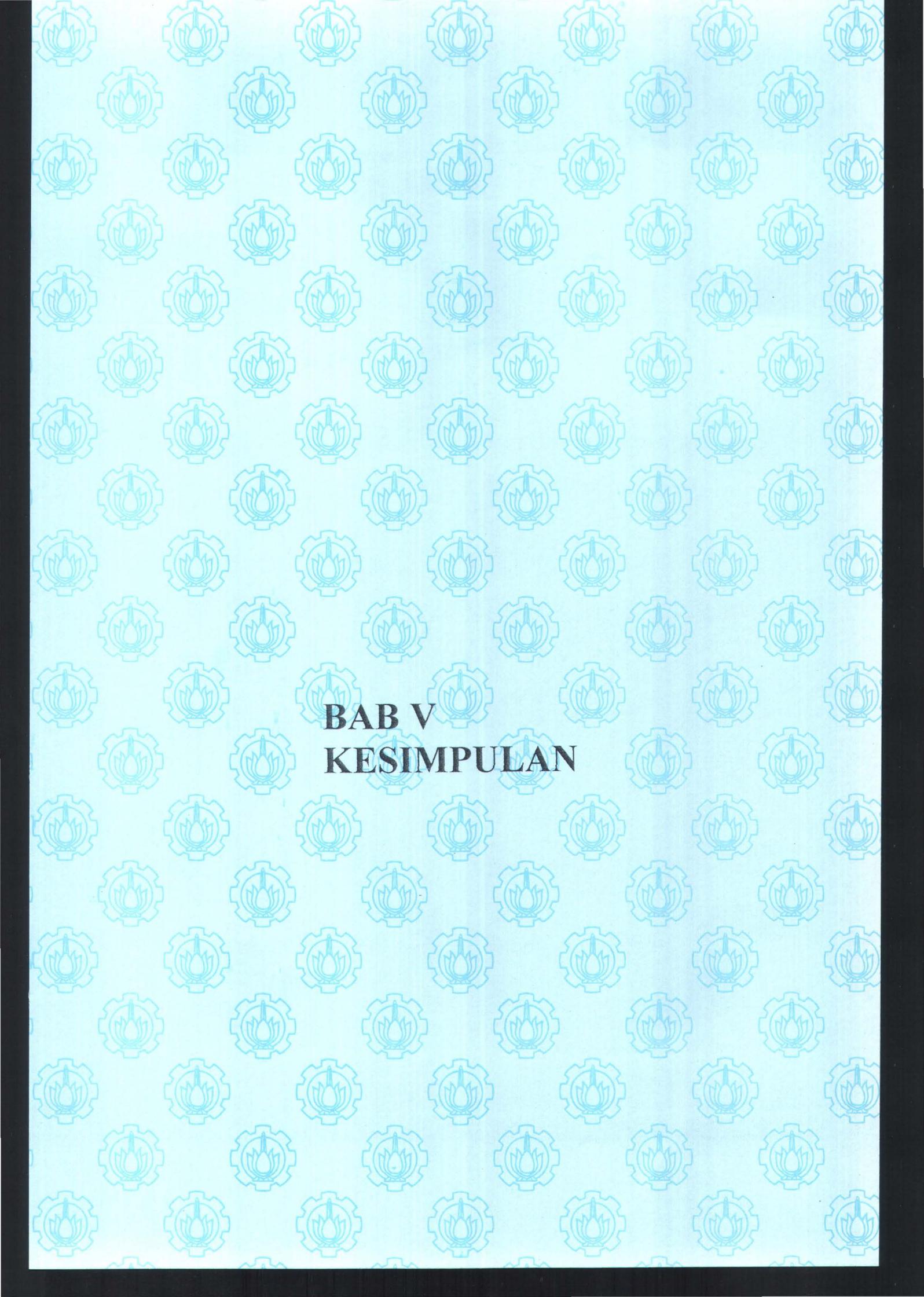
Berdasarkan hasil studi literature terhadap berbagai tugas akhir sarjana di Jurusan Teknik Informatika, terdapat dua tugas akhir mengenai pengenalan wajah yang dapat dijadikan sebagai bahan perbandingan dengan sistem pengenalan wajah yang dibuat dalam tugas akhir ini. Kedua tugas akhir yang relevan tersebut adalah :

"PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK PENGENALAN WAJAH DENGAN MENGGUNAKAN JARINGAN SYARAF YANG DIDASARKAN PADA KEPUTUSAN PROBABILISTIK" [SUB-99]

Kemampuan pengenalan yang dimiliki oleh jaringan syaraf PDBNN ini tergantung dari jumlah variasi data pelatihan yang digunakan untuk pelatihan. Pengambilan data pelatihan tanpa memandang variasi data yang mewakili, seperti yang dilakukan pada uji coba, menghasilkan tingkat keberhasilan 56,5% untuk subnet yang mewakili bernilai tertinggi dan 69,5% untuk subnet yang mewakili bernilai positif. Sedangkan pengambilan data dengan pemilihan yang mewakili variasi dari data yang ada, menghasilkan tingkat keberhasilan 72% untuk subnet yang mewakili bernilai tertinggi dan 85% untuk subnet yang mewakili bernilai positif. Waktu pelatihan untuk jaringan pendeteksi wajah dan pencari lokasi mata 4 menit dan waktu pelatihan jaringan syaraf 2 menit. Waktu rata-rata yang digunakan untuk menemukan area wajah yang memiliki nilai log-likelihood tertinggi 25 detik dan untuk menemukan posisi kedua mata 6 detik.

"PENGENALAN WAJAH DENGAN MENGGUNAKAN JARINGAN SYARAF KONVOLUSIONAL"[FAT-98]

Dari sebanyak 280 data yang diujicobakan, 168 data dapat dikenali sedangkan 112 data lainnya gagal. Prosentase keberhasilan dari uji coba ini sebesar 71,5% untuk data pelatihan terurut dan 82% untuk data pelatihan terkendali. Jumlah data pelatihan yang semakin besar tidak menjamin meningkatnya kemampuan jaringan syaraf konvolusional dalam melakukan klasifikasi. Hal ini mungkin dipengaruhi oleh variasi data pelatihan. Data pelatihan yang tidak terlalu variatif akan menyempitkan jangkauan dari kelas tersebut. Akan tetapi data pelatihan dengan variasi yang tinggi akan mengurangi ketepatan dalam generalisasi kelas tersebut. Waktu yang dibutuhkan untuk data pelatihan terurut untuk pelatihan jaringan SOM 10 menit dan untuk pelatihan jaringan konvolusional 19 jam. Sedangkan untuk data pelatihan terkendali untuk pelatiha jaringan SOM 52 menit dan untuk pelatihan jaringan konvolusional 26 jam.



BAB V
KESIMPULAN

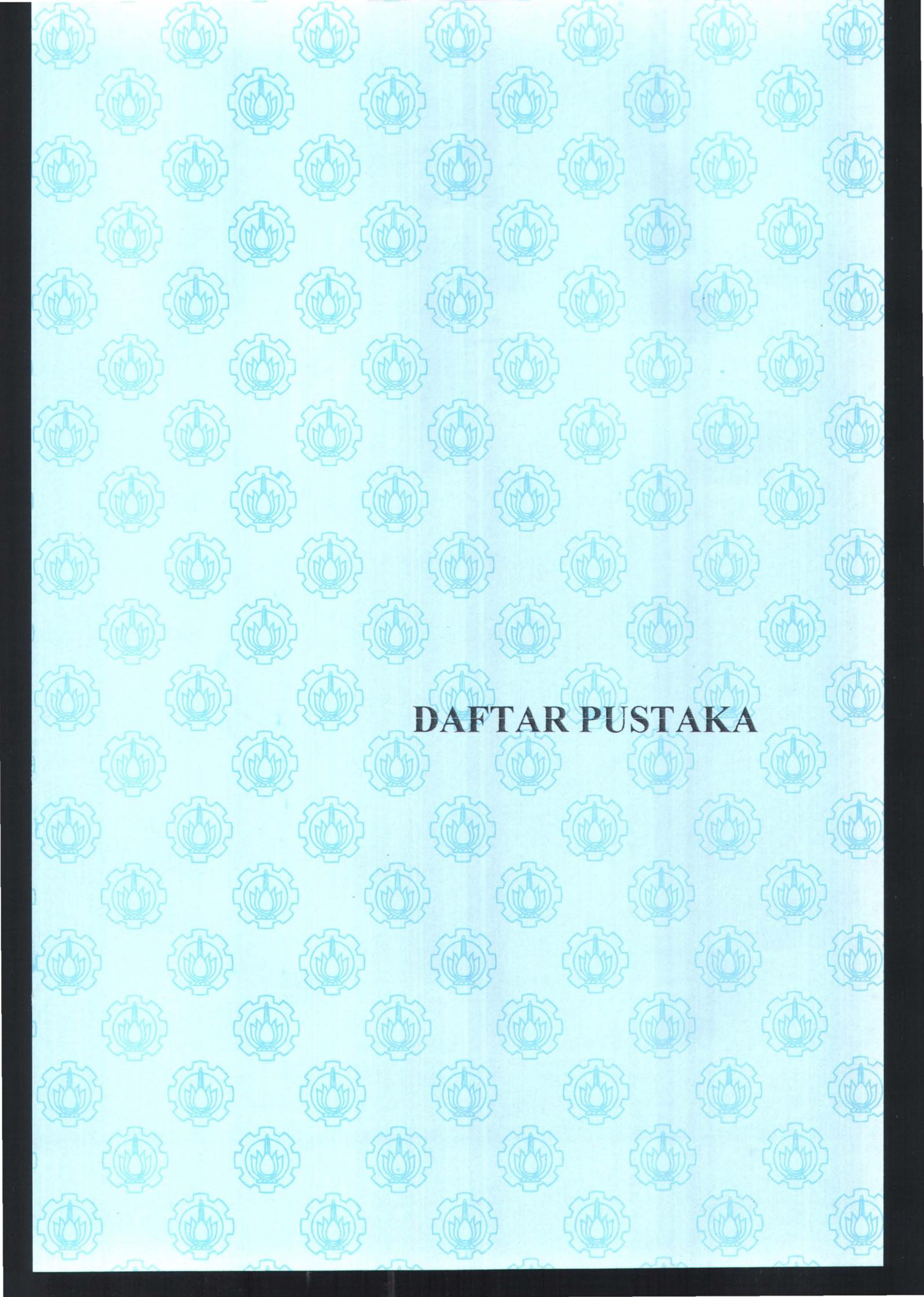
BAB V

KESIMPULAN

Dalam bab ini dibahas mengenai beberapa kesimpulan yang dapat diambil dari keseluruhan tugas akhir yang telah dibuat. Kesimpulan diambil berdasarkan hasil ujicoba yang telah dilakukan terhadap sistem pengenalan wajah dengan data yang tersedia.

Dari ujicoba yang telah dilakukan sistem pengenalan wajah ini dapat diambil beberapa kesimpulan :

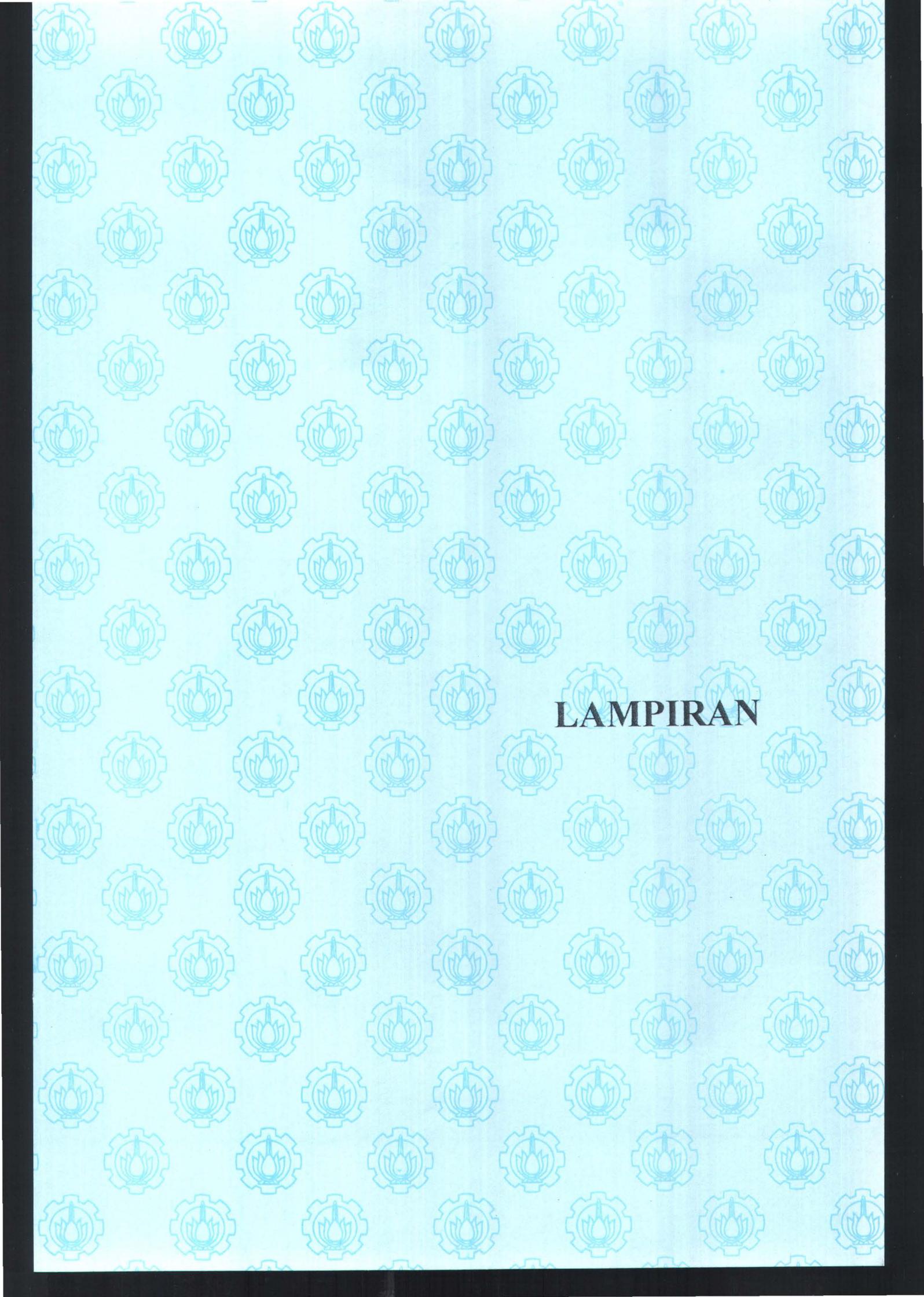
1. Hasil yang diperoleh dalam proses pengenalan wajah dengan menggunakan Metode *Fisherface* untuk data pelatihan terurut 80% sedangkan untuk data pelatihan terkendali 88%. Hasil ini menunjukkan bahwa pengenalan wajah dengan menggunakan Metode *Fisherface* cukup baik untuk diterapkan dalam sistem pengenalan wajah.
2. Hasil perbandingan dengan 2 sistem pengenalan wajah yang lain. Dengan menggunakan jaringan syaraf yang didasarkan pada keputusan probabilistik untuk data pelatihan terurut 69,5% sedangkan untuk data pelatihan terkendali 85%. Dengan menggunakan jaringan syaraf konvolusional untuk data pelatihan terurut 71,5% sedangkan untuk data pelatihan terkendali 82%.
3. Untuk setiap tambahan input wajah baru dilakukan pelatihan ulang sehingga konsekuensinya dibutuhkan memori yang cukup besar.



DAFTAR PUSTAKA

DAFTAR PUSTAKA

- [SUB-99] Agus Subhan , “Perancangan dan Pembuatan Perangkat Lunak Pengenalan Wajah Dengan Menggunakan Jaringan Syaraf yang didasarkan pada Keputusan Probabilistik”, *Tugas Akhir*, Sarjana Jurusan Teknik Informatika FTI-ITS, 1999
- [ANT-98] Howard Anton, *Elementary Linear Algebra*, Drexel University Press, 1998
- [BEL-97] Belhumeur P.N, J.P.Hespanha, and D.J.Kriegman, “Face Recognition Using Class Specific Linear Projection” , *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol 19, No 7, July 1997
- [FAT-98] Aminudin Al Fathhoni, “Pengenalan Wajah Dengan Menggunakan Jaringan Syaraf Konvolusional”, *Tugas Akhir*, Sarjana Jurusan Teknik Informatika FTI-ITS, 1998
- [SET-00] Setyati Endang, “Pengenalan Wajah dengan Menggunakan PCA”, *Tesis Pasca Sarjana*, Jurusan Teknik Informatika, 2000
- [SAM-97] Sami Romdhani, *Face Recognition by Using PCA*, 1997
- [WIL-89] William H, Teukolsky, Saul A, Vetterling, William T, and Flannery Brian P, *Numerical Recipes in C*, Cambridge University Press, 1989

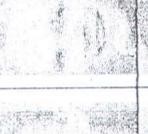
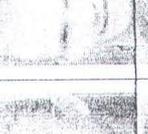
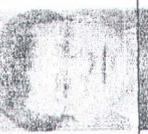
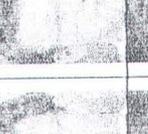
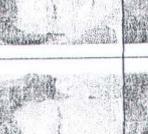


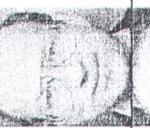
LAMPIRAN

LAMPIRAN A
DATA PELATIHAN

Gambar yang digunakan untuk pelatihan maupun untuk uji coba terdapat pada tabel A.1. Gambar tersebut terdiri dari 40 orang yang berbeda, dengan setiap orang memiliki 10 variasi.

Pola	1	2	3	4	5	6	7	8	9	10
1										
2										
3										

Pola	1	2	3	4	5	6	7	8	9	10
4										
5										
6										
7										
8										
9										

Pola	1	2	3	4	5	6	7	8	9	10
10										
11										
12										
13										
14										
15										

4.3.1 Uji Coba dengan Data Pelatihan Terurut

Hasil yang dicapai dari uji coba dengan data pelatihan terurut dapat dilihat dalam tabel 4.1, yang kesemua data tersebut terurut sesuai dengan nomor indeksinya.

Tabel 4.1
Hasil uji coba pengenalan wajah dengan data pelatihan terurut

Pola	Tipe data terurut 1				Tipe data terurut 2				Tipe data terurut 3			
	Keberhasilan		Kegagalan		Keberhasilan		Kegagalan		Keberhasilan		Kegagalan	
	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase
1	3	43%	4	57%	5	83%	1	17%	4	80%	1	20%
2	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
3	2	29%	5	71%	2	33%	4	67%	2	40%	3	60%
4	6	86%	1	14%	5	83%	1	17%	5	100%	0	0%
5	6	86%	1	14%	5	83%	1	17%	5	100%	0	0%
6	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
7	5	71%	2	29%	5	83%	1	17%	5	100%	0	0%
8	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
9	4	57%	3	43%	5	83%	1	17%	4	80%	1	20%
10	3	43%	4	57%	3	50%	3	50%	3	60%	2	40%
11	4	57%	3	43%	3	50%	3	50%	5	100%	0	0%
12	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
13	4	57%	3	43%	4	67%	2	33%	4	80%	1	20%
14	3	43%	4	57%	2	33%	4	67%	3	60%	2	40%
15	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
16	3	43%	4	57%	4	67%	2	33%	3	60%	2	40%
17	2	29%	5	71%	1	17%	5	83%	1	20%	4	80%
18	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
19	2	29%	5	71%	5	83%	1	17%	4	80%	1	20%

Tabel 4.1
 Hasil uji coba pengenalan wajah dengan data pelatihan terurut
 (lanjutan)

Pola	Tipe data terurut 1				Tipe data terurut 2				Tipe data terurut 3			
	Keberhasilan		Kegagalan		Keberhasilan		Kegagalan		Keberhasilan		Kegagalan	
	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase
20	3	43%	4	57%	2	33%	4	67%	2	40%	4	80%
21	4	57%	3	43%	5	83%	1	17%	4	80%	1	20%
22	7	100%	0	0%	6	100%	0	0%	5	100%	1	20%
23	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
24	6	86%	1	14%	5	83%	1	17%	5	100%	2	40%
25	4	57%	3	43%	4	67%	2	33%	3	60%	2	40%
26	3	43%	4	57%	4	67%	2	33%	3	60%	0	0%
27	3	43%	4	57%	2	33%	4	67%	1	20%	4	80%
28	6	86%	1	14%	5	83%	1	17%	4	80%	1	20%
29	4	57%	3	43%	3	50%	3	50%	4	80%	1	20%
30	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
31	5	71%	2	29%	4	67%	2	33%	3	60%	2	40%
32	4	57%	3	43%	3	50%	3	50%	3	60%	2	40%
33	4	57%	3	43%	3	50%	3	50%	5	100%	0	0%
34	5	71%	2	29%	4	67%	2	33%	4	80%	1	20%
35	6	86%	1	14%	5	83%	1	17%	4	80%	1	20%
36	4	57%	3	43%	3	50%	3	50%	3	60%	2	40%
37	3	43%	4	57%	2	33%	4	67%	2	40%	3	60%
38	6	86%	1	14%	5	83%	1	17%	5	100%	0	0%
39	5	71%	2	29%	4	67%	2	33%	3	60%	2	40%
40	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
Rata-rata		67%		33%		72%		28%		80%		20%
Waktu komputasi	Pelatihan : 1,5 menit				2,5 menit				3,5 menit			
	Pengenalan : 10 detik				20 detik				30 detik			

4.3.2 Uji Coba dengan Data Pelatihan Terkendali

Hasil yang dicapai dari uji coba dengan data pelatihan terkendali dapat dilihat dalam tabel 4.2, yang kesemua data tersebut dipilih dari 10 variasi data dengan pertimbangan variasi data yang ada anpa memandang urutan indeks data..

Tabel 4.2
Hasil uji coba pengenalan wajah dengan data pelatihan terkendali

Pola	Tipe data terkendali 1				Tipe data terkendali 2				Tipe data terkendali 3			
	Keberhasilan		Kegagalan		Keberhasilan		Kegagalan		Keberhasilan		Kegagalan	
	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase
1	3	43%	4	57%	5	83%	1	17%	4	80%	1	20%
2	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
3	3	43%	4	57%	4	67%	2	33%	3	60%	2	40%
4	6	86%	1	14%	5	83%	1	17%	5	100%	0	0%
5	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
6	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
7	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
8	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
9	4	57%	3	43%	5	83%	1	17%	5	100%	0	0%
10	5	71%	2	23%	4	67%	2	33%	3	60%	2	40%
11	4	57%	3	43%	3	50%	3	50%	4	80%	1	20%
12	6	86%	1	14%	5	83%	1	17%	4	80%	1	20%
13	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
14	3	43%	4	57%	5	83%	1	17%	5	100%	0	0%
15	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
16	5	71%	2	29%	4	67%	2	33%	4	80%	1	20%
17	2	29%	5	71%	5	83%	1	17%	5	100%	0	0%
18	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%

Tabel 4.2
 Hasil uji coba pengenalan wajah dengan data pelatihan terkendali
 (lanjutan)

Pola	Tipe data terurut 1				Tipe data terurut 2				Tipe data terurut 3			
	Keberhasilan		Kegagalan		Keberhasilan		Kegagalan		Keberhasilan		Kegagalan	
	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase
19	6	86%	1	14%	5	83%	1	17%	4	80%	1	20%
20	4	57%	3	43%	5	83%	1	17%	4	80%	1	20%
21	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
22	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
23	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
24	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
25	4	57%	3	43%	4	67%	2	33%	4	80%	1	20%
26	3	43%	4	57%	3	50%	3	50%	4	80%	1	20%
27	3	43%	4	57%	3	50%	3	50%	3	60%	2	40%
28	6	86%	1	14%	5	83%	1	17%	4	80%	1	20%
29	4	57%	3	43%	3	50%	3	50%	5	100%	0	0%
30	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
31	5	71%	2	29%	4	67%	2	33%	3	60%	2	40%
32	6	86%	1	14%	5	83%	1	17%	4	80%	1	20%
33	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
34	5	71%	2	29%	4	67%	2	33%	5	100%	0	0%
35	6	86%	1	14%	5	83%	1	17%	4	80%	1	20%
36	5	71%	2	29%	5	83%	1	17%	4	80%	1	20%
37	3	43%	4	57%	3	50%	3	50%	2	40%	3	60%
38	6	86%	1	14%	5	83%	1	17%	4	80%	1	20%
39	5	71%	2	29%	5	83%	1	17%	4	80%	1	20%
40	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
Rata-rata		76%		24%		83%		17%		88%		12%
Waktu komputasi	Pelatihan : 1,5 menit				2,5 menit				3,5 menit			
	Pengenalan : 10 detik				20 detik				30 detik			

4.3.3 Evaluasi Hasil Uji Coba

Prosentase keberhasilan pengenalan wajah seperti yang terlihat pada tabel 4.1 dan 4.2 sangat tergantung pada pemilihan data pelatihan yang digunakan. Pemilihan data pelatihan yang mewakili semua variasi atau mendekati semua variasi dalam data uji coba memberikan hasil pengenalan yang meningkat jika dibandingkan dengan mengambil secara acak atau urut data pelatihan tanpa memandang variasi yang ada. Tabel berikut memberikan perbandingan tingkat keberhasilan dari pemilihan data dari uji coba yang dilakukan.

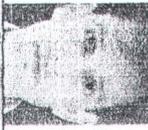
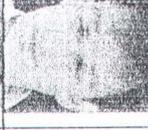
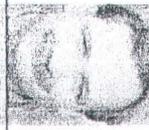
Tabel 4.3
Perbandingan keberhasilan pengenalan berdasarkan metode pemilihan data pelatihan yang digunakan.

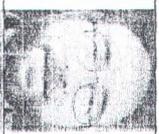
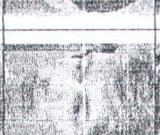
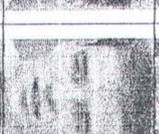
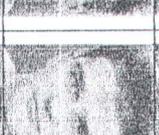
Tipe Data Uji Coba	Uji Coba Terurut			Uji Coba Terkendali		
	Presentase Keberhasilan	Waktu Komputasi		Presentase Keberhasilan	Waktu Komputasi	
		Pelatihan	Pengenalan		Pelatihan	Pengenalan
Jumlah Data 3	67%	1,5 menit	10 detik	76%	1,5 menit	10 detik
Jumlah Data 4	72%	2,5 menit	20 detik	83%	2,5 menit	20 detik
Jumlah Data 5	80%	3,5 menit	30 detik	88%	3,5 menit	30 detik

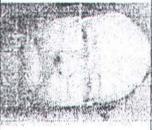
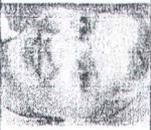
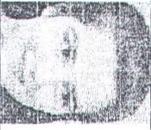
Waktu yang diperlukan untuk melakukan proses pelatihan berbanding lurus dengan banyaknya data yang ditraining. Semakin banyak jumlah data yang ditraining, maka semakin banyak pula waktu yang dibutuhkan. Dari pengamatan hasil uji coba baik dengan data pelatihan terurut maupun data pelatihan terkendali, ada beberapa data yang pengenalannya kurang baik. Perbedaan-perbedaan yang signifikan ini merupakan penyebab faktor kegagalan dalam mengenali citra wajah yang diinputkan ke dalam program tersebut, yaitu :

- Adanya kemiripan wajah antara satu orang dengan orang yang lain.
- Adanya ekspresi wajah yang bervariasi, sedangkan data untuk pelatihan belum mewakili variasi data yang ada.

Pola	1	2	3	4	5	6	7	8	9	10
16										
17										
18										
19										
20										
21										

Pola	1	2	3	4	5	6	7	8	9	10
22										
23										
24										
25										
26										
27										

Pola	1	2	3	4	5	6	7	8	9	10
28										
29										
30										
31										
32										
33										

Pola	1	2	3	4	5	6	7	8	9	10
34										
35										
36										
37										
38										
39										



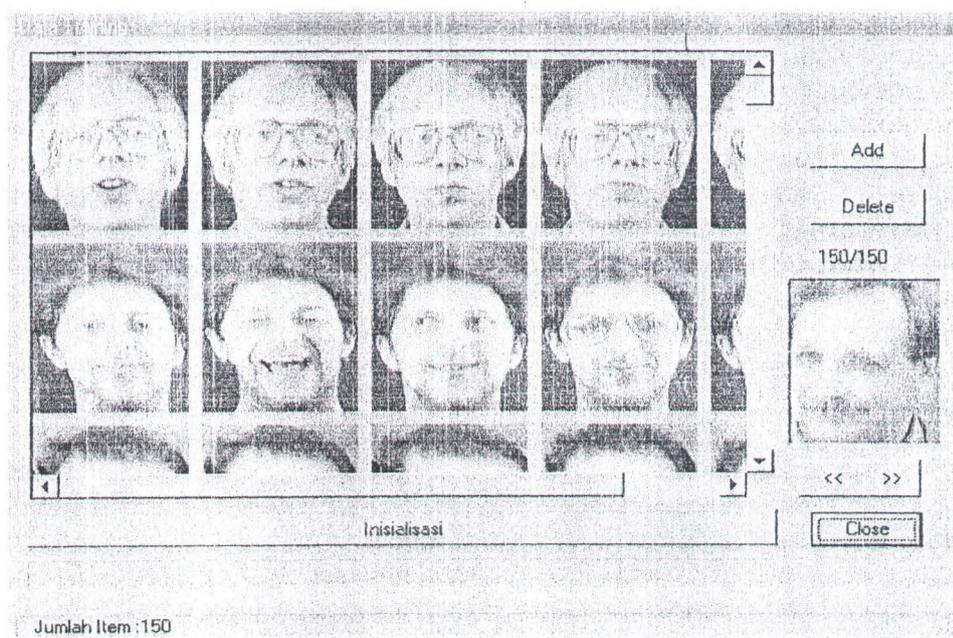
LAMPIRAN B

PETUNJUK PEMAKAIAAN PERANGKAT LUNAK

Langkah-langkah dalam implementasi program :

Langkah-langkah dalam tahap training:

Langkah 1: Aktifkan file prjFace, kemudian click tombol *Open*, dalam tombol open ini dipilih tabel mana yang akan ditraining. Apabila ingin menambah record tabel tekan tombol *Add* dan apabila ingin menghapus record tabel tekan tombol *Delete*.



Gambar B.1 Tampilan dari sebagian wajah training

Langkah 2: Click tombol *Inisialisai*. Tombol ini digunakan untuk membaca gambar yang akan ditraining.

Langkah 3: Click tombol *Training*. Setelah proses training selesai, macam-macam matriks hasil training dapat dilihat dengan menekan tombol *Training Result*.

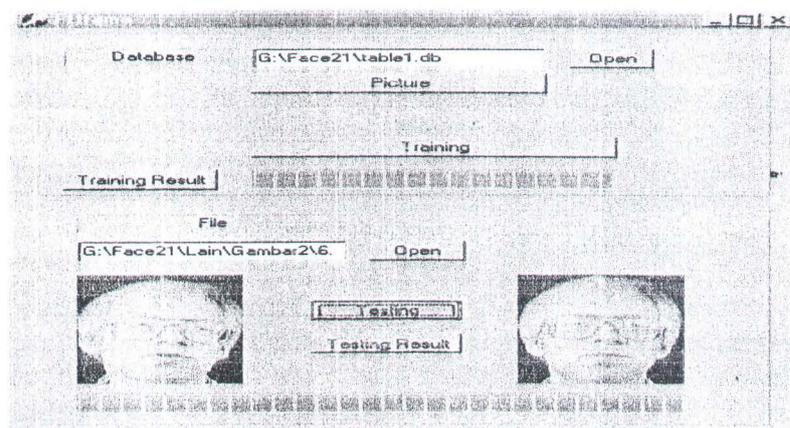
		0.01171492E	0.000665651	-0.01369784E	0.01210447E	0.02169935E	0.001091112	-0.02409829E	0.
	0.01037314E	-0.01800332E	0.005880244	-0.01723666E	0.004329447	-0.00177895E	-0.00068854E	-0.00775574E	0.
	-0.01002904E	-0.01592338E	0.005312817	-0.01696295E	0.00133929E	-0.00373673E	-0.00225634E	-0.00582339E	0.
	0.01025365E	-0.01762612E	0.004259417	-0.01804291E	0.00382479E	-0.00483987E	-0.00256227E	-0.00392725E	0.
	0.01396186E	-0.01795291E	0.00527013E	-0.01719109E	0.00106065E	-0.00637404E	-0.00077291E	-0.00545689E	0.
	-0.01234256E	-0.01734838E	0.00270587E	-0.01880883E	-0.00231825E	-0.00697301E	-0.00101475E	-0.00294918E	0.
	0.00823159E	-0.01948548E	0.00420268E	-0.01846832E	-0.00002780E	-0.00808657E	0.00069007E	-0.00451015E	0.
	-0.00728855E	0.01863453E	0.004423714	-0.01866194E	0.00276340E	-0.00673841E	-0.00190066E	-0.00450492E	0.
	0.00903551E	-0.01889763E	0.005745261	-0.01734636E	0.001183901	-0.00747378E	-0.00199146E	-0.00519384E	0.
	-0.00518875E	-0.01495896E	0.004372822	-0.01485503E	-0.00084990E	-0.00982137E	-0.00075790E	-0.00335315E	0.
	-0.00555196E	-0.01547944E	-0.00165077E	-0.01457653E	-0.00394995E	-0.01075567E	0.00068226E	-0.00000157E	0.

Gambar B.2 Tampilan matriks hasil dari proses training

Langkah-langkah dalam tahap testing:

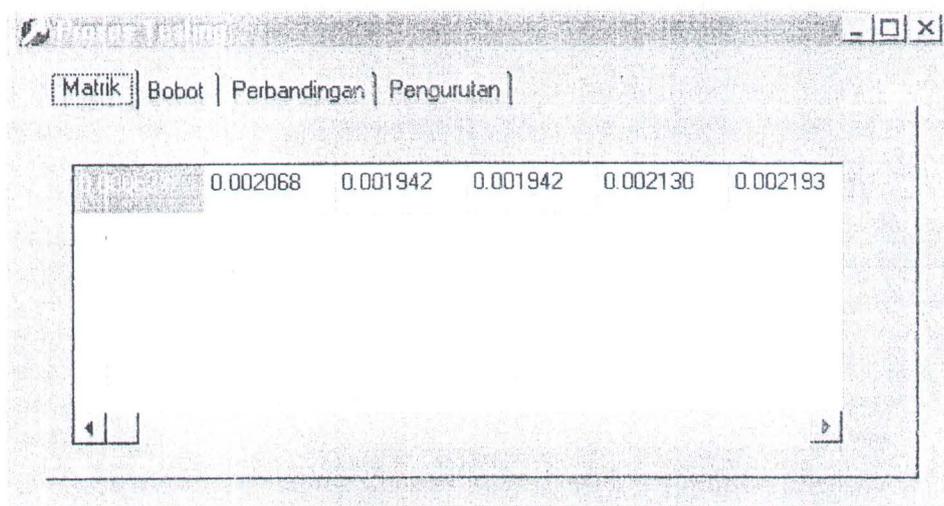
Langkah 1: Click tombol *Open*, tombol ini digunakan untuk menginputkan gambar yang akan diuji coba.

Langkah 2 : Click tombol *Testing*, tombol ini digunakan untuk melakukan proses testing. Setelah proses testing selesai, maka akan tampak gambar yang hampir mendekati dengan gambar ujicoba.



Gambar B.3 Tampilan hasil dari Testing

Langkah 3 :Macam-macam matriks hasil testing dapat dilihat dengan menekan tombol *Testing Result*.



The screenshot shows a software window with a title bar and a menu bar. The menu bar contains the following items: Matrik, Bobot, Perbandingan, and Pengurutan. The main area of the window displays a matrix of numerical values. The first row of the matrix contains the following values: 0.002068, 0.001942, 0.001942, 0.002130, and 0.002193. The window also features a scroll bar on the right side and navigation arrows at the bottom left and bottom right corners.

0.002068	0.001942	0.001942	0.002130	0.002193
----------	----------	----------	----------	----------

Gambar B.4 Tampilan matriks dari hasil proses Testing