TA / SEKJUR TE / 2007 / 066

PENGENALAN SIDIK JARI DENGAN JARINGAN SARAF TIRUAN BERBASIS METODE PERAMBATAN BALIK

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Elektro



Oleh:

Nama : Amalia Al Annas

No. Mahasiswa : 01 524 106



JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
JOGJAKARTA
2007

LEMBAR PENGESAHAN PEMBIMBING

PENGENALAN SIDIK JARI DENGAN JARINGAN SARAF TIRUAN BERBASIS METODE PERAMBATAN BALIK

TUGAS AKHIR

Disusun Oleh :

Nama : Amalia Al Annas

No.Mahasiswa : 01 524 106

Jogjakarta, Juli 2007

Dosen Pembimbing I	Dosen Pembimbing II
(Tito Yuwono, ST, M.Sc)	(Dwi Ana Ratna Wati, ST)

HALAMAN PERSEMBAHAN

Kupersembahkan Untuk

Keluargaku Tercinta :

Bapak Djahuri dan Mama Ruminah

Serta

Mba Rima, Mba Inna, Annisa & Ugi

Terima kasih atas semua pengorbanan,

dorongan semangat yang tiada henti dan

do'anya yang tulus



MOTTO

Allah meninggikan orang yang beriman diantara kamu dan orang - orang yang diberikan ilmu pengetahuan beberapa derajat.

(Al Mujaadilah : 11)

Dialah Yang menjadikan bumi itu mudah bagi kamu, maka berjalanlah di segala penjurunya dan makanlah sebagian dan rezki-Nya. Dan hanya kepada-Nya-lah kamu kembali setelah dibangkitkan.

(Al Mulk : 15)

Pengetahuan adalah satu - satunya kekayaan yang tidak dapat dirampas. Hanya kematian yang bisa memadamkan lampu pengatahuan yang ada dalam dirimu.

(Kahlil Gibran)

Hidup adalah soal keberanian, menghadapi tanda tanya. Tanpa kita bisa mengerti, tanpa kita bisa menawar. Terimalah dan hadapilah.

(Soe Hoek-gie)

Don't say "I can't do it" if you never try yet. Believe, that your future in your hand, which depend on your own capability and Allah Blessing's

(By Fich)

KATA PENGANTAR

بسم الله الرحمن الرحيم

Assalamu'alaikum Wr.Wb.,

Alhamdulillah, segala puji dan syukur kehadirat Allah SWT, semoga kita semua senantiasa berada dalam lindungan, rahmat dan hidayahNya, karena berkat rahmat dan hidayahNyalah sehingga penulis dapat menyelesaikan tugas akhir ini. Tugas akhir ini adalah salah satu syarat guna menyelesaikan jenjang kesarjanaan strata 1 (S1) pada jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Dalam penulisan tugas akhir ini penulis banyak memperoleh petunjuk, bimbingan, bantuan, serta dorongan dari berbagai pihak. Oleh karena itu, penulis mengucapkan banyak terima kasih kepada :

1. Bapak Fathul Wahid, ST., MSc., selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
2. Bapak Tito Yuwono, ST., MSc., selaku Kepala Jurusan Teknik Elektro, Universitas Islam Indonesia dan juga selaku dosen pembimbing I, yang telah memberikan pengarahan dan bimbingan selama penyusunan tugas akhir ini.
3. Ibu Dwi Ana Ratna Wati, ST., selaku dosen pembimbing II, yang telah membantu dan memberikan bimbingan kepada penulis dalam penyusunan laporan ini.
4. Bapak Wahyudi Budi Pramono, ST., selaku dosen penguji.
5. Bapak Hendra Setiawan, ST., MT., selaku dosen penguji.

6. Seluruh karyawan dan Staff Fakultas Teknologi Industri, Universitas Islam Indonesia.
7. Kedua orang tuaku dan keluarga besar kami atas dorongan moril dan materiilnya sehingga laporan tugas akhir ini dapat terselesaikan.
8. Teman - teman kost seperti Seli, Ruli, Linda, mba Erna, mba Surati, dan tak lupa buat Bapak dan Ibu kost tercinta.
9. Teman - teman Elektro terutama cewek - cewek angkatan 2001 yaitu Dewi, Amin El Badr, V3, Wina, Ani, Indri, Nihla. Untuk Arif terima kasih atas bantuannya selama ini. Serta ucapan terima kasih untuk April, Asri, Rahma, Rulli, Hima, Adi dan lain - lain yang tidak bisa disebutkan satu - persatu.
10. Semua pihak yang telah membantu dalam penyusunan laporan tugas akhir ini.

Banyak kendala dan pengorbanan, yang penulis alami selama penyusunan laporan ini. Hanya dengan pertolongan Allah SWT penulis bisa mengatasi semua itu. Penulis menyadari bahwa laporan tugas akhir ini masih terdapat kekurangan di dalamnya, sehingga penulis mengharapkan kritik dan saran yang bersifat membangun dari pembaca demi lengkapnya laporan ini. Semoga laporan ini dapat bermanfaat bagi kita semua. Amiin yaa robbal 'alamin

Wassalamu'alaikum Wr.Wb.,

Jogjakarta, Juni 2007

Penulis

ABSTRAK

Dalam penelitian ini dimanfaatkan Jaringan Saraf Tiruan metode *backpropagation* untuk mengenali berbagai macam pola sidik jari sebagai tanda pengenal spesifik bagi setiap individu yang berbeda - beda. Data pola sidik jari diperoleh melalui identifikasi dalam bentuk *image* sidik jari yang di-*capture* dengan menggunakan *biopod fingerprint sensor*. Metode yang digunakan dalam pendeteksi sidik jari adalah metode pengenalan pola. Pengolahan *image* digunakan untuk memproses gambar atau *image* sidik jari dengan jalan memanipulasinya menjadi data gambar yang siap diolah dengan Jaringan Saraf Tiruan. Proses *training* dilakukan untuk mengenali pola sidik jari yang telah didapatkan. Dengan metode *backpropagation* diperoleh struktur jaringan yang terbaik yaitu terdiri dari 4 lapisan tersembunyi dengan *neuron* 80, 60, 40, 20 dan 6 sel *neuron* lapisan *output*. Fungsi aktivasi pada setiap lapisan menggunakan fungsi *sigmoid biner* (*logsig*), dengan *learning rate* 0.02, *momentum coefisient* 0.01, *maximum epoch* 20000 dan *galat* 0.00001, menghasilkan *Mean Square Error* (MSE) 9.96498e-006. Penelitian ini mencoba mengenali sidik jari dalam pengujian pembacaan nama pemilik sidik jari tersebut. Proses pengujian dilakukan untuk mendeteksi masukan pola sidik jari dan menghasilkan keluaran yang sesuai dengan data yang tersimpan dalam *database*. Pengujian dilakukan terhadap 10 orang responden yang mempunyai karakteristik berbeda - beda dan masing - masing diambil 5 sidik jari tangan kanan sehingga didapatkan 50 data sampel sidik jari. Pengujian dengan menggunakan data pelatihan, memiliki *prosentase* keberhasilan 100%. Sedangkan pengujian dengan data bukan dari data pelatihan menghasilkan *prosentase* keberhasilan 40%.

DAFTAR ISI

Halaman Judul i

Lembar Pengesahan Pembimbing ii

Lembar Pengesahan Penguji iii

Halaman Persembahan iv

Halaman Motto v

Kata Pengantar vi

Abstrak viii

Daftar Isi ix

Daftar Gambar xiii

Daftar Tabel xv

BAB I PENDAHULUAN 1

1.1 Latar Belakang Masalah 1

1.2 Rumusan Masalah 2

1.3 Batasan Masalah 2

1.4 Tujuan Penelitian 3

1.5 Sistematika Penulisan 4

BAB II LANDASAN TEORI 5

2.1 Biometrik Sidik Jari 5

2.2 Alat Deteksi Sidik Jari 9

2.3 Pengolahan *Image* 10

2.3.1 *Image* Digital 10

2.3.2 *Image* Berwarna..... 11

2.3.3 *Image* Keabuan..... 11

2.3.4 Proses *Capture Image*..... 11

2.3.5 Peningkatan Nilai Intensitas *Image*..... 12

2.3.6 *Image* Biner..... 12

2.4 Pembentukan Pola..... 12

2.5 Deteksi Sidik Jari dalam *Image*..... 13

2.6 Jaringan Saraf Tiruan..... 14

2.6.1 Permodelan *Neuron*..... 15

2.6.2 Arsitektur Jaringan Saraf Tiruan..... 16

2.6.2.1 Jaringan Umpanmaju Lapis Tunggal (*Feed Forward Single Layer*)..... 17

2.6.2.2 Jaringan Umpanmaju Lapis Banyak (*Multilayer Perceptron*)..... 17

2.6.2.3 Jaringan Saraf Tiruan Berulang (*Recurrent*)..... 18

2.6.3 Pengaturan Bobot..... 18

2.6.3.1 Pelatihan Terbimbing (*Supervised Training*)..... 18

2.6.3.2 Pelatihan Tak Terbimbing (*Unsupervised Training*)..... 18

2.7 Perambatan Balik (*Backpropagation*)..... 19

2.8 Fungsi Aktivasi *Backpropagation*..... 20

BAB III PERANCANGAN SISTEM..... 22

3.1 Deskripsi Sistem..... 23

3.1.1 Perangkat Keras (*hardware*)..... 23

3.1.2 Perangkat Lunak (*software*)..... 23

3.2 Perancangan Sistem Pengenalan Pola..... 24

3.2.1 *Interface* Menu Utama Program..... 25

3.2.2 *Interface* Pelatihan..... 26

3.2.3 *Interface* Pengujian..... 29

3.3 Objek yang Digunakan..... 32

3.4 Membangun Jaringan Saraf Tiruan *Backproagation*..... 35

3.5 Pelatihan Jaringan Saraf Tiruan *Backpropagation*..... 37

3.6 Pengujian Jaringan Saraf Tiruan *Backpropagation*..... 40

BAB IV ANALISIS DAN PEMBAHASAN..... 42

4.1 Hasil Pelatihan Jaringan Saraf Tiruan *Backpropagation*..... 42

4.1.1 Pelatihan Satu Lapisan Tersembunyi..... 45

4.1.2 Pelatihan Lebih Dari 1 lapisan Tersembunyi..... 45

4.2 Hasil Pengujian Jaringan Saraf Tiruan *Backpropagation*..... 48

4.2.1 Hasil Pengujian Jaringan Saraf Tiruan *Backpropagation* Pada Sistem Pengenalan Pola Sidik Jari Terhadap Data Pelatihan..... 48

4.2.2 Hasil Pengujian Jaringan Saraf Tiruan *Backpropagation* Pada Sistem Pengenalan Pola Sidik Jari Tanpa Menggunakan Data Pelatihan..... 50

4.2.3 *Prosentase* Hasil Pengenalan Pola Sidik Jari..... 52

BAB V PENUTUP 54

5.1 Kesimpulan 54

5.2 Saran 55

Daftar Pustaka xvi

Lampiran xvii

DAFTAR GAMBAR

Gambar 2.1 Perangkat akses menggunakan sidik jari..... 6

Gambar 2.2 Contoh dari pola sidik jari..... 7

Gambar 2.3 Beberapa contoh pola *papillary ridge*..... 9

Gambar 2.4 *Biopod Fingerprint Sensor*..... 9

Gambar 2.5 Contoh matriks 130x130 yang digunakan sebagai pola *input* pada sistem pengenalan pola sidik jari..... 13

Gambar 2.6 Contoh matriks 1x50 yang berisi matriks 130x130 digunakan sebagai pola *input* pada sistem pengenalan pola sidik jari..... 13

Gambar 2.7 Sistem neuron biologis dan sistem neuron buatan..... 15

Gambar 2.8 Model *McCulloch - Pitts* untuk *neuron*..... 16

Gambar 2.9 Jaringan umpanmaju lapis tunggal..... 17

Gambar 2.10 Jaringan umpanmaju tiga lapis..... 17

Gambar 2.11 Arsitektur jaringan saraf tiruan *backpropagation*..... 19

Gambar 2.12 Fungsi aktivasi *linear* / identitas..... 20

Gambar 2.13 Fungsi aktivasi *sigmoid biner*..... 21

Gambar 2.14 Fungsi aktivasi *sigmoid bipolar*..... 21

Gambar 3.1 Blok diagram sistem pengenalan pola sidik jari..... 24

Gambar 3.2 *Interface* menu_utama sistem pengenalan pola sidik jari..... 25

Gambar 3.3 *Interface* modus_pelatihan sistem pengenalan pola sidik jari..... 26

Gambar 3.4 Diagram alir (*flow chart*) proses pelatihan jaringan saraf tiruan *backpropagation* menggunakan *interface* pelatihan..... 28

Gambar 3.5 *Interface* modus_pengujian sistem pengenalan pola sidik jari.... 29

Gambar 3.6 Diagram alir (*flow chart*) proses pengenalan pola sidik jari..... 32

Gambar 3.7 Blok diagram pelatihan jaringan saraf tiruan *backpropagation*.. 38

Gambar 3.8 Diagram alir (*flow chart*) pelatihan jaringan saraf tiruan *backpropagation* dengan satu lapisan tersembunyi 39

Gambar 3.9 Diagram alir (*flow chart*) pengujian jaringan saraf tiruan *backpropagation* dengan satu lapisan tersembunyi..... 41

Gambar 4.1 Grafik keluaran dari Matlab sebagai hasil pelatihan dengan 4 lapisan tersembunyi, jumlah neuron 80, 60, 40, 20 *learning rate* 0.02, *momentum* 0.01, galat = 0.00001, dan epoh 20000..... 47

DAFTAR TABEL

Tabel 3.1 Objek yang akan digunakan dalam program pengenalan pola sidik jari..... 33

Tabel 3.2 Pola *input* dan pola target pelatihan..... 36

Tabel 4.1 Hasil pelatihan jaringan saraf tiruan pada sistem pengenalan pola sidik jari..... 43

Tabel 4.2 Hasil pengenalan pola sidik jari melalui pelatihan..... 49

Tabel 4.3 Hasil pengenalan pola sidik jari tanpa melalui pelatihan..... 50

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pola sidik jari bagi setiap orang mempunyai karakteristik yang berbeda - beda antara satu dengan yang lain, dan kecil kemungkinan terjadinya persamaan pola sidik jari diantara individu tersebut, dengan memanfaatkan suatu sistem yang dapat mengenali berbagai macam pola sidik jari, dapat diciptakan suatu sistem tanda pengenal spesifik bagi setiap individu yang berbeda - beda. Metode yang digunakan dalam penelitian sidik jari pada umumnya menggunakan metode pengenalan pola. Penggunaan metode ini akan berhasil dengan baik jika digunakan untuk mendeteksi sidik jari dengan tekstur atau pola tertentu dan tidak akan berhasil dengan baik jika digunakan untuk mendeteksi sidik jari dengan tekstur atau pola yang berbeda. Karena keterbatasan dari metode pengenalan pola maka untuk mendeteksi sidik jari diperlukan metode yang lebih baik.

Pengolahan *image* adalah suatu teknik yang dapat digunakan untuk memproses gambar atau *image* dengan jalan memanipulasinya menjadi data gambar yang diinginkan untuk mendapat informasi tertentu. Dengan mata, manusia dapat mendeteksi benda dengan menggunakan otak. Dengan menggunakan kecerdasan buatan (*artificial intelligence*) dan bantuan pengolahan *image*, komputer dapat menirukan prinsip kerja otak.

Salah satu cara yang dapat digunakan untuk mencapai tujuan tersebut adalah dengan menggunakan sistem *Artificial Neural Network. Artificial Neural*

Network adalah suatu sistem analisis yang digunakan untuk memecahkan permasalahan yang persamaan penyelesaiannya dinyatakan secara implisit. Jaringan saraf tiruan adalah salah satu metode komputasi yang berusaha meniru cara kerja sistem saraf otak manusia. Keunggulan metode ini dibandingkan metode lain adalah kemampuannya belajar dan memecahkan hubungan yang rumit, yang sulit dideskripsikan antara data masukan dan data keluaran. Hal ini dimungkinkan karena pengetahuan yang ada pada jaringan saraf tiruan tidak diprogram, namun dilatih berdasarkan informasi yang diterimanya. Jaringan saraf tiruan telah dikembangkan sebagai generalisasi model matematik dari pembelajaran manusia. Beberapa peneliti telah berhasil mengaplikasikan jaringan saraf tiruan untuk menyelesaikan masalah identifikasi.

Dalam tugas akhir ini akan dicoba untuk mengaplikasikan jaringan saraf tiruan untuk mengenali pemilik sidik jari dengan program bantu perangkat lunak Matlab v.7.0. Algoritma belajar yang akan digunakan dalam tugas akhir ini adalah perambatan - balik (*backpropagation*).

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, maka dapat di ambil suatu rumusan masalah sebagai berikut : "Bagaimana merealisasikan perancangan model dan pemanfaatan aplikasi jaringan saraf tiruan pada pembacaan sidik jari".

1.3 Batasan Masalah

Agar permasalahan yang dibahas dalam tugas akhir ini tidak menyimpang dari judul yang telah ditetapkan maka penulis membatasinya pada masalah :

1. Perancangan sistem pendeteksi sidik jari dengan algoritma belajar perambatan - balik (*back propagation*) dalam pembuatan simulasi jaringan saraf tiruan.
2. Pembuatan program simulasi pada perancangan model deteksi sidik jari menggunakan Matlab v.7.0
3. Proses *capturing*, pengolahan *image* dan pembentukan jaringan saraf tiruan *backpropagation* dilakukan dengan menggunakan fungsi-fungsi yang sudah ada *Image Processing Toolbox* dan *Neural Network Toolbox.*
4. Objek sidik jari yang akan dikenali dibatasi hanya sepuluh orang dan masing - masing diambil lima sampel sidik jari tangan kanan sehingga totalnya lima puluh image sidik jari.
5. Pelatihan dan pengujian jaringan saraf tiruan menggunakan fungsi yang terdapat dalam *Neural Network Toolbox.*
6. Pelatihan terhadap jaringan saraf tiruan dilakukan dengan memberikan contoh matriks dari pola bentuk objek yang akan dikenali, dengan menggunakan *image* objek yang telah di-*capture* secara tidak langsung menggunakan *biopod fingerprint sensor.*
7. Pengujian terhadap jaringan saraf tiruan dilakukan dengan menggunakan sidik jari yang *image*nya secara tidak langsung di-*capture* menggunakan *biopod fingerprint sensor.*

1.4 Tujuan Penelitian

Tujuan dari pembuatan tugas akhir ini adalah membuat program bantu berupa simulasi pendeteksi sidik jari yang berfungsi mengenali nama pemilik

sidik jari dengan menggunakan jaringan saraf tiruan dengan algoritma belajar perambatan - balik (*backpropagation*), yang kemudian sistem ini bisa digunakan sebagai penunjang pada aplikasi seperti mesin absensi sidik jari, sistem sekuriti *database* dengan sidik jari untuk membuka password dan lain sebagainya.

1.5 Sistematika Penulisan

Sistematika penulisan pada laporan tugas akhir ini terdiri dari lima bab yaitu :

BAB I : PENDAHULUAN

Bab ini menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, dan sistematika penulisan.

BAB II : LANDASAN TEORI

Bab ini menjelaskan teori - teori yang berhubungan dengan penelitian yang akan dilakukan.

BAB III : PERANCANGAN SISTEM

Bab ini menjelaskan proses perancangan perangkat lunak aplikasi pengenalan pola sidik jari dengan jaringan saraf tiruan menggunakan perangkat keras *biopod fingerprint sensor* secara tidak langsung.

BAB IV : ANALISIS DAN PEMBAHASAN

Bab ini menjelaskan analisis hasil pengujian terhadap sistem pengenalan pola sidik jari yang telah dibuat.

BAB V : PENUTUP

Bab ini berisi kesimpulan dan saran dari perancangan sistem yang telah dilakukan.

BAB II

LANDASAN TEORI

2.1 Biometrik Sidik Jari

Sistem keamanan menggunakan sidik jari, kadang - kadang sering juga disebut sebagai sistem identifikasi biometrik. Teknologi biometrik dikembangkan karena dapat memenuhi dua fungsi yaitu identifikasi dan verifikasi, disamping itu biometrik memiliki karakteristik seperti tidak dapat hilang, tidak dapat lupa dan tidak mudah dipalsukan karena keberadaannya melekat pada manusia, dimana satu dengan yang lain tidak akan sama, maka keunikannya akan lebih terjamin.

Biometric autentification dalam *security* adalah hal yang sangat penting untuk menjaga keamanan data, namun sudah banyak teknologi yang diterapkan untuk menjaga keautentikan tersebut, akan tetapi hal itu banyak kendala dalam penerapannya dan masih kurang memberikan perlindungan yang aman. Teknologi biometrik menawarkan autentikasi secara biologis memungkinkan sistem dapat mengenali penggunanya lebih tepat.

Identifikasi dengan menggunakan pola sidik jari merupakan teknik biometrik yang tertua di dunia. Pada tahun 6000 sebelum Masehi, sidik jari digunakan sebagai kode personal, dan telah digunakan oleh bangsa Assyiria, Babylonia, Jepang dan Cina. Bangsa Cina kuno menggunakan sidik jari sebagai alat identifikasi pembuktian seseorang sebagai pengarang dari suatu dokumen, dan sejak tahun 1897, *dactyloscopy* (sinonim dari identifikasi sidik jari tanpa berbasis komputer) telah digunakan untuk identifikasi kejahatan.

Sidik jari telah terbukti cukup akurat, aman, mudah, dan nyaman untuk dipakai sebagai identifikasi bila dibandingkan dengan sistem biometrik lainnya seperti retina mata atau DNA. Hal ini dapat dilihat pada sifat yang dimiliki oleh sidik jari, antara lain :

1. *Perennial nature*, yaitu guratan - guratan pada sidik jari yang melekat pada kulit manusia seumur hidup.
2. *Immutability*, yaitu sidik jari seseorang tidak pernah berubah, kecuali mendapatkan kecelakaan yang serius.
3. *Individuality*, pola sidik jari adalah unik dan berbeda untuk setiap orang.

Dari ketiga sifat ini, sidik jari dapat digunakan sebagai sistem identifikasi yang dapat digunakan dalam aplikasi teknologi informasi seperti :

1. *Access System Security*, yaitu akses untuk masuk ke suatu area atau ruangan tertentu yang terlarang dan terbatas *(restricted)*.
2. *Authentification System*, yaitu untuk akses data yang sifatnya rahasia dan terbatas (misalnya data pada perbankan, militer dan diplomatik).

Contoh aplikasi sidik jari yang digunakan dalam teknologi informasi, dapat dilihat pada gambar 2.1



Gambar 2.1 Perangkat akses menggunakan sidik jari

Suatu pola sidik jari normal dibentuk dari garis - garis *(lines)* dan spasi *(spaces)*. Garis - garis ini dinamakan *ridge* sedangkan spasi antara garis - garis ini dinamakan *valley*. Dengan melalui pola *ridge* dan *valley* inilah sebuah sidik jari yang unik disesuaikan untuk verifikasi dan autorisasi. Ciri yang unik dari sidik jari ini dinamakan *minutiae* dan perbandingannya berdasarkan ciri - ciri minutiae tersebut.

Di Amerika Utara, seorang bernama *E. Henry* pada tahun 1901 telah sukses lebih dahulu menggunakan sidik jari untuk identifikasi pemberhentian pekerja untuk mengatasi pemberian upah ganda.



Gambar 2.2 Contoh dari pola sidik jari

Ridge berwarna hitam sedangkan valley berwarna putih

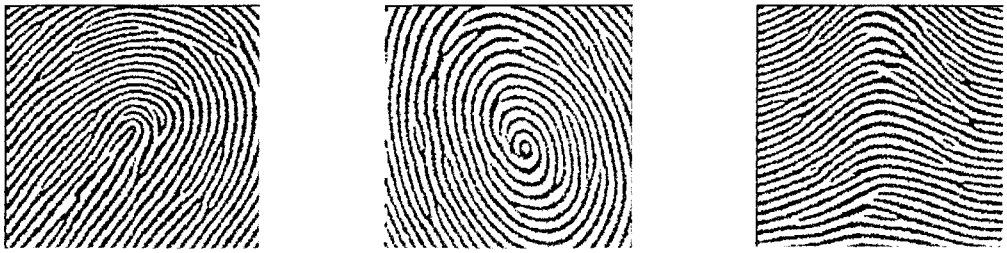
Sistem *Henry* berasal dari pola *ridge* yang terpusat pada pola jari tangan, jari kaki, khususnya telunjuk. Metoda yang klasik dari tinta dan menggulung jari pada suatu kartu cetakan menghasilkan suatu pola *ridge* yang unik bagi masing - masing digit individu.

Karakteristik sidik jari dari setiap orang adalah unik dan secara normal tidak berubah selama hidupnya. Dari penelitian yang seksama, diperkirakan bahwa peluang dua orang, meskipun kembar, mempunyai pola sidik jari yang sama adalah lebih kecil dari satu di dalam satu milyar.

Hal ini telah dapat dipercaya membuktikan bahwa tidak ada dua individu mempunyai pola *ridge* serupa, pola *ridge* tidaklah bisa menerima warisan, pola *ridge* dibentuk embrio, pola *ridge* tidak pernah berubah dalam hidup, dan hanya setelah kematian dapat berubah sebagai hasil pembusukan. Dalam hidup, pola *ridge* hanya diubah secara kebetulan akibat luka - luka, kebakar, penyakit atau penyebab lain yang tidak wajar. Identifikasi dari sidik jari memerlukan pembedaan tentang bentuk keliling *papillary ridge* tak terputuskan yang diikuti oleh pemetaan tentang gangguan atau tanda *anatomic ridge* yang sama. Ada 7 pola *papillary ridge*:

- *Loop*
- *Whorl*
- *Arch*
- *Tented Arch*
- *Double Loop*
- *Central Pocked Loop dan*
- *Accidental*

Menurut *Francis Galton* (1822 - 1916) mengatakan bahwa tidak ada dua sidik jari yang sama, artinya setiap sidik jari yang dimiliki oleh seseorang adalah unik. Berdasarkan klasifikasi, dari ketujuh pola tersebut di atas ada tiga bentuk yang paling umum digambarkan yaitu *loop* (mempunyai 1 delta), *whorl* (mempunyai 2 delta dan antar baris delta terlihat jelas), dan *arch* tidak mempunyai delta.



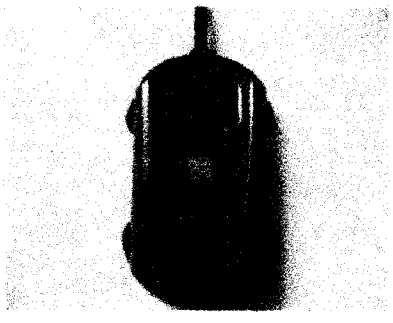
loop pattern whorl pattern arch pattern

Gambar 2.3 Beberapa contoh pola *papillary ridge*

Semua pola di atas dapat dibedakan oleh mata biasa dan dapat memberi suatu *binning* atau *indexing* yang menghasilkan *database*. Sebuah komputer dapat menganalisis garis tengah perubahan arah bentuk *ridge*. Karakteristik *anatomic* terjadi sebab *papillary ridge* tidaklah berlanjut. Masing - masing perubahan arah, pencabangan dua, gangguan menghasilkan karakteristik *anatomic* (*minutiae* karena penyelenggaraan perkawinan). Karakteristik ini tidak mungkin dilihat langsung oleh mata manusia tetapi mudah di-*tracked* oleh komputer.

2.2 Alat Deteksi Sidik Jari

Pada perancangan sistem pendeteksi sidik jari ini alat yang digunakan untuk pengambilan image adalah *biopod fingerprint sensor*. Sehingga data yang diolah merupakan data hasil *capturing image* yang menggunakan *biopod fingerprint sensor* secara tidak langsung.



Gambar 2.4 *Biopod Fingerprint Sensor*

2.3 Pengolahan Image

Pengolahan *image* adalah suatu metode yang dapat digunakan untuk memproses gambar atau *image* dengan jalan memanipulasinya menjadi data gambar yang diinginkan untuk mendapat informasi tertentu.

Dengan mata, manusia dapat mendeteksi benda dengan menggunakan otak. Dengan menggunakan kecerdasan buatan dan bantuan pengolahan *image*, komputer dapat menirukan prinsip kerja otak. Jaringan saraf tiruan telah dikembangkan sebagai generalisasi model matematik dari pembelajaran manusia. Beberapa peneliti telah berhasil mengaplikasikan jaringan saraf tiruan untuk menyelesaikan masalah identifikasi.

Pengolahan *image* bertujuan memperbaiki kualitas agar mudah diiterpretasikan oleh manusia atau mesin (dalam hal ini komputer). Supaya *image* dapat diolah oleh sebuah komputer digital, maka sebuah *image* harus disimpan pada format yang dapat diolah oleh sebuah program komputer. Cara paling praktis yang dapat dilakukan adalah dengan membagi *image* menjadi sekumpulan sel-sel diskret, yang disebut piksel.

2.3.1 *Image* Digital

Image digital merupakan keluaran proses fotografi digital yang mengandung informasi. *Image* digital tersebut dicuplik dan dipetakan pada kisi-kisi elemen *image* yang disebut dengan piksel. *Image* dapat dinyatakan dalam titik - titik koordinat pada kawasan ruang (spasial) atau bidang dan untuk menentukan warna atau menyatakan nilai keabuan suatu *image*.

2.3.2 *Image* Berwarna

Sebuah *image* berwarna (RGB/*red green blue*) adalah sebuah *image* multi spektral dengan satu pita (*band*) untuk masing-masing warna merah, hijau, dan biru, sehingga setiap piksel pada domain spasial *image* merupakan kombinasi dari ketiga warna primer tersebut. Pada *image* berwarna (RGB) intensitas tiga komponen warna dinyatakan pada masing - masing piksel.

2.3.3 *Image* Keabuan

Image beraras keabuan adalah *image* yang hanya menggunakan warna yang merupakan tingkatan warna abu - abu. Warna abu - abu adalah satu-satunya warna pada ruang RGB dengan komponen merah, hijau, dan biru mempunyai intensitas yang sama. Pada *image* beraras keabuan hanya perlu menyatakan nilai intensitas untuk tiap piksel sebagai nilai tunggal, sedangkan pada *image* berwarna perlu tiga nilai intensitas untuk tiap pikselnya.

2.3.4 Proses *Capture Image*

Image yang dihasilkan dari proses *capturing* menggunakan *biopod fingerprint sensor* secara tidak langsung, di*edit* dan dibuat file image berformat jpg. Hal ini berarti bahwa *image* merupakan sebuah *image* RGB (*Red, Blue, Green*) yang memiliki tiga derajat keabuan untuk setiap pikselnya. Hal ini akan mempersulit pada saat penentuan *feature* yang akan digunakan untuk pengenalan pola, sehingga sebelum diproses lebih lanjut *image RGB* harus diubah terlebih dahulu menjadi sebuah *image grayscale* yang hanya memiliki dua derajat keabuan. Pada Matlab hal ini dapat dilakukan dengan menggunakan fungsi :

```
Image = rgb2gray (Image hasil proses capturing );
```

2.3.5 Peningkatan Nilai Intensitas *Image*

Pada tugas akhir ini setelah *image* diubah kedalam tipe *grayscale* kemudian *image* ditingkatkan kontrasnya dengan cara menambahkan nilai intensitasnya dengan menggunakan fungsi berikut:

```
Image = imadjust(Image yang ditingkatkan kontrasnya);
```

2.3.6 *Image* Biner

Image biner (*binary image*) adalah *image* yang hanya mempunyai dua nilai derajat keabuan: hitam dan putih. Pada beberapa aplikasi *image* biner masih tetap dibutuhkan, meskipun *image* berwarna lebih disukai. *Image* biner hanya mempunyai dua derajat keabuan: hitam dan putih, piksel-piksel objek bernilai '1' dan piksel-piksel *background* bernilai '0'. Pada waktu menampilkan gambar, '0' adalah hitam dan '1' adalah putih. Proses pembentukan *image* biner dari *image grayscale* pada Matlab menggunakan fungsi sebagai berikut :

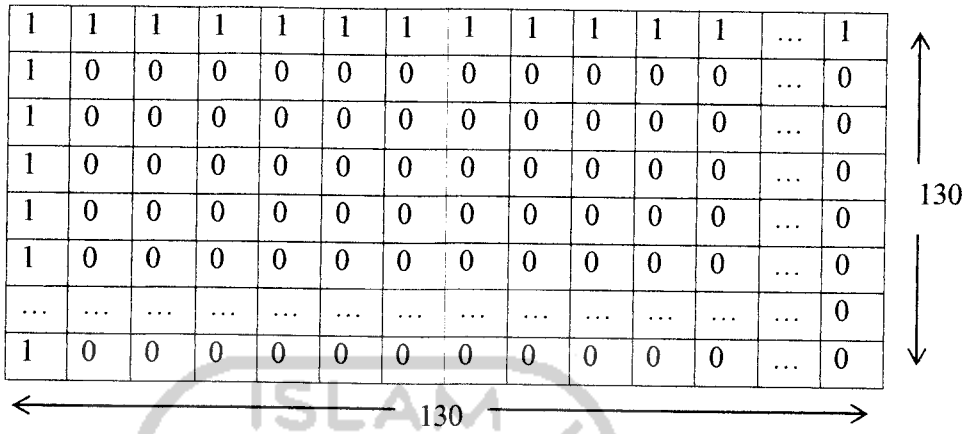
```
Image = im2bw (Image yang akan diubah,graythresh(Image
yang akan diubah);
```

2.4 Pembentukan Pola

Pola yang digunakan pada tugas akhir ini merupakan sebuah matriks 130x130, dari *image* biner yang dihasilkan pada proses pengolahan *image*. *Image* biner hanya memiliki piksel-piksel yang nilai bernilai '1' dan '0', sehingga elemen-elemen dalam matriks tersebut juga hanya terdiri nilai '1' dan '0' dengan kata lain *feature* yang dipilih untuk digunakan pada tugas akhir ini adalah elemen-elemen dalam matriks *image* biner. Untuk memperoleh matriks 130x130 dari sebuah *image* biner dapat dilakukan dengan menggunakan fungsi sebagai berikut:

```
I = imresize(I, [130 130], 'bicubic');
```

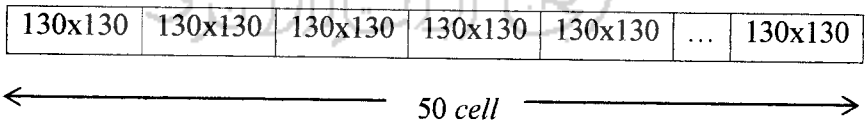
Sehingga diperoleh keluaran sebuah matriks berukuran 130x130 sebagai berikut :



Gambar 2.5 Contoh matriks 130x130 yang digunakan sebagai pola input pada sistem pengenalan pola sidik jari

Kemudian matriks berukuran 130x130 yang akan dikenali dibentuk kembali menjadi matriks berukuran 1x50, sehingga tiap *cell* berisi matriks berukuran 130x130 yang akan digunakan sebagai input pada jaringan saraf tiruan *backpropagation* menggunakan fungsi sebagai berikut:

```
IProc = reshape(IProc,16900,1);
IProc = num2cell(IProc,1);
```



Gambar 2.6 Contoh matriks 1x50 yang berisi matriks 130x130 digunakan sebagai pola input pada sistem pengenalan pola sidik jari

2.5 Deteksi Sidik Jari dalam *Image*

Tujuan dari deteksi sidik jari ialah untuk menentukan siapa nama pemilik suatu sidik jari tertentu yang akan dikenali. Masalah seperti ini merupakan suatu

tantangan karena *image* sidik jari antara individu satu dengan yang lainnya memiliki corak dan ciri yang beragam. Pengenalan atau identifikasi sidik jari akan memberikan informasi jika ada kecocokan antara *image* masukan dengan *image* di dalam *database*. Pengenalan sidik jari bertujuan untuk menentukan nama pemilik sidik jari tertentu, dengan demikian pengenalan sidik jari merupakan langkah awal dalam penggunaan aplikasi sidik jari lainnya diberbagai bidang.

2.6 Jaringan Saraf Tiruan

Jaringan saraf tiruan merupakan jaringan dengan elemen - elemen pemproses sederhana, yaitu masukan dan neuron yang saling terhubung, dan beroperasi secara paralel. Elemen pemroses berinteraksi melalui sambungan yang disebut bobot. Perancangan jaringan saraf tiruan diilhami oleh bagaimana otak manusia bekerja. Otak adalah jaringan yang terdiri dari sekitar 2,5 milyar *neuron*, saling dihubungkan melalui *axon* dan *dendrit*. *Sinapsis* merupakan titik hubung *axon* dan *dendrit* dari suatu *neuron* ke *neuron* lain.

Tujuan dari jaringan saraf tiruan adalah meniru *neuron* dalam otak manusia dengan cara saling menghubungkan elemen pemproses sederhana yang disebut *neuron* tiruan atau simpul (*node*). Sambungan atau hubungan yang disebut bobot, menirukan *sinapsis*. Peniruan tersebut menjadikan jaringan saraf tiruan sangat cocok untuk menyelesaikan masalah dengan tipe sama seperti otak manusia.

Sebelum suatu jaringan saraf tiruan digunakan untuk mengklasifikasikan suatu pola, terlebih dahulu dilakukan proses pembelajaran untuk menentukan struktur jaringan, terutama dalam penentuan nilai bobot.

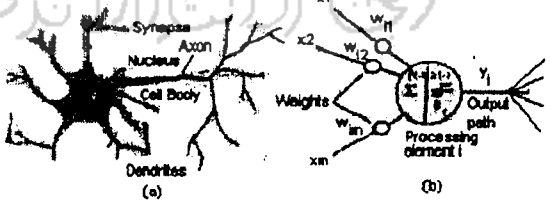
Jaringan saraf tiruan dikembangkan sebagai generalisasi model matematis dari saraf biologis, berdasarkan asumsi bahwa :

1. Pengenalan informasi terjadi pada beberapa elemen sederhana disebut *neuron*.
2. Isyarat yang lewat diantara *neuron-neuron* melalui rantai penghubung.
3. Tiap rantai penghubung mempunyai bobot penghubung yang dalam jaringan saraf merupakan pengali/penguat sinyal yang ditransmisikan.
4. Tiap *neuron* menerapkan suatu fungsi aktivasi ke masukan jaringan penjumlah bobot sinyal masukan untuk menentukan sinyal keluaran.

Suatu jaringan saraf tiruan dinyatakan oleh :

1. Pola - pola hubungan antar *neuron* yang disebut arsitektur jaringan saraf.
2. Metode penentuan bobot pada penghubung.
3. Fungsi aktivasi yang digunakan.

Sebuah jaringan saraf terdiri dari *neuron*, unit *sel* atau *node*. Tiap *neuron* dihubungkan ke *neuron* lain oleh suatu rantai penghubung yang mempunyai bobot. Bobot ini yang menunjukkan informasi yang digunakan oleh jaringan untuk memecahkan suatu permasalahan.



Gambar 2.7 (a)Sistem neuron biologis (b)Sistem neuron buatan

2.6.1 Pemodelan *Neuron*

Dalam usaha untuk menirukan kemampuan dari otak, pada tahun 1943, *Warren McCulloch dan Walter Pitts* menyusun model sederhana dari *neuron* yang

disebut model *McCulloch-Pitts*, terdiri dari banyak masukan dan 1 (satu) keluaran.

Gambar 2.8 memperlihatkan model dari *neuron* yang mempunyai persamaan :

$$y = f\left(\sum_{i=1}^{N} w_i x_i - \theta\right) \qquad (2.1)$$

Dengan :

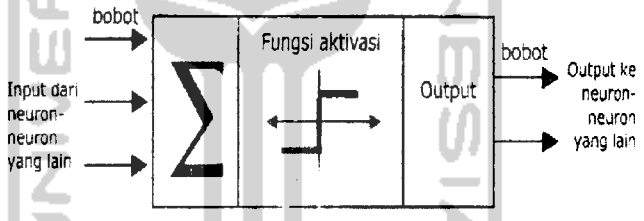
x_i = sinyal masukan, i = 1,2,3,...,N(N = banyaknya simpul masukan)

w_i = bobot hubungan atau *sinapsis*

θ = bias

f(.) = fungsi aktivasi

y = sinyal keluaran dari *neuron*



Gambar 2.8 Model *McCulloch - Pitts* untuk *neuron*

2.6.2 Arsitektur Jaringan Saraf Tiruan

Suatu algoritma belajar dari jaringan saraf tiruan tergantung dari arsitektur atau struktur dari jaringan saraf tersebut, dimana arsitektur tersebut menggambarkan jumlah *neuron* dan hubungan bobot yang dipakai. Secara umum arsitektur jaringan saraf tiruan dapat diklasifikasikan menjadi tiga kelas yang berbeda, sebagai berikut :

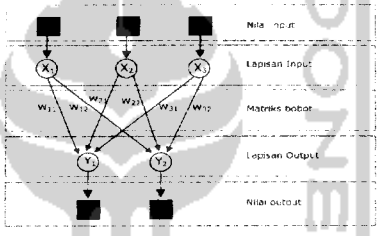
2.6.2.1 Jaringan Umpanmaju Lapis Tunggal (*Single Layer Feedforward Network*).

Perceptron disusun dari model *neuron McCulloch-Pitts* yang dihubungkan dan sering disebut jaringan umpan maju lapis tunggal (*single layer feedforward network*). Gambar 2.9 memperlihatkan *perceptron* yang mempunyai persamaan :

$$y_i = f\left(\sum_{1=1}^{N} w_{ji} x_j - \theta j\right) \qquad (2.2)$$

Dimana : i = 1,2,...,N (simpul masukan)

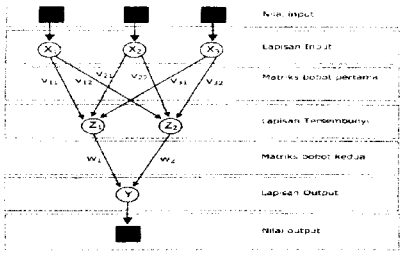
j = 1,2,...,N (simpul keluaran)



Gambar 2.9 Jaringan umpanmaju lapis tunggal

2.6.2.2 Jaringan Umpanmaju Lapis Banyak (*Multilayer Feedforward Network*)

Jaringan umpanmaju lapis banyak (*multilayer feedforward network*) disarankan oleh *Minsky* dan *Papert* untuk penyelesaian masalah yang lebih kompleks. Jaringan umpanmaju lapis banyak memperkenalkan satu atau lebih lapis tersembunyi (*hidden layer*) yang mempunyai simpul yang disebut *neuron*.



Gambar 2.10 Jaringan umpanmaju tiga lapis

2.6.2.3 Jaringan Saraf Tiruan Berulang (*Recurrent*)

Model jaringan saraf tiruan ini berbeda dengan model umpan maju karena mempunyai setidaknya ada satu *loop* umpan balik. Sebagai contoh, sebuah jaringan saraf tiruan berulang dapat terdiri dari *neuron* lapis tunggal dengan tiap *neuron* menggunakan sinyal *output*nya kembali ke *input* tiap *neuron* yang lain.

Pada struktur tersebut tidak ada *loop* umpan balik sendiri (*self - feedback*) dalam jaringan. Umpan balik sendiri terjadi jika *output* suatu *neuron* mengumpan balik ke *input*nya sendiri. Jaringan saraf tiruan berulang juga tidak mempunyai *hidden layer*.

2.6.3 Pengaturan Bobot

Metode pangaturan bobot (pelatihan jaringan) merupakan salah satu karakteristik pembeda dari beberapa jaringan lainnya. Secara umum ada dua macam metode pelatihan yaitu pelatihan terbimbing (*supervised training*) dan pelatihan tak terbimbing (*unsupervised training*).

2.6.3.1 Pelatihan Terbimbing (*Supervised Training*)

Pelatihan terbimbing merupakan suatu pelatihan jaringan dengan menghadirkan suatu deret vektor latihan atau pola - pola dan masing - masing dirangkai dengan suatu vektor target pada keluaran. Bobot kemudian disesuaikan atau diatur dengan algoritma belajar.

2.6.3.2 Pelatihan Tak Terbimbing (*Unsupervised Training*)

Jaringan saraf tiruan pengaturan sendiri (*self organizing*) mengelompokkan vektor masukan bersama tanpa menggunakan data latih menspesifikasikan ke kelompok mana suatu anggota itu tergolong. Suatu vektor

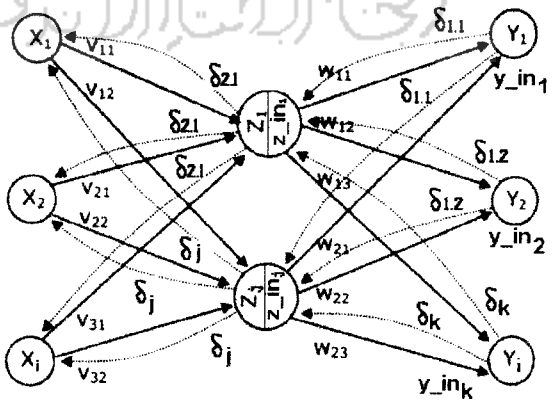
masukan dibuat tapi tanpa vektor target yang terspesifikasi. Jaringan memodifikasi bobot sehingga vektor masukan dibagikan ke unit keluaran serupa. Jaringan saraf tiruan akan menghasilkan suatu vektor contoh (*representative*) untuk tiap kelompok yang diberikan.

2.7 Perambatan Balik (*Backpropagation*)

Jaringan saraf tiruan perambatan - balik (*backpropagation*) dikembangkan oleh *Rumelhart, Hinton, dan Williams*. Prinsip dasar algoritma perambatan - balik memiliki tiga fase :

1. Fase *feedforward* pola input pembelajaran.
2. Fase kalkulasi dan *backpropagation error* yang didapat.
3. Fase penyesuaian bobot.

Arsitektur yang digunakan adalah jaringan lapis banyak (*multilayer perceptrons*). Arsitekturnya sendiri tersusun atas lapisan - lapisan yaitu lapisan masukan (*input layer*), lapis tersembunyi (*hidden layer*), dan lapis keluaran (*output layer*). Secara mendasar, proses dari *error backpropagation* ini terdiri atas dua tahap, yaitu umpan maju dan umpan mundur.



Gambar 2.11 Arsitektur jaringan saraf tiruan *backpropagation*

Lapisan paling kiri adalah lapisan masukan dan di dalam jaringan saraf tiruan hanya *neuron - neuron* pada lapis inilah yang menerima masukan dari luar. Lapisan berikutnya adalah lapisan tersembunyi, dan pada lapisan ini *neuron - neuron* diinterkoneksikan secara penuh ke lapisan di atas dan di bawahnya. Lapisan paling kanan adalah lapisan keluaran sebagai akhir dari proses jaringan tersebut.

2.8 Fungsi Aktivasi *Backpropagation*

1. Fungsi aktivasi untuk jaringan perambatan - balik harus memiliki beberapa karakteristik : kontinyu, dapat didiferensiasikan, dan monoton tidak turun.
2. Fungsi yang sering digunakan adalah :

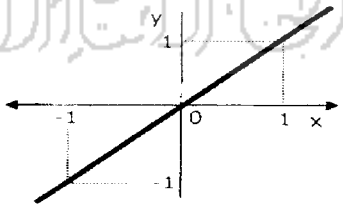
   Purelin atau fungsi linear (identitas), memiliki nilai *output* yang sama dengan nilai *input*

   a. Fungsi linear (identitas)

   $$y = f(x) = x \tag{2.3}$$

   Dengan turunannya,

   $$f'(x) = 1 \tag{2.4}$$



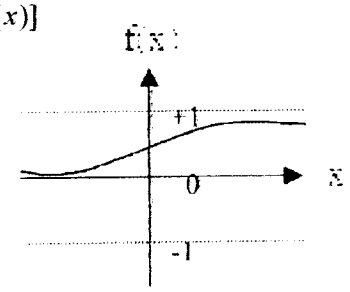
Gambar 2.12 Fungsi aktivasi *linear* identitas

   b. Fungsi *sigmoid biner* (*range* : [0,1])

   $$f(x) = \frac{1}{1+e^{-x}} \tag{2.5}$$

   Dengan turunannya,

$$f'(x) = f(x)[1 - f(x)] \tag{2.6}$$



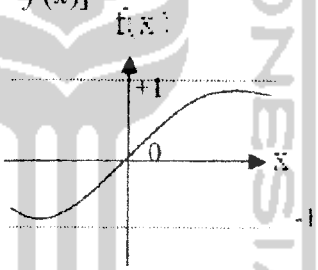
Gambar 2.13 Fungsi aktivasi *sigmoid biner*

c. Fungsi *sigmoid bipolar* (*range* : [-1,1])

$$f(x) = \frac{2}{1+e^{-x}} - 1 \tag{2.7}$$

Dengan turunannya,

$$f'(x) = \frac{1}{2}[1 + f(x)][1 - f(x)] \tag{2.8}$$



Gambar 2.14 Fungsi aktivasi *sigmoid bipolar*

BAB III

PERANCANGAN SISTEM

Perancangan sistem pengenalan sidik jari dengan algoritma belajar perambatan - balik merupakan tahapan penting dalam pembuatan simulasi jaringan saraf tiruan ini, dengan tujuan agar dalam pembuatan simulasinya berjalan secara sistematis, terstruktur, dan rapi sehingga hasil program dapat optimal dan sesuai dengan yang dikehendaki.

Suatu perancangan sistem diperlukan ragam dialog untuk memerlukan antarmuka yang digunakan sebagai media untuk berinteraksi dengan pengguna (*user*) sistem dengan komputer. Antarmuka yang menggunakan ragam dialog pada dasarnya adalah untuk mendapatkan satu kriteria yang sangat penting dalam pengoperasian sebuah program aplikasi, yakni aspek ramah penguna (*user friendly*).

Dalam perancangan, terdapat beberapa aspek yang harus diperhatikan meliputi efektifitas dan efisiensi program, kemampuan program, kemudahan untuk dipahami pengguna, dan kemungkinan pengembangan di masa depan.

Simulasi ini coba dirancang dengan tidak menutup kemungkinan untuk dikembangkan lebih lanjut nantinya. Untuk pengembangan sistem, pemodelan atau untuk perluasan fungsi dan penerapannya. Agar program mudah untuk dimengerti dan dipakai (*user friendly*) maka simulasi dirancang dengan perangkat lunak (*software*) yang bertampilan grafis (*Graphic User Interface*). Tampilan

grafis memudahkan pemakai untuk mengetahui cara kerja dan cara menggunakan program.

Bahasa pemprograman adalah perintah - perintah atau intruksi yang dimengerti oleh komputer untuk melakukan tugas - tugas tertentu. Bahasa yang digunakan untuk aplikasi ini adalah bahasa C dari software Matlab v.7.0.

3.1 Deskripsi Sistem

3.1.1 Perangkat Keras (*hardware*)

Perangkat keras yang dipakai untuk merancang dan menjalankan program simulasi dalam tugas akhir ini adalah sebuah *Notebook* dengan spesifikasi sebagai berikut :

1. *Notebook Toshiba Satellite M115*

   Sistem *Notebook* : *Intel Centrino Duo* ® CPU T2050 *Core Duo Processor* @1,60 GHz

   Sistem Operasi : *Microsoft Windows XP Media Center Edition*

   Media Tampilan : 14.1" WXGA TFT *Crystal*

   Media masukan : Papan ketik (*keyboard*) dan *mouse*

   Memori : 512 MB RAM

2. *Biopod Fingerprint Sensor*

3.1.2 Perangkat Lunak (*software*)

Perangkat lunak yang digunakan dalam tugas akhir ini adalah :

1. MATLAB v.7.0 : *Toolbox* matlab yang akan digunakan adalah *Image processing Toolbox* dan *Neural Network Toolbox*. *Image processing Toolbox* digunakan untuk menangani pengolahan *image*, sedangkan *Neural Network*

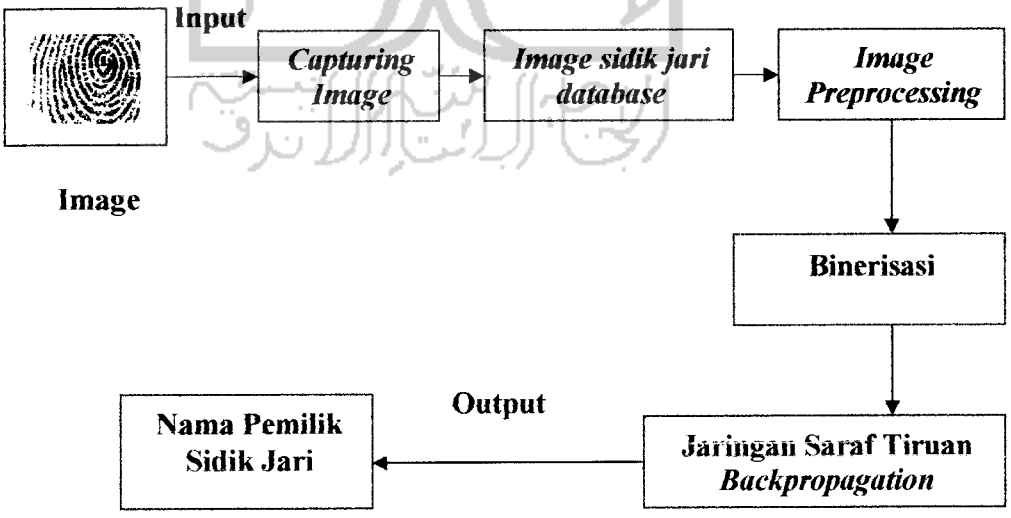
Toolbox digunakan untuk menangani jaringan saraf tiruan dengan menggunakan metode *backpropagation*, termasuk untuk pembuatan tampilan antarmuka pengguna grafis (GUI).

2. OmniPass Enrollment Wizard : digunakan untuk mengambil sampel sidik jari.
3. Adobe Photoshop 7.0 : digunakan sebagai program untuk melakukan proses pemotongan *image* sidik jari pada pengambilan awal objek, dimana *image* ini nantinya digunakan sebagai data untuk proses pelatihan jaringan.

3.2 Perancangan Sistem Pengenalan Pola

Perancangan sistem pengenalan pola ini dibagi menjadi empat tahapan utama, yaitu:

1. Mengambil *image* dari objek sidik jari yang telah disimpan dalam *database*.
2. Mengolah *image* tersebut menjadi sebuah *image* biner (*binerisasi*).
3. Membuat matriks dari *image* biner tersebut.
4. Melakukan pelatihan dan pengujian menggunakan jaringan saraf tiruan.

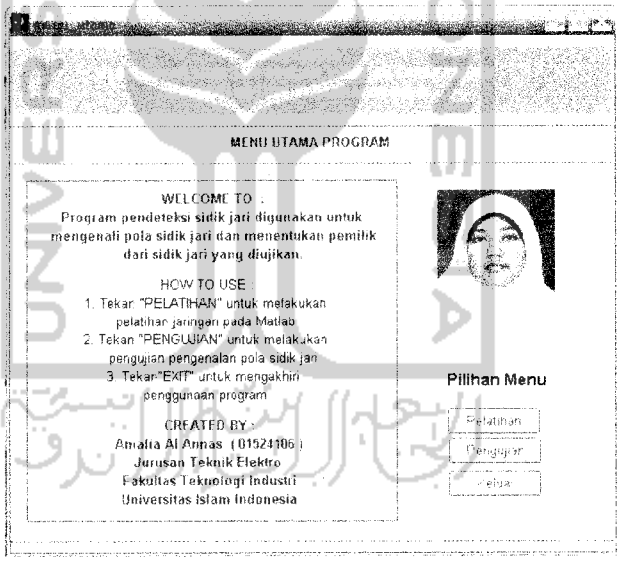


Gambar 3.1 Blok diagram sistem pengenalan pola sidik jari

Pada sistem pengenalan pola sidik jari ini dirancang tiga buah GUI (*Graphical User Interface*) yang terdiri atas *interface* menu utama program, *interface* pelatihan, dan *interface* pengujian,. Adapun ketiga *interface* tersebut dijelaskan sebagai berikut :

3.2.1 *Interface* Menu Utama Program

Interface menu utama program digunakan sebagai *interface* pembuka. Terdapat tombol - tombol untuk menampilkan *interface* - *interface* untuk pelatihan dan pengujian. Cara menampilkannya dengan menjalankan *run* pada *debug* yang ada pada masing-masing M-file programnya atau dapat dilakukan dengan GUI untuk *interface* utama di tunjukkan pada gambar di bawah ini :



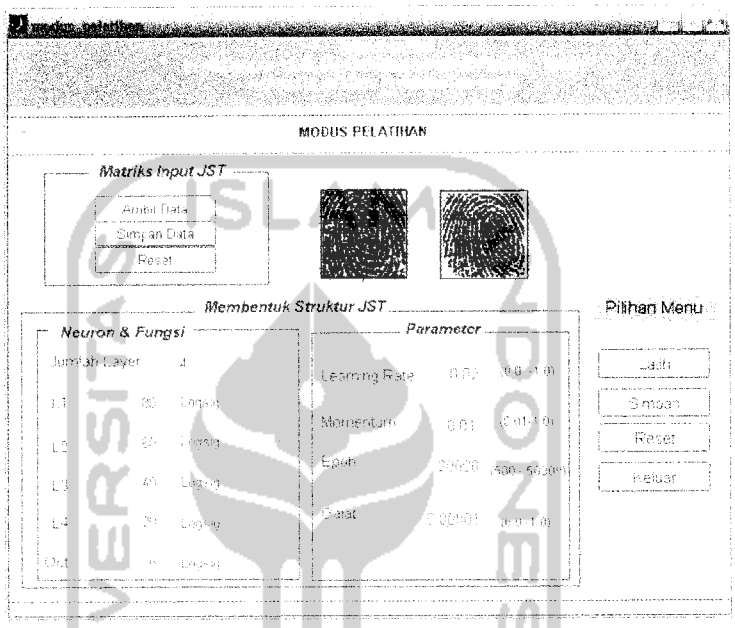
Gambar 3.2 *Interface* menu_utama sistem pengenalan pola sidik jari

Interface menu utama program terdiri atas:

- Tombol "Pelatihan" digunakan untuk memanggil *interface* pelatihan.
- Tombol "Pengujian" digunakan untuk memanggil *interface* pengujian.
- Tombol "Keluar" untuk selesai menggunakan program.

3.2.2 *Interface* Pelatihan

Pada *interface* pelatihan di dalamnya terdapat proses pengolahan *image* yang digunakan untuk mengubah *image* sidik jari menjadi *image* biner yang berisi matriks *image* sidik jari 130x130 piksel.



Gambar 3.3 *Interface* modus_pelatihan sistem pengenalan pola sidik jari

Interface pelatihan untuk pengolahan *image* terdiri atas :

- Tombol "Ambil Data" digunakan untuk mengambil objek sidik jari dari *database* komputer.
- Tombol "Simpan Data" digunakan untuk menyimpan *image*.
- Tombol "*Reset*" digunakan untuk mereset kedua *axes*.

Interface pelatihan digunakan untuk melakukan pelatihan jaringan saraf tiruan yang akan digunakan pada pengujian.

Interface pelatihan untuk pembentukan struktur JST terdiri atas :

- Tombol "Latih" digunakan untuk melakukan proses pelatihan.

- Tombol "Simpan" digunakan untuk menyimpan hasil pelatihan.
- Tombol "*Reset*" digunakan untuk mereset parameter jaringan.
- Tombol "Keluar" digunakan untuk selesai melakukan pelatihan.

Proses - proses yang terjadi dalam pelatihan adalah sebagai berikut :

a. Mengambil data - data pola *input* dan target yang sudah disimpan.

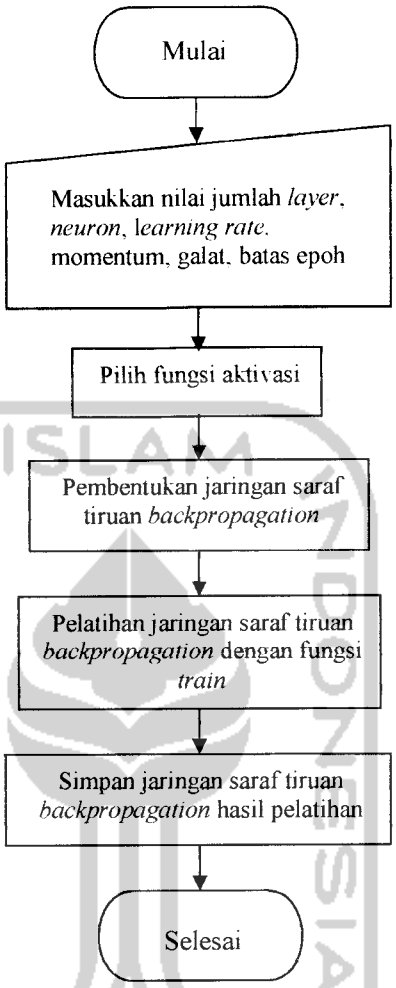
```
DataLea = load('D:\Program DeSidJar\Form GUI
Matlab\MatrixJari_1.mat');
```

b. Melakukan pelatihan jaringan berdasarkan nilai yang diisikan ke dalam kolom parameter jaringan saraf tiruan.

```
Net = newff(minmax(Src), Neuron, [SetF OutF],
   'traingdx');
Src = double(Src);
Des = DestLea.Des;
Net = init(Net);
Net = train(Net, Src, Des);
```

c. Menyimpan jaringan yang sudah dilatihkan

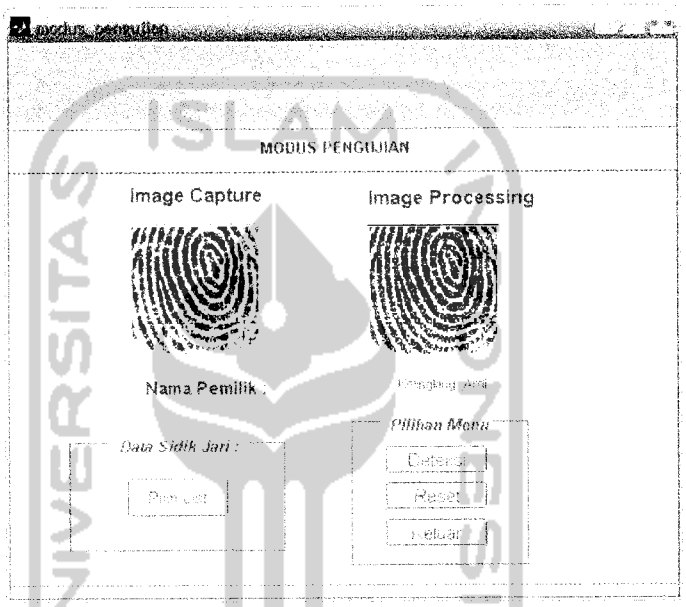
```
save('D:\Program DeSidJar\Form GUI Matlab
\VnetJari_1.mat','Net');
```



Gambar 3.4 Diagram alir (*flow chart*) proses pelatihan jaringan saraf tiruan *backpropagation* menggunakan *interface* pelatihan

3.2.3 *Interface* Pengujian

Interface pengujian digunakan untuk melakukan pengujian pada sistem pengenalan pola menggunakan jaringan saraf tiruan hasil pelatihan dari *interface* pelatihan.



Gambar 3.5 *Interface* modus_pengujian sistem pengenalan pola sidik jari

Interface pengujian terdiri atas :

- Tombol "Pilih List" digunakan untuk meng-*capture* objek yang akan dikenali yaitu data sidik jari yang sudah tersimpan dalam *database* dan menampilkannya pada *axes*1.
- Tombol "Deteksi' digunakan untuk melakukan pengujian terhadap *input* matriks hasil proses binerisasi sehingga diketahui nama pemiliknya.
- Tombol "*Reset*" digunakan untuk mereset kedua axes dan text.
- Tombol "Keluar" digunakan untuk selesai melakukan pengujian.

Proses - proses yang terjadi dalam pengujian adalah sebagai berikut:

a. Proses pengolahan citra

- Mengubah tipe citra menjadi *grayscale image*

```
image = rgb2gray(image);
```

- Menambah kontras citra

```
image = imadjust(image);
```

- Memisahkan objek dari benda

```
BW  = imclose(BW,SE);
```

- Mengubah citra menjadi citra biner

```
I = im2bw(I,graythresh(I));
```

- Membersihkan lubang pada citra

```
B   = bwboundaries(BW,'noholes');
```

- Membuang objek dalam citra yang lebih kecil dari 50 piksel

```
Are = bwarea(BW)/2;
BW  = bwareaopen(BW,round(Are));
```

- Mengisi lubang pada citra

```
BW  = imfill(BW,'holes');
```

- Proses filter multidimensional dimana nilai - nilai kecil dibulatkan sedangkan bila melebihi dipotong.

```
FltType = fspecial('unsharp');
I       = imfilter(I,FltType);
```

- Melaksanakan persamaan histogram adaptip *contrast-limited* (CLAHE)

```
I = adapthisteq(I);
```

- Mengubah ukuran citra *image* menjadi 130x130 piksel

```
I = imresize(I, [130 130], 'bicubic');
```

b. Proses pengujian

- Mengambil data - data pola *input* dan target yang sudah disimpan.
- Mengambil jaringan hasil pelatihan *interface* pelatihan yang sudah disimpan.

```
Network = load('D:\ Program DeSidJar\Form GUI
Matlab\VnetJari_1.mat');
```

- Mengubah citra hasil pengolahan citra (*image preprocessing*) menjadi sebuah mariks 130x130.

```
IProc = reshape(Iproc,16900,1);
```

- Mengubah matrik kebentuk *cell*

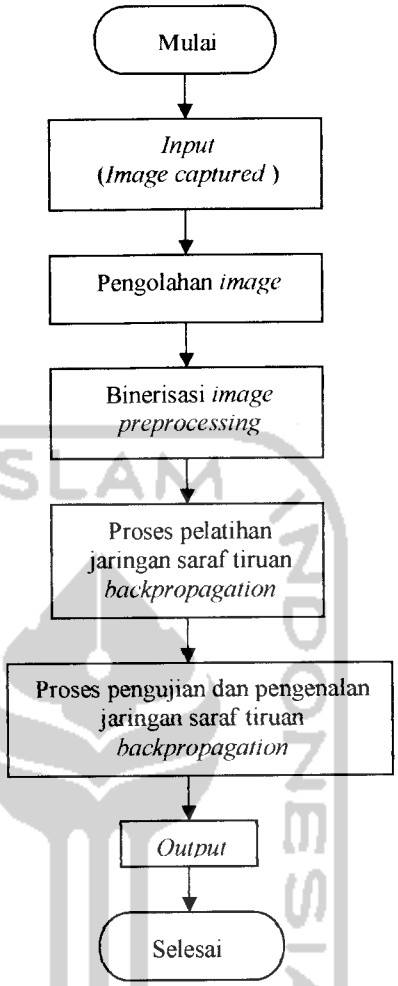
```
IProc = num2cell(IProc,1);
```

- Mengujikan matrik *input* baru ke jaringan.

```
DestLea = load('D:\ Program DeSidJar\Form GUI
Matlab\MDestJari_1.mat');
```

- Melakukan pengklasifikasian dengan mencocokkan *output* jaringan dengan target yang sudah ditentukan

```
MatDest = DestLea.Des;
pos = 0;
Str = 'Tidak Dikenal';
```



Gambar 3.6 Diagram alir (*flow chart*) proses pengenalan pola sidik jari

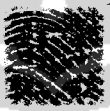
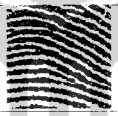
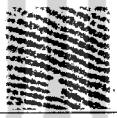
3.3 Objek Yang Digunakan

Sistem pengenalan pola akan mengenali limapuluh objek yaitu dari 10 orang yang diambil masing - masing lima *image* sidik jari tangan kanan dengan berbagai karakteristik yang dimiliki tiap - tiap individu. Pada tugas akhir ini kelimapuluh objek tersebut akan digunakan untuk pelatihan jaringan saraf tiruan.

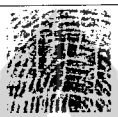
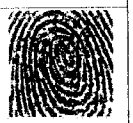
Setelah *image* objek sidik jari di-*capture* oleh *biopod fingerprint sensor* secara tidak langsung, *image* di*edit* dengan resolusi 130x130 *pixel* dan *image* hasil

capture tersebut akan disimpan dengan format *JPG*, yaitu format *default* gambar. Objek - objek tersebut ditunjukkan pada tabel berikut :

Tabel 3.1 Objek yang akan digunakan dalam program pengenalan pola sidik jari

No.	Nama Objek	*Image* Objek	No.	Nama Objek	*Image* Objek
1.	Jempol_April		26.	Manis_Linda	
2.	Jempol_Ardi		27.	Manis_Mufi	
3.	Jempol_Dewi		28.	Manis_Ratih	
4.	Jempol_Erna		29.	Manis_Rendy	
5.	Jempol_Fitri		30.	Manis_Seli	
6.	Jempol_Linda		31.	Telunjuk_April	
7.	Jempol_Mufi		32.	Telunjuk_Ardi	
8.	Jempol_Ratih		33.	Telunjuk_Dewi	
9.	Jempol_Rendy		34.	Telunjuk_Erna	
10.	Jempol_Seli		35.	Telunjuk_Fitri	
11.	Kelingking_April		36.	Telunjuk_Linda	

Lanjutan Tabel 3.1

12.	Kelingking_Ardi		37.	Telunjuk_Mufi	
13.	Kelingking_Dewi		38.	Telunjuk_Ratih	
14.	Kelingking_Erna		39.	Telunjuk_Rendy	
15.	Kelingking_Fitri		40.	Telunjuk_Seli	
16.	Kelingking_Linda		41.	Tengah_April	
17.	Kelingking_Mufi		42.	Tengah_Ardi	
18.	Kelingking_Ratih		43.	Tengah_Dewi	
19.	Kelingking_Rendy		44.	Tengah_Erna	
20.	Kelingking_Seli		45.	Tengah_Fitri	
21.	Manis_April		46.	Tengah_Linda	
22.	Manis_Ardi		47.	Tengah_Mufi	
23.	Manis_Dewi		48.	Tengah_Ratih	
24.	Manis_Erna		49.	Tengah_Rendy	

Lanjutan Tabel 3.1

25.	Manis_Fitri		50.	Tengah_Seli	

3.4 Membangun Jaringan Saraf Tiruan *Backpropagation*

Arsitektur jaringan saraf tiruan yang sering digunakan oleh algoritma *backpropagation* adalah jaringan dengan banyak lapisan. Pada sistem pengenalan pola, jaringan saraf tiruan dibangun dengan jaringan *feedforward* yaitu dengan menggunakan instruksi "*newff*". Struktur jaringan saraf tiruan *backpropagation* dengan *newff* adalah sebagai berikut:

net = newff (PR, [S1 S2 ....SN1], {TF1 TF2...TFN1}, BTF)

Keterangan:

PR : matriks berukuran Rx2 yang berisi nilai minimum dan maksimum, dengan R adalah jumlah variable *input*.

Si : jumlah *neuron* pada lapisan ke-i, dengan I = 1, 2,...N1.

TFi : fungsi aktivasi pada lapisan ke-i, dengan I = 1, 2,....N1; (*default: tansig*)

BTF : fungsi pelatihan jaringan (*default : trainlm*).

Beberapa parameter yang ditentukan sebelumnya melatih jaringan antara lain yaitu target *error* (*goal*), dan target keluaran. Sedangkan banyaknya epoh, besarnya *learningrate* dan momentum ditentukan oleh *user* program.

Pada perancangan sistem ini, *input* jaringan saraf tiruan berasal dari matriks 130x130 *image* biner yang merupakan hasil dari pengolahan *image*. Target pelatihan yang digunakan sebanyak 50 buah berupa bilangan biner 6 bit ($2^6 = 64$) dengan pola sebagai berikut :

Tabel 3.2 Pola *input* dan pola target pelatihan

No	Objek	*Input* pola	Target
1.	Jempol_April	00000001000001110001111011111....	000001
2.	Jempol_Ardi	0000001100001111001111111111111...	000010
3.	Jempol_Dewi	1111111111111111011111111001111...	000011
4.	Jempol_Erna	000000010000001100001111000111....	000100
5.	Jempol_Fitri	11111111111111111111111111011111....	000101
6.	Jempol_Linda	000011110001111101111111111111....	000110
7.	Jempol_Mufi	0000001000000111000111111001111...	000111
8.	Jempol_Ratih	11000000111100001111100011111....	001000
9.	Jempol_Rendy	0000000100001111000111111011111...	001001
10.	Jempol_Seli	000000100001111001111111111111...	001010
11.	Kelingking_April	000000110000011100011111001111....	001011
12.	Kelingking_Ardi	0000001000001110001111111011111...	001100
13.	Kelingking_Dewi	0000000000011000000110000001111...	001101
14.	Kelingking_Erna	1111111101111110011111110001111...	001110
15.	Kelingking_Fitri	0000001000000111000111111001111...	001111
16.	Kelingking_Linda	000000010000111100111111111111...	010000
17.	Kelingking_Mufi	0000000100001111000111111001111...	010001
18.	Kelingking_Ratih	0011110001111111011111111111111...	010010
19.	Kelingking_Rendy	001111000011111001111110111111...	010011
20.	Kelingking_Seli	011111101111111111111111111111...	010100
21.	Manis_April	000011110011111101111111111111...	010101
22.	Manis_Ardi	111110001111110111111110111111...	010110
23.	Manis_Dewi	000110000111111001111110111111...	010111
24.	Manis_Erna	111111111111111111111111111111...	011000
25.	Manis_Fitri	111100001111111111111111111111...	011001
26.	Manis_Linda	111111111111111111111111111111...	011010
27.	Manis_Mufi	011111110111111111111111111111...	011011
28.	Manis_Ratih	111111101111111011111111111111...	011100
29.	Manis_Rendy	000111111111111111111111111111...	011101
30.	Manis_Seli	111111110111111001111110001111...	011110
31.	Telunjuk_April	001111000011111001111110111111...	011111
32.	Telunjuk_Ardi	111111110111111001111110001111...	100000
33.	Telunjuk_Dewi	001111000011111001111110111111...	100001
34.	Telunjuk_Erna	000000010000111100011111001111...	100010
35.	Telunjuk_Fitri	001111000011111001111110111111...	100011
36.	Telunjuk_Linda	111111110111111001111110001111...	100100
37.	Telunjuk_Mufi	000000010000111100011111001111...	100101
38.	Telunjuk_Ratih	000011110011111101111111111111...	100110
39.	Telunjuk_Rendy	000011110011111101111111111111...	100111
40.	Telunjuk_Seli	000000010000111100011111001111...	101000

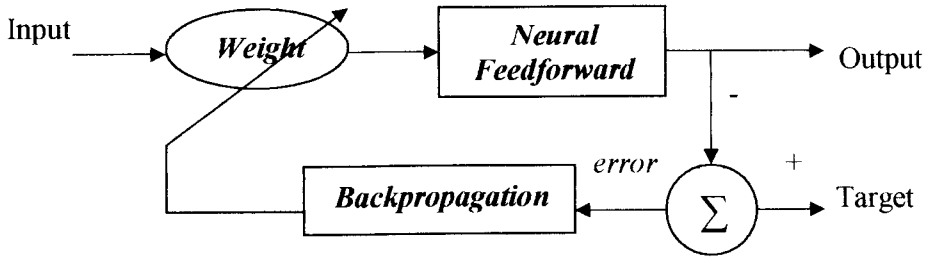
Lanjutan Tabel 3.2

41.	Tengah_April	00000001000011110001111100111...	101001
42.	Tengah_Ardi	00001111001111110111111111111...	101010
43.	Tengah_Dewi	00001111001111110111111111111...	101011
44.	Tengah_Erna	00000001000011110001111100111...	101100
45.	Tengah_Fitri	00111100011111101111111111111...	101101
46.	Tengah_Linda	00111100011111101111111111111...	101110
47.	Tengah_Mufi	11111111011111100111111000111...	101111
48.	Tengah_Ratih	00000001000011110001111100111...	110000
49.	Tengah_Rendy	00111100011111101111111111111...	110001
50.	Tengah_Seli	11111111111111111111111111111...	110010

Struktur jaringan saraf yang digunakan pada sistem pengenalan pola ini adalah jaringan saraf tiruan *backpropagation* dengan empat lapisan tersembunyi dengan susunan 80, 60, 40, 20 untuk setiap lapisan tersembunyi.

3.5 Pelatihan Jaringan Saraf Tiruan *Backpropagation*

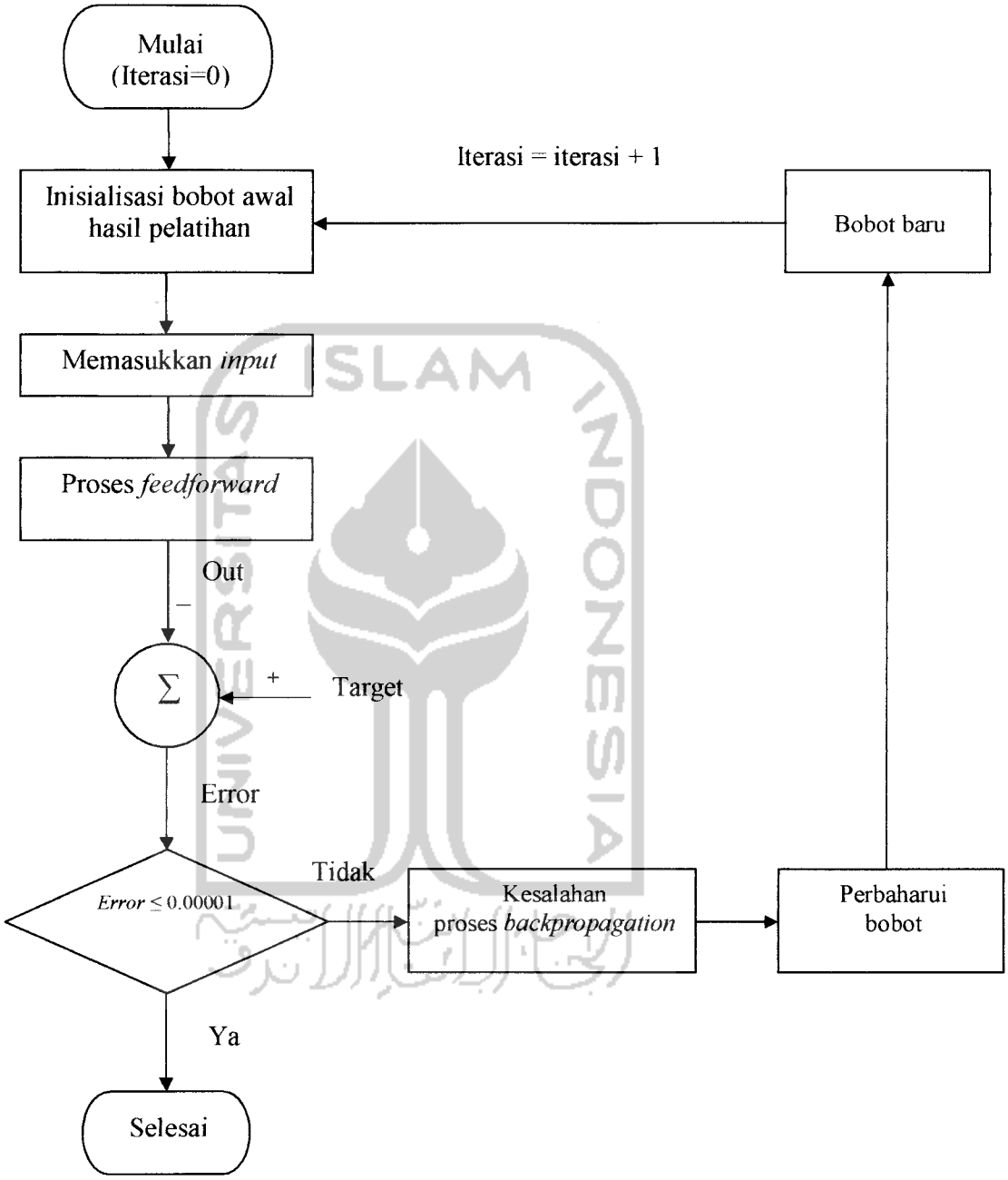
Jaringan saraf tiruan *backpropagation* merupakan jaringan saraf tiruan yang menggunakan algoritma pembelajaran terawasi (*supervised learning)* untuk mengubah bobot - bobot yang terhubung dengan *neuron - neuron* yang ada pada lapisan tersembunyi. Algoritma *backpropagation* menggunakan *error output* untuk mengubah nilai bobot - bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* ini, tahap *feedforward* harus dikerjakan terlebih dahulu. Pada saat *feedforward neuron - neuron* diaktifkan dengan menggunakan fungsi aktivasi. Blok diagram pembelajaran jaringan saraf tiruan *backpropagation* yang digunakan untuk pengujian di tunjukkan pada gambar 3.7 :



Gambar 3.7 Blok diagram pelatihan jaringan saraf tiruan *backpropagation*

Pada perancangan sistem ini, *input* jaringan saraf tiruan berasal dari proses binerisasi *image* biner yang dihasilkan oleh proses pengolahan *image*. Data berupa matriks 130x130. Target pelatihan yang digunakan sebanyak 50 buah yang masing - masing merupakan bilangan biner.

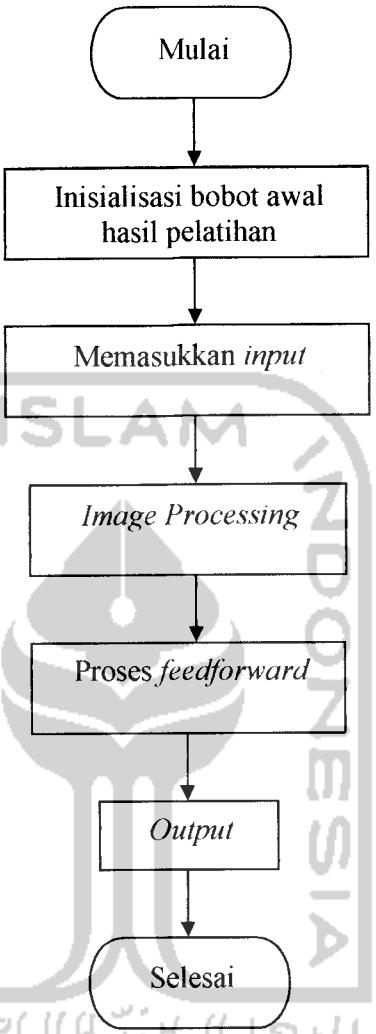
Pelatihan dilakukan dengan meng-*capture* beberapa contoh *image* sidik jari dari objek yang akan dikenali. Pola *image* dari masing - masing objek tersebut di simpan dan diproses terlebih dahulu sehingga didapatkan data *image* biner dan kemudian dilatihkan secara bersamaan ke dalam jaringan saraf tiruan *backpropagation*. Pola *image* biner dimasukkan secara urut berdasarkan urutan pada program Matlab, dimulai dari angka terkecil. Setelah semua data dimasukkan maka proses pelatihan dilakukan sampai *error* yang dihasilkan mencapai nilai yang telah ditentukan yaitu 0.00001. Untuk pelatihan jaringan saraf tiruan *backpropagation* dilakukan dengan menggunakan perintah "*train (net, input, target)*" yaitu dengan menekan tombol "Pelatihan" pada *interface* menu utama kemudian mengisi parameter - parameter jaringan saraf tiruan dan tekan tombol "Latih".



Gambar 3.8 Diagram alir (*flow chart*) pelatihan jaringan saraf tiruan *backpropagation* dengan satu lapisan tersembunyi

3.6 Pengujian Jaringan Saraf Tiruan *Backpropagation*

Proses pengujian pada sistem pengenalan pola sidik jari dilakukan dengan menekan tombol "Pengujian". Pengujian pada tugas akhir ini merupakan pengujian dengan meng-*capture image* sidik jari yang telah tersimpan dalam *database* menggunakan *biopod fingerprint sensor* secara tidak langsung. Kemudian dilakukan *image processing* yaitu pengolahan *image* dengan mengubah image menjadi abu - abu serta perbaikan kualitas *image* hasil *capture dengan cara pengontrasan dan pemfilteran*. Proses pemotongan *image* (*crop*) dilakukan dengan cara memisahkan objek dari *background* dan mengubah ukuran *image* menjadi 130 x 130 piksel untuk selanjutnya diubah menjadi *image black white*. Pengujian jaringan dilakukan dengan menggunakan perintah "*sim(net, input)*". Selanjutnya proses *testing* dilakukan untuk mensimulasikan *input* dengan target. Proses akhir yaitu *checking* untuk menguji hasil simulasi. Dan proses pengenalan pola ini akan langsung dilakukan pada saat menekan tombol "Deteksi".



Gambar 3.9 Diagram alir (*flow chart*) pengujian jaringan saraf tiruan *backpropagation* dengan satu lapisan tersembunyi

BAB IV

ANALISIS DAN PEMBAHASAN

Pada bab ini akan dilakukan pengujian dan analisis penerapan jaringan saraf tiruan pada deteksi sidik jari dengan alur program yang telah dijelaskan pada bab sebelumnya. Pengujian terhadap program simulasi ini dilakukan dengan tujuan agar dapat diketahui apakah program yang dibuat telah sesuai dengan yang dinginkan dengan parameter unjuk kerja diukur dari *prosentase* hasil pendeteksian sidik jari yang diujikan.

Setelah membuat model jaringan saraf tiruan yang akan digunakan dalam proses pendeteksian sidik jari, maka selanjutnya dilakukan suatu pengujian model perancangan agar tujuan dari tugas akhir ini dapat terpenuhi dan hasil dari pengujian tersebut dapat dianalisis sehingga dapat diambil kesimpulan.

4.1 Hasil Pelatihan Jaringan Saraf Tiruan *Backpropagation*

Pada pelatihan jaringan saraf tiruan *backpropagation* untuk pengenalan pola sidik jari dilakukan beberapa pengamatan yaitu pengamatan pada pengaruh jumlah lapisan tersembunyi, fungsi aktivasi, nilai *learning rate* serta momentum yang digunakan. Hasil proses pelatihan yang dilakukan pada jaringan saraf tiruan *backpropagation* pada sistem pengenalan pola sidik jari secara keseluruhan dilakukan sebanyak 50 kali percobaan yang dapat dilihat pada tabel 4.1 sebagai berikut :

Tabel 4.1 Hasil pelatihan jaringan saraf tiruan pada sistem pengenalan pola sidik jari

No	HL	Jumlah Neuron	LR (α)	MC (β)	F.Aktivasi	Iterasi	MSE	Ket
1.	1	20	0.1	0.9	*logsig-logsig*	421	0.00098	Berhasil
2.	1	20	0.5	0.7	*logsig-logsig*	1554	0.00099	Berhasil
3.	1	20	0.1	0.9	*logsig-purelin*	20000	0.00010	Berhasil
4.	1	20	0.2	0.8	*logsig-purelin*	1909	0.00100	Berhasil
5.	1	20	0.3	0.5	*logsig-tansig*	2000	0.20221	Gagal
6.	1	20	0.5	0.7	*logsig-tansig*	2000	0.18823	Gagal
7.	1	20	0.2	0.8	*purelin-logsig*	152	0.41667	Gagal
8.	1	20	0.8	0.9	*purelin-logsig*	55	0.46000	Gagal
9.	1	20	0.6	0.7	*purelin-purelin*	2000	0.14588	Gagal
10.	1	20	0.7	0.9	*purelin-purelin*	2000	0.12653	Gagal
11.	1	20	0.2	0.9	*purelin-tansig*	22	0.54666	Gagal
12.	1	20	0.7	0.7	*purelin-tansig*	5	0.72000	Gagal
13.	1	20	0.4	0.7	*tansig-logsig*	1577	0.00010	Berhasil
14.	1	20	0.6	0.5	*tansig-logsig*	2327	0.00010	Berhasil
15.	1	20	0.5	0.8	*tansig-purelin*	3046	0.00100	Berhasil
16.	1	20	0.7	0.6	*tansig-purelin*	3244	0.00100	Berhasil
17.	1	20	0.4	0.7	*tansig-tansig*	2000	0.22010	Gagal
18.	1	20	0.5	0.9	*tansig-tansig*	2000	0.21061	Gagal
19.	2	40-20	0.2	0.6	*logsig-logsig-logsig*	489	0.00099	Berhasil
20.	2	40-20	0.4	0.8	*logsig-logsig-logsig*	542	0.00099	Berhasil
21.	2	40-20	0.5	0.7	*purelin-purelin-purelin*	2000	1.21482	Gagal
22.	2	40-20	0.6	0.8	*purelin-purelin-purelin*	2000	0.52487	Gagal
23.	2	40-20	0.1	0.9	*tansig-tansig-tansig*	2000	0.11076	Gagal
24.	2	40-20	0.4	0.8	*tansig-tansig-tansig*	2000	0.11962	Gagal
25.	2	40-20	0.1	0.9	*logsig-logsig-purelin*	689	0.00099	Berhasil
26.	2	40-20	0.1	0.5	*logsig-logsig-purelin*	6383	0.00100	Berhasil
27.	2	40-20	0.1	0.5	*logsig-purelin-purelin*	1008	0.00099	Berhasil
28.	2	40-20	0.5	0.9	*logsig-purelin-purelin*	644	0.00099	Berhasil
29.	2	40-20	0.1	0.9	*purelin-logsig-logsig*	2000	0.09594	Gagal
30.	2	40-20	0.5	0.9	*purelin-logsig-logsig*	2000	0.11254	Gagal
31.	2	40-20	0.1	0.9	*purelin-purelin-logsig*	31	0.47333	Gagal
32.	2	40-20	0.5	0.9	*purelin-purelin-logsig*	57	0.44667	Gagal
33.	2	40-20	0.1	0.9	*logsig-logsig-tansig*	2000	0.00457	Gagal
34.	2	40-20	0.5	0.9	*logsig-logsig-tansig*	2000	0.04295	Gagal
35.	2	40-20	0.1	0.9	*logsig-tansig-tansig*	2000	0.00155	Gagal
36.	2	40-20	0.5	0.9	*logsig-tansig-tansig*	2000	0.07644	Gagal
37.	2	40-20	0.1	0.9	*tansig-logsig-logsig*	2000	0.00254	Gagal
38.	2	40-20	0.5	0.9	*tansig-logsig-logsig*	2000	0.01525	Gagal
39.	2	40-20	0.1	0.9	*tansig-tansig-logsig*	393	0.00099	Berhasil
40.	2	40-20	0.5	0.9	*tansig-tansig-logsig*	1092	0.00100	Berhasil
41.	3	60-40-20	0.02	0.01	*logsig-logsig-logsig-logsig*	1809	0.00010	Berhasil
42.	3	60-40-20	0.02	0.01	*purelin-purelin-purelin-purelin*	10000	4.31023	Gagal
43.	3	60-40-20	0.02	0.01	*tansig-tansig-tansig-tansig*	10000	0.00037	Gagal

Lanjutan Tabel 4.1

44.	3	60-40-20	0.02	0.01	*purelin-logsig-purelin-logsig*	4745	0.06333	Gagal
45.	3	60-40-20	0.02	0.01	*logsig-logsig-tansig-tansig*	10000	0.00022	Gagal
46.	**4**	**80-60-40-20**	**0.02**	**0.01**	***logsig-logsig-logsig-logsig-logsig***	**2645**	**0.00001**	**Berhasil**
47.	4	80- 60-40-20	0.02	0.01	*purelin-purelin-purelin-purelin-purelin*	20000	0.00689	Gagal
48.	4	80- 60-40-20	0.02	0.01	*tansig-tansig-tansig-tansig-tansig*	20000	0.00012	Gagal
49.	4	80- 60-40-20	0.02	0.01	*logsig-logsig-purelin-purelin-logsig*	20000	0.10333	Gagal
50.	4	80- 60-40-20	0.02	0.01	*logsig-logsig-tansig-tansig-logsig*	2846	0.00001	Berhasil

Penjelasan tabel hasil pelatihan :

No : Nomor.

HL : Jumlah lapisan tersembunyi (*hidden layer*) .

Jumlah Neuron : Banyaknya neuron pada lapisan tersembunyi.

LR (α) : Nilai kecepatan pelatihan (*Learning Rate*).

MC (β) : Nilai koefisien momentum (*Momentum Coefisient*).

F. Aktivasi : Fungsi aktivasi pada *hidden layer* dan *output layer*.

Iterasi : Banyaknya iterasi yang dicapai (*Epoch*).

MSE : Nilai rata - rata *error* kuadrat (*Mean Square Error*).

Ket : Keterangan hasil pelatihan yaitu Berhasil atau Gagal.

"Berhasil" berarti *performance goal met* atau proses pelatihan mencapai error yang telah ditetapkan sedangkan "Gagal" berarti *performance goal was not met* atau proses pelatihan tidak mencapai *error* yang telah ditetapkan".

Cetak tebal : Hasil pelatihan terbaik dan akan digunakan sebagai struktur jaringan saraf tiruan pada sistem pengenalan pola sidik jari.

Grafik hasil pengamatan pelatihan dengan satu atau lebih lapisan tersembunyi (*hidden layer*) dapat dilihat pada lampiran data hasil pelatihan.

4.1.1 Pelatihan Satu Lapisan Tersembunyi

Pelatihan dengan menggunakan 1 lapisan tersembuyi berarti menggunakan 1 lapisan tersembunyi (*hidden layer*) dengan jumlah neuron tertentu pada proses pelatihan jaringan, dan dikelompokan ke dalam tabel - tabel sesuai fungsi aktivasi yang digunakan.

4.1.2 Pelatihan Lebih Dari 1 Lapisan Tersembunyi

Pelatihan dengan menggunakan lebih dari 1 lapisan tersembuyi berarti menggunakan 2 atau lebih sampai dengan 4 lapisan tersembunyi (*hidden layer*) dengan jumlah neuron tertentu pada proses pelatihan jaringan, dan dikelompokan ke dalam tabel - tabel sesuai fungsi aktivasi yang digunakan dan jumlah lapisan tersembunyi.

Dari hasil pelatihan, terdapat beberapa hasil pelatihan yang "Gagal" karena *minimum gradient* telah tercapai atau iterasi maksimum telah tercapai sedangkan *error* belum tercapai. Hal ini dapat disebabkan oleh bentuk fungsi aktivasi yang digunakan ataupun nilai *learning rate* serta *momentum*.

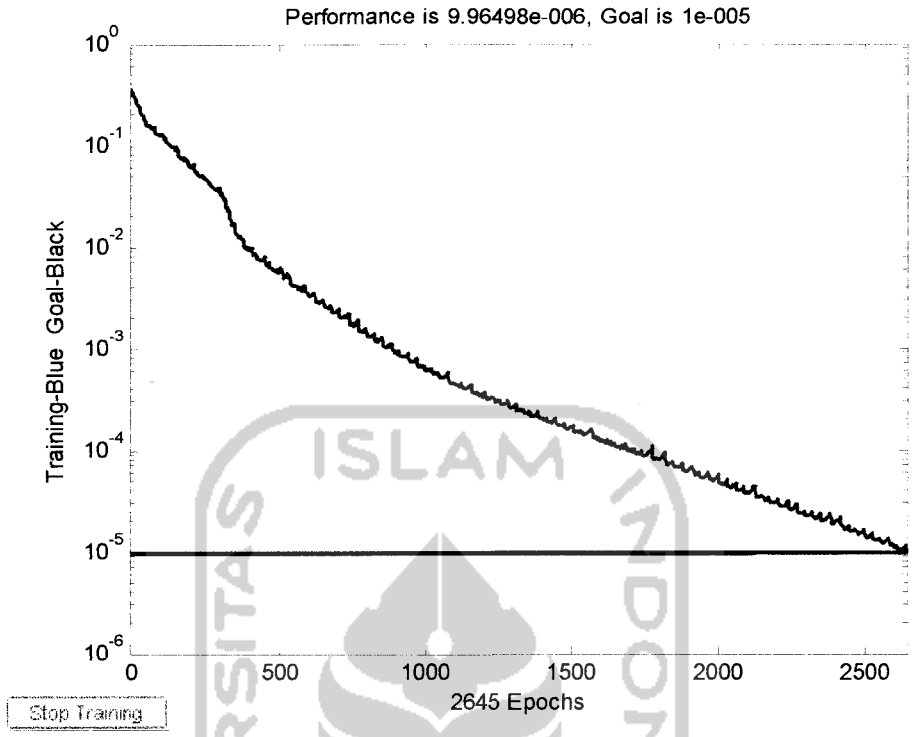
Fungsi aktivasi purelin yang digunakan dalam pelatihan dengan menggunakan 1 lapisan tersembunyi dapat mencapai target *error* meskipun dengan jumlah iterasi yang paling banyak tetapi dengan 2 lapisan tersembunyi, target *error* tidak tercapai. Hal ini disebabkan output dari *purelin* sifatnya *linear* sedangkan pada jaringan saraf tiruan *backpropagation* dengan lapisan

tersembunyi membuat jaringan memiliki sifat tidak *linear* karena terjadi proses komputasi. Sehingga semakin banyak jumlah lapisan tersembunyi maka hasil pelatihan akan semakin tidak *linear*.

Pada pelatihan menggunakan fungsi aktivasi *logsig* dimana hasil pelatihan hampir keseluruhan berhasil. Hal ini karena fungsi aktivasi *logsig* mempunyai keluaran antara 0 dan 1 sehingga dapat mendekati target keluaran yang telah ditentukan. Sedangkan kegagalan pada pelatihan dengan fungsi aktivasi *logsig* karena *minimum gradient* telah tercapai disebabkan oleh nilai *learning rate* dan *momentum* yang digunakan tidak optimum.

Pada pelatihan menggunakan fungsi aktivasi *tansig*, pelatihan dengan menggunakan 1 lapisan tersembunyi dan 2 lapisan tersembunyi dapat mencapai target *error* meskipun dengan jumlah iterasi yang banyak atau bahkan tidak dapat mencapai target *error*, hal ini disebabkan oleh nilai output dari fungsi aktivasi *tansig* yaitu antara 1 dan -1 sedangkan pada pelatihan ini digunakan target output yang harus dicapai adalah pola biner 0 dan 1.

Berdasarkan hasil pengamatan proses pelatihan jaringan, diperoleh struktur jaringan yang terbaik untuk proses pengenalan pola sidik jari yaitu 4 lapisan tersembunyi yang jumlah *neuron* lapisan tersembunyi 80, 60, 40 dan 20 dengan nilai *learning rate* 0.02 dan *momentum* 0,01 serta fungsi aktivasi *logsig-logsig-logsig-logsig*. Grafik pelatihannya dapat dilihat berikut ini :



Gambar 4.1 Grafik keluaran dari Matlab sebagai hasil pelatihan dengan 4 lapisan tersembunyi, jumlah neuron 80, 60, 40, 20 *learning rate* 0.02, *momentum* 0.01, galat = 0.00001, dan epoh 20000

Grafik di atas menunjukkan epoh maksimal 20000 tercapai pada epoh 2645, nilai rata - rata *error* kuadrat (*Mean Square Error*) yang dicapai adalah 0.00000996498 dari galat maksimal yang ditentukan 0.00001, sehingga proses pelatihan mencapai error yang telah ditetapkan (*performance goal met*).

Penambahan jumlah *neuron* lapisan tersembunyi tidak menghasilkan penurunan jumlah iterasi yang berarti. Sedangkan untuk penggunaan nilai *learning rate* dan *momentum* harus dicari nilai yang terbaik. Makin besar nilai *learning rate* yang digunakan maka makin sedikit jumlah iterasi yang dibutuhkan untuk mencapai *error* yang kecil akan tetapi penggunaan nilai *learning rate* yang

terlalu besar akan memperbesar kemungkinan *error* yang terjadi dan makin kecil nilai *momentum* yang digunakan maka makin banyak iterasi yang dibutuhkan untuk mencapai *error* yang kecil. Setelah mendapatkan struktur jaringan saraf tiruan yang terbaik, dilakukan pengujian jaringan.

4.2 Hasil Pengujian Jaringan Saraf Tiruan *Backpropagation*

Pengujian pada aplikasi jaringan saraf tiruan pada sistem pendeteksi sidik jari dilakukan dengan men*capture image* sidik jari secara tidak langsung menggunakan *biopod fingerprint sensor. Images* sidik jari yang akan dikenali diwakili oleh masing - masing *image* sidik jari dalam bentuk biner (0 dan 1) untuk setiap sidik jari yang dikenali. Kemudian program akan menentukan siapa nama pemilik sidik jari dari setiap *images* yang tersimpan di *database*.

Untuk pengujian dapat dilakukan dengan terlebih dahulu menekan tombol "Pengujian" pada *interface* menu utama kemudian tekan "Pilih List" pada *interface* pengujian dan dilanjutkan dengan menekan tombol "Deteksi" untuk pengenalan pola *image* sidik jari. *Image* sidik jari yang akan dikenali dapat dilihat pada *axes GUI* berupa *image* hasil *capture* sidik jari yang akan dikenali.

4.2.1 Hasil Pengujian Jaringan Saraf Tiruan *Backpropagation* Pada Sistem Pengenalan Pola Sidik Jari Terhadap Data Pelatihan

Pengujian dapat dicoba dengan menggunakan salah satu jaringan dari hasil pelatihan yang dianggap terbaik untuk mengenali pola sidik jari. Dimana hasil pelatihan terbaik menggunakan jaringan dengan 4 lapisan tersembunyi, jumlah neuron 80, 60, 40, 20 *learning rate* 0.02, *momentum* 0.1, galat = 0.00001 dan maksimal epoh 20000 dengan fungsi aktivasi *logsig-logsig-logsig-logsig*,

diperlihatkan pada gambar 4.1. Hasil pengujian dapat dilihat sebagai berikut :

Tabel 4.2 Hasil pengenalan pola sidik jari melalui pelatihan

No	Nama *Input*	Target	Nama pemilik sidik jari yang dikenali	Output
1	Jempol_April	000001	Jempol_April	000001
2	Jempol_Ardi	000010	Jempol_Ardi	000010
3	Jempol_Dewi	000011	Jempol_Dewi	000011
4	Jempol_Erna	000100	Jempol_Erna	000100
5	Jempol_Fitri	000101	Jempol_Fitri	000101
6	Jempol_Linda	000110	Jempol_Linda	000110
7	Jempol_Mufi	000111	Jempol_Mufi	000111
8	Jempol_Ratih	001000	Jempol_Ratih	001000
9	Jempol_Rendy	001001	Jempol_Rendy	001001
10	Jempol_Seli	001010	Jempol_Seli	001010
11	Kelingking_April	001011	Kelingking_April	001011
12	Kelingking_Ardi	001100	Kelingking_Ardi	001100
13	Kelingking_Dewi	001101	Kelingking_Dewi	001101
14	Kelingking_Erna	001110	Kelingking_Erna	001110
15	Kelingking_Fitri	001111	Kelingking_Fitri	001111
16	Kelingking_Linda	010000	Kelingking_Linda	010000
17	Kelingking_Mufi	010001	Kelingking_Mufi	010001
18	Kelingking_Ratih	010010	Kelingking_Ratih	010010
19	Kelingking_Rendy	010011	Kelingking_Rendy	010011
20	Kelingking_Seli	010100	Kelingking_Seli	010100
21	Manis_April	010101	Manis_April	010101
22	Manis_Ardi	010110	Manis_Ardi	010110
23	Manis_Dewi	010111	Manis_Dewi	010111
24	Manis_Erna	011000	Manis_Erna	011000
25	Manis_Fitri	011001	Manis_Fitri	011001
26	Manis_Linda	011010	Manis_Linda	011010
27	Manis_Mufi	011011	Manis_Mufi	011011
28	Manis_Ratih	011100	Manis_Ratih	011100
29	Manis_Rendy	011101	Manis_Rendy	011101
30	Manis_Seli	011110	Manis_Seli	011110
31	Telunjuk_April	011111	Telunjuk_April	011111
32	Telunjuk_Ardi	100000	Telunjuk_Ardi	100000
33	Telunjuk_Dewi	100001	Telunjuk_Dewi	100001
34	Telunjuk_Erna	100010	Telunjuk_Erna	100010
35	Telunjuk_Fitri	100011	Telunjuk_Fitri	100011
36	Telunjuk_Linda	100100	Telunjuk_Linda	100100
37	Telunjuk_Mufi	100101	Telunjuk_Mufi	100101

Lanjutan Tabel 4.2

38	Telunjuk_Ratih	100110	Telunjuk_Ratih	100110
39	Telunjuk_Rendy	100111	Telunjuk_Rendy	100111
40	Telunjuk_Seli	101000	Telunjuk_Seli	101000
41	Tengah_April	101001	Tengah_April	101001
42	Tengah_Ardi	101010	Tengah_Ardi	101010
43	Tengah_Dewi	101011	Tengah_Dewi	101011
44	Tengah_Erna	101100	Tengah_Erna	101100
45	Tengah_Fitri	101101	Tengah_Fitri	101101
46	Tengah_Linda	101110	Tengah_Linda	101110
47	Tengah_Mufi	101111	Tengah_Mufi	101111
48	Tengah_Ratih	110000	Tengah_Ratih	110000
49	Tengah_Rendy	110001	Tengah_Rendy	110001
50	Tengah_Seli	110010	Tengah_Seli	110010

4.2.2 Hasil Pengujian Jaringan Saraf Tiruan *Backpropagation* Pada Sistem Pengenalan Pola Sidik Jari Tanpa Menggunakan Data Pelatihan

Pengujian sistem pendeteksi sidik jari dapat dilakukan secara langsung tanpa melalui proses pelatihan dengan menambahkan satu objek yang berisi lima *images* sidik jari sehingga total data yang akan diuji berjumlah 55 sampel.

Tabel 4.3 Hasil pengenalan pola sidik jari tanpa melalui pelatihan

No	Nama *Input*	Target	Nama pemilik sidik jari yang dikenali	Output
1	Jempol_April	000001	Jempol_April	000001
2	Jempol_Ardi	000010	Jempol_Ardi	000010
3	Jempol_Dewi	000011	Jempol_Dewi	000011
4	Jempol_Erna	000100	*Tidak dikenal*	000000
5	Jempol_Fitri	000101	*Kelingking_Linda*	010010
6	Jempol_Lia	000110	Jempol_Lia	000110
7	Jempol_Linda	000111	*Jempol_Fitri*	000101
8	Jempol_Mufi	001000	Jempol_Mufi	001000
9	Jempol_Ratih	001001	Jempol_Ratih	001001
10	Jempol_Rendy	001010	*Jempol_Ardi*	000010
11	Jempol_Seli	001011	Jempol_Seli	001011
12	Kelingking_April	001100	Kelingking_April	001100
13	Kelingking_Ardi	001101	*Kelingking_April*	001100
14	Kelingking_Dewi	001110	*Manis_April*	010111
15	Kelingking_Erna	001111	Kelingking_Erna	001111

Lanjutan Tabel 4.3

16	Kelingking_Fitri	010000	*Tidak dikenal*	000000
17	Kelingking_Lia	010001	*Kelingking_Fitri*	010000
18	Kelingking_Linda	010010	*Jempol_Ardi*	000010
19	Kelingking_Mufi	010011	*Jempol_April*	000001
20	Kelingking_Ratih	010100	Kelingking_Ratih	010100
21	Kelingking_Rendy	010101	*Kelingking_Ratih*	010100
22	Kelingking_Seli	010110	Kelingking_Seli	010110
23	Manis_April	010111	*Kelingking_Rendy*	010101
24	Manis_Ardi	011000	Manis_Ardi	011000
25	Manis_Dewi	011001	*Jempol_Seli*	001011
26	Manis_Erna	011010	Manis_Erna	011010
27	Manis_Fitri	011011	*Kelingking_Fitri*	010000
28	Manis_Lia	011100	*Manis_Erna*	011010
29	Manis_Linda	011101	Manis_Linda	011101
30	Manis_Mufi	011110	Manis_Mufi	011110
31	Manis_Ratih	011111	Manis_Ratih	011111
32	Manis_Rendy	100000	Manis_Rendy	100000
33	Manis_Seli	100001	*Jempol_April*	000001
34	Telunjuk_April	100010	*Jempol_Ardi*	000010
35	Telunjuk_Ardi	100011	*Jempol_Dewi*	000011
36	Telunjuk_Dewi	100100	*Manis_Rendy*	100000
37	Telunjuk_Erna	100101	*Telunjuk_Lia*	100111
38	Telunjuk_Fitri	100110	Telunjuk_Fitri	100110
39	Telunjuk_Lia	100111	Telunjuk_Lia	100111
40	Telunjuk_Linda	101000	*Jempol_Erna*	000100
41	Telunjuk_Mufi	101001	*Manis_Seli*	100001
42	Telunjuk_Ratih	101010	*Tengah_Dewi*	101111
43	Telunjuk_Rendy	101011	Telunjuk_Rendy	101011
44	Telunjuk_Seli	101100	*Telunjuk_Linda*	101000
45	Tengah_April	101101	*Telunjuk_Seli*	101100
46	Tengah_Ardi	101110	*Kelingking_Dewi*	001110
47	Tengah_Dewi	101111	Tengah_Dewi	101111
48	Tengah_Erna	110000	*Kelingking_Fitri*	010000
49	Tengah_Fitri	110001	*Kelingking_Lia*	010001
50	Tengah_Lia	110010	*Tidak dikenal*	111010
51	Tengah_Linda	110011	*Tengah_Fitri*	110001
52	Tengah_Mufi	110100	*Kelingking_Ratih*	010100
53	Tengah_Ratih	110101	Tengah_Ratih	110101
54	Tengah_Rendy	110110	*Tengah_Lia*	110010
55	Tengah_Seli	110111	*Manis_April*	010111

Keterangan :

Cetak miring : Menunjukkan kesalahan pengenalan (tidak dikenal), yang berarti output tidak sama dengan target yang ditentukan.

4.2.3 *Prosentase* Hasil Pengenalan Pola Sidik Jari

Klasifikasi yang dilakukan oleh jaringan saraf tiruan terhadap *image* sidik jari dapat dihitung *prosentase*-nya untuk sistem pengenalan pola *image* sidik jari. *Prosentase* pengenalan pola *image* sidik jari dihitung dengan cara sebagai berikut :

$$\%pengenalan\ pola = \frac{\sum N_T}{\sum N_S} x100\% \qquad (4.1)$$

Dengan :

$\sum N_T$ merupakan jumlah *image* yang terdeteksi

$\sum N_S$ merupakan jumlah *image* yang diujikan

Berdasarkan perhitungan di atas, maka hasil pengujian terhadap data pelatihan dapat dituliskan sebagai berikut :

$$\%pengenalan\ pola = \frac{50}{50} x100\%$$

$$= 100\ \%$$

Sedangkan untuk perhitungan *prosentase* pengenalan pola *image* sidik jari hasil pengujian tanpa menggunakan data pelatihan adalah sebagai berikut :

$$\%pengenalan pola = \frac{22}{55} x100\%$$

$$= 40\%$$

Hasil pengujian dengan pengenalan *image* sidik jari tanpa menggunakan data pelatihan diperoleh *prosentase* keberhasilan sebesar 40%. Sehingga ketika pengujian sering terjadi kesalahan dalam pembacaan nama pemilik sidik jari. Sedangkan jika pengujian menggunakan data pelatihan diperoleh *prosentase* keberhasilan sebesar 100%.

Terjadinya *error* karena adanya perbedaan jumlah antara *image* sidik jari yang tidak mengalami proses pelatihan dengan *image* sidik jari yang pernah dilatihkan. Dengan kata lain disebabkan oleh proses pengolahan *image* yaitu dengan penambahan *image* sidik jari otomatis mengubah program yang ada di dalamnya. Akibatnya jika *image* sidik jari yang tidak melalui proses pelatihan, memungkinkan program akan salah menentukan beberapa nama pemilik sidik jari dari *image* tersebut atau bahkan tidak dapat dikenali.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil pengamatan terhadap program simulasi jaringan saraf tiruan *backpropagation* sebagai pendeteksi sidik jari dapat disimpulkan sebagai berikut :

1. Struktur jaringan saraf tiruan *backpropagation* yang terbaik untuk sistem pengenalan pola sidik jari yaitu dengan menggunakan 4 lapisan tersembunyi dengan jumlah *neuron* pada masing - masing lapisan adalah 80, 60, 40 dan 20. Nilai *Learning Rate* yang digunakan adalah 0,02 dan *Momentum* 0,01 dengan target error 0,00001 (10^{-5}) serta fungsi aktivasi yang digunakan untuk semua hidden layer yaitu *logsig* ,sedangkan fungsi aktivasi untuk lapisan *output* yaitu *logsig*.
2. Penambahan jumlah neuron dan jumlah lapisan tersembunyi akan menghasilkan penurunan iterasi (*epoch*) yang berarti sehingga akan didapat jaringan yang terbaik untuk proses pengujian *image* sidik jari.
3. Untuk penggunaan nilai *learning rate* dan *momentum* harus dicari nilai yang terbaik. Makin besar nilai *learning rate* yang digunakan maka makin sedikit jumlah iterasi yang dibutuhkan untuk mencapai *error* yang kecil akan tetapi penggunaan nilai *learning rate* yang terlalu besar akan memperbesar kemungkinan *error* yang terjadi dan makin kecil nilai *momentum* yang digunakan maka makin banyak iterasi yang dibutuhkan untuk mencapai *error* yang kecil.

4. *Prosentase* keberhasilan hasil pengujian sistem pengenalan pola sidik jari menggunakan data pelatihan adalah 100 % sedangkan jika tidak menggunakan data pelatihan diperoleh 40 % untuk tingkat keberhasilan.
5. Kesalahan pembacaan nama pemilik sidik jari pada saat pengujian disebabkan oleh perbedaan pola sidik jari yang pernah dilatihkan dan juga belum didapatnya jaringan yang terbaik untuk proses pengenalan sidik jari pada saat pelatihan.

5.2 Saran

1. Pada tugas akhir ini digunakan jaringan saraf tiruan dengan algoritma belajar perambatan - balik (*backpropagation*) untuk mendeteksi *image* sidik jari. Diharapkan dipakai bentuk metode lain yang lebih baik untuk mendeteksi *image* sidik jari pada pengembangan berikutnya.
2. *Image* sidik jari yang akan dikenali dibatasi hanya sepuluh orang. Diharapkan pada penerapannya nanti dapat mengenali sidik jari lebih dari 10 orang.
3. Pada tugas akhir ini jaringan hanya dapat menentukan nama pemilik dari *image* sidik jari saja. Diharapkan jaringan dapat mendeteksi biodata lain seperti tanggal lahir, alamat, nomor telepon, dan lain - lain.
4. Untuk pengembangan selanjutnya supaya dapat dilakukan penelitian lanjutan tentang aplikasi sidik jari seperti peralatan - peralatan yang menggunakan akses sidik jari untuk keperluan perusahaan, perkantoran, bandara, rumah sakit, sekolah, warnet dan sarana umum lainnya, misalkan dalam pengadaan mesin absensi sidik jari, mesin pencari urutan daftar penumpang untuk daftar tunggu bandara, sistem keamanan *database* untuk membuka password

tertentu, dan untuk menyimpan data medis pasien pada suatu rumah sakit, dan lain sebagainya.

5. Untuk penelitian selanjutnya dikembangkan arsitektur JST *backpropagation* yang dipadukan dengan metode jaringan saraf tiruan yang lain agar memiliki kemampuan belajar yang lebih baik lagi yaitu dimisalkan penggabungan dengan arsitektur JST *Widrow - Hoff,* sehingga diharapkan memberikan proses belajar yang lebih cepat dan akurat untuk sistem yang kompleks.

DAFTAR PUSTAKA

Ahmad, Usman. 2005. *Pengolahan Citra Digital & Teknik Pemrogramannya,* Jogjakarta : Graha Ilmu.

Elvayandri. 2002. *Sistem Keamanan Akses Menggunakan Pola Sidik Jari Berbasis Jaringan Saraf Tiruan*, Bandung : ITB, http://www.cert.or.id/~budi/courses/el695/projects/report-elva.pdf

Fahmi, Ahmad. 2003. *Pengenalan Pola Sidik Jari Dengan Artificial Neural Network Menggunakan Metode Backpropagation*, Malang : Universitas Brawijaya, http://jurnalft.brawijaya.ac.id/?hlm=jfull&edisi=1059670800&idj=1099449023

Jain, Anil K. 1989. *Fundamentals of Digital Image Processing,* Singapore : Prentice-Hall International Inc.

Kristanto, Andri. 2004. *Jaringan Saraf Tiruan (Konsep Dasar, Algoritma dan Aplikasi)*, Jogjakarta : Gava Media.

Kusumadewi, Sri. 2004. *Membangun Jaringan Saraf Tiruan Menggunakan Matlab & Excel Link,* Jogjakarta : Graha Ilmu.

Russ, John C. 1995. *The Image Processing Handbook*, , Florida : CRC Press.

Siang, Jong Jek. 2005. *Jaringan Saraf Tiruan dan Pemrograman menggunakan Matlab*, Jogjakarta : ANDI

Thompson, Clay M dan Shure, Loren. 1993. *Image Processing Toolbox*, United States : The Mathworks Inc.

_____, *Help Matlab version 7.0,* The Mathworks.