

SKRIPSI

SISTEM PENGENAL TANDA TANGAN DENGAN JARINGAN SARAF TIRUAN METODE PROPAGASI BALIK



Disusun Oleh :

ZULAIKHA PRABAWATI

05.12.696

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2010**

1950

INSTITUTIONAL RESEARCH AND STATISTICS
RESEARCH AND STATISTICS
RESEARCH AND STATISTICS
RESEARCH AND STATISTICS

RESEARCH AND STATISTICS
RESEARCH AND STATISTICS
RESEARCH AND STATISTICS

RESEARCH AND STATISTICS
RESEARCH AND STATISTICS
RESEARCH AND STATISTICS

RESEARCH AND STATISTICS

LEMBAR PERSETUJUAN

**SISTEM PENGENAL TANDA TANGAN
DENGAN JARINGAN SARAF TIRUAN METODE PROPAGASI BALIK**

SKRIPSI

*Disusun dan Diajukan sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Komputer Dan Informatika Strata Satu (S-1)*

Disusun Oleh :

ZULAIKHA PRABAWATI

NIM : 05.12.696

Mengetahui

Ketua Jurusan Teknik Elektro S-1



Ir. Yusuf Ismail Nakhoda, MT

NIP. Y.1018800189

Diperiksa dan Disetujui
Dosen Pembimbing

I Komang Somawirata, ST. MT

NIP. Y.1030100361

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

2010

ABSTRAK

SISTEM PENGENAL TANDA TANGAN DENGAN JARINGAN SARAF TIRUAN METODE PROPAGASI BALIK

Zulaikha Prabawati

Pembimbing : I Komang Somawirata ST, MT.

Verifikasi orang yang memiliki kepentingan dapat dilakukan dengan cara mengecek keabsahan tanda tangannya. Tanda tangan memiliki keunikan masing-masing sesuai dengan pribadi pemiliknya. Semakin rumit sebuah tanda tangan, maka kecil pula resiko yang ditirukan. Contohnya seperti tanda tangan pada transaksi pengambilan uang pada perbankan.

Sistem pengecekan tanda tangan secara manual memiliki kelemahan, di mana si pemeriksa kurang teliti dalam membandingkan tanda tangan. Sistem pengenalan tanda tangan merupakan aplikasi yang berfungsi untuk mengenali tanda tangan manusia, apakah tanda tangan tersebut merupakan tanda tangan dari orang yang membubuhkan tanda tangan tersebut atau bukan. Dengan menggunakan Jaringan Saraf Tiruan metode propagasi balik, sistem dapat mengenali dan kemudian membandingkan dengan data tangan yang sudah disimpan sebelumnya. Sistem yang akan dibuat terdiri dari dua tahap, yaitu tahap pelatihan (*training*) dan tahap pengenalan (*recognize*).

Tahap pelatihan atau pembelajaran merupakan tahapan pertama sebelum dapat melakukan pengenalan, di mana dalam tahapan ini sistem akan melakukan pembelajaran terhadap citra tanda tangan tersebut yang dimasukkan ke dalam sistem.

Pada tahap pengenalan, sistem akan melakukan pengenalan terhadap citra yang diinputkan oleh user. Dalam tahapan ini, data hasil tanda tangan yang diinputkan oleh user akan dicocokkan dengan data pengetahuan hasil dari tahapan sebelumnya, yaitu proses pembelajaran. Dalam setiap tahapan tersebut di atas, terdapat proses pengolahan citra (*image processing*).Diharapkan aplikasi ini dapat mengatasi kelemahan pengecekan tanda tangan secara manual.

Kata kunci : Tanda Tangan, Jaringan Saraf Tiruan, Propagasi Balik

ABSTRACT

The verification of the importance person could done with the way to check the validation of its signature. Each signature have unique character appropriate its owner. The more complicated signature character is the smaller change to imitativative, for the example is a signature of bank transaction.

The checker system of signature manually have enervation, which the checker can be careless when the signature compared. The signature recognize system is an application that used to identify the human signature, whether its person's append or other. By the Artificial Neural Network , the system backpropagation method, the system could identify and compare the past signature with the new one. The system will be made for two steps, the training and the recognize.

The training is the first step before the recognize could do, which is in this step, system will do trained of acquire to the signature images for input to the system.

On the steps of recognize, the system will recognized to the image for input by user. In this steps, the data of signature character by user will matched with the data erudition from the steps before (trained of acquire). On each steps, there were image processing. Wish this application can break the enervation of signature examine manually.

Keyword : Signature, Artificial Neural Network, Backpropagation

KATA PENGANTAR

Dengan mengucapkan puji syukur atas kehadiran Allah SWT, yang telah memberikan rahmat dan hidayah-Nya, sehingga saya dapat menyelesaikan skripsi dengan judul:

“ SISTEM PENGENAL TANDA TANGAN DENGAN JARINGAN SARAF TIRUAN METODE PROPAGASI BALIK“

Pembuatan skripsi ini disusun untuk memenuhi syarat akhir kelulusan pendidikan jenjang Strata I di Institut Teknologi Nasional Malang.

Terima kasih yang mendalam penulis haturkan kepada kedua orang tua yang telah berjasa membentuk kepribadian penulis, khususnya memberikan inspirasi yang kuat serta dukungan dan doa yang tiada henti. Terima kasih juga penulis sampaikan kepada :

1. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE selaku rektor ITN Malang.
2. Bapak Ir. H. Sidik Noertjahjono, MT selaku Dekan Fakultas Teknologi Industri ITN Malang.
3. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Ketua Jurusan Teknik Elektro S-1 ITN Malang.
4. Bapak I Komang Somawirata ST, MT selaku Dosen Pembimbing.
5. Semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Penulis menyadari bahwa skripsi ini masih banyak yang perlu disempurnakan. Oleh sebab itu kritik dan saran yang membangun sangat diharapkan.

Akhir kata penulis mohon maaf kepada semua pihak jika dalam proses pembuatan skripsi ini penulis melakukan kesalahan baik yang disengaja maupun tidak disengaja.

Semoga Allah SWT selalu melimpahkan rahmat dan hidayah-Nya kepada kita semua. Amin.

Malang, Agustus 2010

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
ABSTRAK	ii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan	2
1.4. Batasan Masalah	2
1.5. Metodologi	3
1.6. Sistematika Penulisan	4
BAB II TEORI DASAR.....	5
2.1. Microsoft Visual Basic 6.0	5
2.1.1. Pengenalan Microsoft Visual Basic.....	5
2.1.2. Sekilas tentang ADO.....	9
2.2. Microsoft Access XP	9

2.3. Pengertian Tanda Tangan	11
2.4. Jaringan Saraf Tiruan (<i>Artificial Neural Network</i>)	12
2.4.1. Pengertian.....	12
2.4.2. Arsitektur Jaringan	14
2.4.3. Metode Pelatihan / Pembelajaran.....	15
2.4.4. Fungsi Aktivasi	17
2.4.5. Propagasi Balik (<i>Backpropagation</i>).....	19
2.4.6. Algoritma Propagasi Balik (<i>Backpropagation</i>).....	21
2.5. Citra Digital	30
2.6. Pengolahan Citra (<i>Image Processing</i>)	31
2.6.1. Teknik-Teknik Pengolahan Citra Digital.....	31
2.6.2. Interpolasi Citra Digital	34
2.6.3. <i>Cropping</i>	36
2.6.4. Warna	36
2.6.5. Ekstrasi Ciri	38
2.6.6. Pixel.....	38
2.6.7. Dasar-Dasar Pengolahan Citra	39
2.6.7.1. Mengubah Citra Berwarna menjadi <i>Grayscale</i>	39
2.6.7.2. <i>Thresholding</i>	40
2.6.7.3. Konversi ke Citra <i>Biner</i>	41
BAB III METODOLOGI DAN PERANCANGAN SISTEM.....	43
3.1. Gambaran Umum	43

3.2. Perancangan Sistem	44
3.2.1. Perancangan Arsitektur dan Proses.....	44
3.2.1.1. Proses Pelatihan.....	55
3.2.1.2. Proses Pengenalan	57
3.2.2. Tempat Penyimpanan Data (<i>Data Storage</i>).....	59
3.2.3. Perancangan Antarmuka	62
3.2.4. Contoh Perhitungan Matematis.....	64
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM.....	88
4.1. Perancangan Sistem	88
4.1.1. Form Menu Utama	89
4.1.2. Form User.....	89
4.1.3. Form Pelatihan	89
4.1.4. Form Pengujian	90
4.2. Pengujian Sistem	91
4.2.1. Tahap Pelatihan.....	91
4.2.2. Tahap Pengenalan.....	93
4.3. Analisa Hasil	94
BAB V PENUTUP	97
5.1. Kesimpulan	97

5.2. Saran	98
DAFTAR PUSTAKA	99
LAMPIRAN	100

DAFTAR TABEL

Tabel 2.1.	Simbol-Simbol Algoritma <i>Backpropagation</i>	29
Tabel 2.2.	Contoh-Contoh Warna dalam Hexadesimal	38
Tabel 3.1.	Tabel User	57
Tabel 3.2.	Tabel Pelatihan	57
Tabel 3.3.	Tabel Bobot Awal Bias To <i>Hidden</i>	57
Tabel 3.4.	Tabel Bobot Awal <i>Input</i> To <i>Hidden</i>	58
Tabel 3.5.	Tabel Bobot Awal Bias To <i>Output</i>	58
Tabel 3.6.	Tabel Bobot Awal <i>Hidden</i> To <i>Output</i>	58
Tabel 4.6.	Hasil Persentase dengan Bataas Pengenalan 90%	91

DAFTAR GAMBAR

Gambar 2.1. Tampilan Visual Basic 6.0	5
Gambar 2.2. Form <i>Designer</i>	6
Gambar 2.3. <i>Toolbox</i>	6
Gambar 2.4. <i>Properties</i>	7
Gambar 2.5. Menu <i>Toolbar</i>	7
Gambar 2.6. Jaringan dengan Lapisan Tunggal (<i>Single Layer Net</i>)	14
Gambar 2.7. Jaringan dengan Banyak Lapisan (<i>Multilayer Net</i>)	15
Gambar 2.8. Jaringan dengan Lapisan Kompetitif (<i>Competitive Layer Net</i>)	15
Gambar 2.9. Diagram Alir Proses Pembelajaran Jaringan Saraf Tiruan	16
Gambar 2.10. Metode Propagasi Balik	20
Gambar 2.11. Ilustrasi Interpolasi Citra Digital	35
Gambar 2.12. Nilai Warna RGB dalam <i>Hexadecimal</i>	37
Gambar 2.13. Komposisi Warna RGB	37
Gambar 2.14. Proses <i>Grayscale</i>	40
Gambar 2.15. Proses <i>Thresholding</i> dengan Derajat Keabuan 256	40
Gambar 2.16. Proses <i>Thresholding</i> dengan Derajat Keabuan 16	41
Gambar 2.17. Proses <i>Thresholding</i> dengan Derajat Keabuan 4	41
Gambar 2.18. Konversi ke <i>Biner</i>	41
Gambar 3.1. Diagram Konteks	43
Gambar 3.2. Diagram Rinci Pelatihan	43
Gambar 3.3. Diagram Rinci Pengenalan	44

Gambar 3.4. Diagram Alir Pengolahan Citra	44
Gambar 3.5. Pengelompokan Matriks 100x100 dengan Setiap Sepuluh Satuan Data secara Horizontal dan Vertikal	47
Gambar 3.6. Matriks Kolom Pertama Baris Terakhir	47
Gambar 3.7. Matriks 10x100	51
Gambar 3.8. Konversi ke Matriks 10x10	52
Gambar 3.9. Matriks 10x10	52
Gambar 3.10. Matriks 100x1	53
Gambar 3.11. Diagram Alir Proses Pelatihan	54
Gambar 3.12. Diagram Alir Proses Pengenalan/Pengujian.....	55
Gambar 3.13. Desain Form Menu Utama	59
Gambar 3.14. Desain Form User.....	59
Gambar 3.15. Desain Form Pelatihan	60
Gambar 3.16. Desain Form Pengujian	60
Gambar 4.1. Form Menu Utama.....	84
Gambar 4.2. Form User	85
Gambar 4.3. Form Pelatihan.....	85
Gambar 4.4. Form Pengujian.....	86
Gambar 4.5. Citra Tanda Tangan Tahap Pelatihan.....	87
Gambar 4.6. Pengisian Nama <i>User</i>	88
Gambar 4.7. Mengambil File Citra Tanda Tangan.....	88
Gambar 4.8. Tampilan Hasil Pelatihan.....	89
Gambar 4.9. Tampilan Hasil Pengujian.....	90

BAB I

PENDAHULUAN

1.1 Latar Belakang

Verifikasi orang yang memiliki kepentingan dapat dilakukan dengan cara mengecek keabsahan tanda tangannya. Tanda tangan memiliki keunikan masing-masing sesuai dengan pribadi pemiliknya. Semakin rumit sebuah tanda tangan, maka kecil pula resiko yang ditirukan. Contohnya seperti tanda tangan pada transaksi pengambilan uang pada perbankan. Sistem pengecekan tanda tangan secara manual memiliki kelemahan, di mana si pemeriksa kurang teliti dalam membandingkan tanda tangan.

Saat ini, teknologi komputer berkembang sangat pesat. Salah satunya yaitu komputer dapat diberi kecerdasan buatan (*Intelligence*). Dengan demikian, muncullah berbagai ilmu baru, diantaranya, AI (*Artificial Intelligence*) dan ANN (*Artificial Neural Network*). Perkembangan AI (*Artificial Intelligence*) yang semakin pesat dewasa ini ditunjukkan oleh aplikasinya di berbagai bidang.

Berdasarkan keadaan tersebut, salah satu solusi untuk kelemahan sistem pengecekan tanda tangan ini adalah menggantikannya dengan sistem Jaringan Saraf Tiruan (*Artificial Neural Network*) dengan metode Propagasi Balik (*Backpropagation*). Jaringan Saraf Tiruan merupakan sebuah sistem pemroses informasi yang memiliki performa karakteristik tertentu seperti jaringan saraf biologi.

Dengan menggunakan Jaringan Saraf Tiruan, maka sebuah komputer dapat mengenali dan kemudian membandingkan dengan data tanda tangan yang sudah disimpan sebelumnya untuk memastikan orang yang membubuhkan tanda tangan adalah orang yang memiliki tanda tangan tersebut.

1.2 Rumusan Masalah

Perumusan masalah dalam pembuatan skripsi “Sistem Pengenal Tanda Tangan dengan Jaringan Saraf Tiruan Metode Propagasi Balik” ini, adalah bagaimana sistem dapat mengenali tanda tangan.

1.3 Tujuan

Tujuan dari pembuatan skripsi ini adalah menciptakan suatu sistem perangkat lunak untuk membantu mengenali tanda tangan dari pemilik aslinya. Sehingga dapat mengurangi faktor kesalahan manusia (*human error*) dalam pengecekan tanda tangan.

1.4 Batasan Masalah

Batasan masalah yang digunakan diharapkan mampu membatasi pembahasan. Agar dalam pembahasan skripsi ini tidak melebar, maka penulis membatasi permasalahan sebagai berikut:

- a. Citra tanda tangan yang dipakai harus berupa file bitmap dengan resolusi maksimum 100x100 piksel.

- b. Citra tanda tangan bersih dari noise.
- c. Sebelum proses *threshold*, citra diubah ke *grayscale* 8 bit.
- d. Hanya membahas proses sistem pengenalan tanda tangan dengan jaringan saraf tiruan metode propagasi balik

1.5 Metodologi

Metode - metode yang digunakan dalam penyusunan skripsi ini adalah :

a. Studi Pustaka dan Literatur

Dalam pembuatan skripsi ini dilakukan penelitian kepustakaan, yaitu dengan mempelajari buku-buku literatur atau laporan hasil penelitian sejenis dan juga data-data yang dikumpulkan dari internet yang ada kaitannya dengan masalah yang akan dibahas.

b. Perancangan dan Implementasi Perangkat Lunak

Dalam pembuatan skripsi ini, dilakukan perancangan dan implementasi perangkat lunak. Beberapa perangkat lunak yang digunakan dalam skripsi ini adalah :

- o Microsoft Visual Basic 6.0
- o Microsoft Access XP

c. Uji Coba dan Analisa

Melakukan uji coba dan analisa terhadap perangkat lunak yang telah dibuat dengan data yang telah didapat untuk proses klasifikasi data.

d. Penulisan Laporan.

1.6 Sistematika Penulisan

Sistematika pembahasan dari penyusunan skripsi ini direncanakan sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi tentang pendahuluan yang terdiri dari latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi serta sistematika penulisan dari skripsi ini.

BAB II DASAR TEORI

Bab ini membahas mengenai teori – teori yang berkaitan dalam perencanaan dan pembuatan sistem.

BAB III METODOLOGI DAN PERANCANGAN SISTEM

Bab ini membahas tentang metodologi dan perancangan yang akan digunakan dalam sistem ini.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Berisi tentang implementasi dari perancangan sistem yang telah dibuat serta pengujian terhadap sistem tersebut.

BAB V PENUTUP

Bab ini merupakan bagian terakhir dari laporan skripsi ini yang berisi tentang kesimpulan dari hasil pembahasan pada perancangan dan pengujian akhir sistem yang dibuat serta saran-saran untuk penyempurnaan dalam pengembangannya.

BAB II

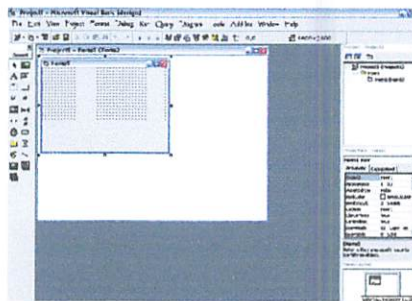
DASAR TEORI

2.1 Microsoft Visual Basic 6.0

2.1.1 Pengenalan Microsoft Visual Basic 6.0

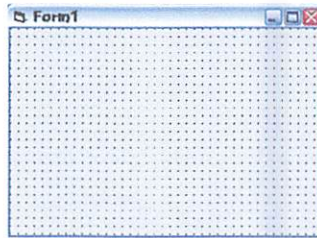
Microsoft Visual Basic 6.0 merupakan produk pengembangan dari Microsoft Visual Basic yang sebelumnya. Visual Basic dapat berinteraksi dengan aplikasi lain di dalam sistem operasi Windows dengan komponen ActiveX Control. Dengan komponen ini memungkinkan pengguna untuk memanggil dan menggunakan semua model data yang ada di dalam sistem operasi windows.

Teknologi akses data yang dimiliki oleh Visual Basic 6.0 yaitu teknologi ActiveX Data Object atau yang lebih dikenal dengan nama ADO. ADO adalah teknologi terbaru dari Microsoft untuk memanipulasi informasi dari *database* relasional dan nonrelasional. ADO mampu mengintegrasikan program aplikasi *database* yang dibangun dengan berbagai sumber data seperti Microsoft Access, SQL Server, ODBC, Oracle dan lain sebagainya.



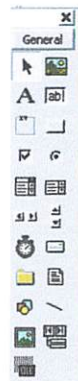
Gambar 2.1
Tampilan Visual Basic 6.0

Form adalah bagian dari *project* yang digunakan untuk mendesain windows yang merupakan tampilan antar muka program. *Form* juga digunakan untuk menampilkan *label*, *textbox*, grafik dan kontrol – kontrol lainnya yang digunakan dalam mendesain tampilan program atau untuk memasukan data.



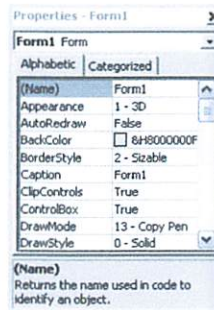
Gambar 2.2
Form *Designer* pada Visual Basic 6.0

Toolbox menampilkan semua standart Visual Basic kontrol ditambah dengan kontrol yang bisa kita *custom* dalam *project* tersebut. Di sinilah tempat komponen-komponen yang disediakan untuk merancang *user interface*. Masing-masing komponen memiliki ciri dan kegunaan masing-masing. Komponen kita gunakan sesuai kebutuhan kita.



Gambar 2.3
Toolbox

Kita dapat mengatur *properties* dari *form* dan kontrol dengan menggunakan *properties window*. Dengan *properties* ini kita dapat merubah nilai dari *form* atau kontrol tersebut seperti *caption*, *name*, *font*, *colour*, *size*, dan *position*.



Gambar 2.4
Properties

Visual Basic *Toolbar* juga menyediakan *shortcuts* untuk operasi yang biasa kita gunakan.



Gambar 2.5
Menu *Toolbar*

1. Label

Label adalah kontrol grafis yang dapat anda gunakan untuk menampilkan *text* dimana *user* tidak dapat merubahnya secara langsung. Kita harus merubahnya melalui kode pemrograman pada saat *run time*. Untuk merubah *text* - nya, kita bisa mengubah *caption* - nya dan mengisikan *text* sesuai dengan keinginan kita. Ubah *AutoSize* dan *WordWrap* properti menjadi *true* jika anda ingin menampilkan seluruh *text* tanpa harus merubah kontrol label tersebut secara manual.

2. *TextBox*

TextBox kontrol digunakan untuk menampilkan informasi dimana *user* dapat merubahnya secara langsung. *TextBox* juga dapat dirubah melalui kode pemograman pada saat *run time*. Untuk merubah *text* - nya kita bisa merubah *text* properti pada *TextBox* tersebut. Untuk menampilkan banyak *text*, kita bisa merubah *MultiLine* properti menjadi *true*. Anda juga bisa menampilkan *scrollbar* pada *textbox* tersebut dengan merubah *scrollbar* properti.

3. *Command Button*

Kontrol ini yang paling sering digunakan karena berfungsi seperti tombol pada umumnya. Kita bisa merubah *text* - nya dengan merubah *caption* properti.

4. *List Box*

ListBox digunakan untuk menampilkan sejumlah item dimana *user* dapat memilih satu atau lebih item yang tersedia. Jika jumlah item yang ditampilkan melebihi ukurannya, *listbox* secara otomatis akan menampilkan *scrollbar*. Jika tidak ada item yang dipilih nilai pada *ListIndex* adalah -1. Item pertama akan diberi nilai 0 sedangkan nilai *ListCount* adalah jumlah item ditambah 1. Untuk menambah item menggunakan *AddItem* *methode* sedangkan untuk menghapusnya, menggunakan *RemoveItem* *methode*.

2.1.2 Sekilas tentang ADO

Teknologi yang dikembangkan oleh Microsoft ini yang memungkinkan aplikasi Visual Basic yang kita buat untuk berkomunikasi dengan *database*. ADO adalah sekumpulan objek yang menyediakan mekanisme untuk mengakses informasi dari sumber data (*database*).

ADO memungkinkan suatu aplikasi untuk mengakses dan memanipulasi data di suatu *server database* dengan metode yang sama walaupun *database* yang digunakan berasal dari vendor yang berbeda. Umumnya untuk mengakses dan memanipulasi *database* langkah-langkah yang dilakukan adalah sebagai berikut:

- menghubungkan aplikasi/skrip dengan *database*
- mengakses data atau melakukan perubahan data
- menampilkan hasil dari manipulasi data
- menutup objek untuk membebaskan memori.

2.2 Microsoft Access XP

Microsoft Access merupakan sebuah perangkat lunak basis data yang digunakan untuk membuat basis data relational. Ms. Access memiliki beberapa jenis basis data yang biasanya dideklarasikan pada waktu membuat tabel. Tipe data ini digunakan untuk mendefinisikan jenis data pada field – fieldnya. Beberapa tipe data pada Ms. Access yaitu :

1. *Text*

Text merupakan tipe data untuk huruf, angka, spasi dan tanda baca. Sebuah *field* dengan tanda baca *text* dapat menampung hingga 255 karakter atau selebar yang ditentukan dalam properti *field size*.

2. *Memo*

Memo dapat menerima *text* apa saja dengan panjang maksimal 65535.

3. *Number*

Number berisi data bilangan yang digunakan untuk perhitungan matematis, ukurannya tergantung properti *FieldSize*.

4. *Date / Time*

Tipe jenis ini hanya dapat digunakan untuk tanggal dan waktu.

5. *Currency*

Tipe data ini digunakan untuk tipe data uang dan data bilangan yang digunakan dalam perhitungan matematis .

6. *Auto Number*

Tipe data ini berisi bilangan yang berurutan secara otomatis diberikan oleh *access*.

7. *Yes / No*

Berisi nilai *Yes / No* atau *field* yang hanya memiliki dua kemungkinan nilai.

8. *OLE Object*

Tipe data ini berisi objek yang dikaitkan (*Link*) ke atau yang disisipkan ke dalam tabel Ms. Access.

9. *Hyperlink*

Tipe data ini berisi alamat URL.

10. *Lookup Wizard*

Tipe data ini memungkinkan untuk memilih nilai dari tabel lain atau dari daftar nilai yang didefinisikan sendiri.

Dalam sebuah tabel unsur pembentukan adalah adanya *field* dan *record*. *Field* menyatakan banyaknya kolom pada sebuah tabel. Sedangkan *record* baris pada tabel. *Record* merupakan isi dari tabel, sehingga jumlah *record* ini bisa bertambah atau berkurang sesuai dengan perubahan yang terjadi.

Field – field yang berada pada suatu tabel harus dipilih satu sebagai *primary key*. *Primary key* ini berfungsi sebagai kunci atau pembeda *field* pada suatu tabel. Syarat sebuah *field* untuk menjadi sebuah *primary key* adalah *field* tersebut harus unik. Unik ini artinya *field* tersebut (nilai / isinya) harus dibedakan antara *record* satu dengan lainnya.

2.3 Pengertian Tanda Tangan

Tanda tangan (Inggris: *signature* berasal dari Latin: *signare* yang berarti "tanda") atau Paraf adalah tulisan tangan, terkadang diberi gaya tulisan tertentu dari nama seseorang atau tanda identifikasi lainnya yang

ditulis pada dokumen sebagai sebuah bukti dari identitas. Tanda tangan berlaku sebagai segel.

2.4 Jaringan Saraf Tiruan (*Artificial Neural Network*)

2.4.1 Pengertian

Jaringan Saraf Tiruan didefinisikan sebagai susunan dari elemen-elemen penghitung yang disebut *neuron* atau titik (*node*) yang saling terhubung guna dimodelkan untuk meniru fungsi otak manusia.

Sistem jaringan saraf tiruan dicirikan dengan adanya proses pembelajaran (*learning/training*) yang berfungsi untuk mengadaptasi parameter-parameter jaringannya.

Menurut Zurada, J.M. (1992), *Introduction To Artificial Neural Systems*, Boston: PWS Publishing Company, mendefinisikan sebagai berikut:

“Sistem saraf tiruan atau jaringan saraf tiruan adalah sistem selular fisik yang dapat memperoleh, menyimpan dan menggunakan pengetahuan yang didapatkan dari pengalaman”.

Jaringan saraf tiruan, seperti manusia, belajar dari suatu contoh karena mempunyai karakteristik yang adaptif, yaitu dapat belajar dari data-data sebelumnya dan mengenal pola data yang selalu berubah. Selain itu, JST merupakan sistem yang tak terprogram, artinya semua keluaran atau kesimpulan yang ditarik oleh jaringan

didasarkan pada pengalamannya selama mengikuti proses pembelajaran/pelatihan.

Jaringan saraf tiruan ditentukan oleh beberapa hal, yaitu :

1. Pola hubungan antar *neuron* (disebut arsitektur jaringan)
2. Metode untuk menentukan bobot penghubung (disebut metode *training/learning*)
3. Fungsi aktivasi, yaitu fungsi yang digunakan untuk menentukan keluaran suatu *neuron*

Jaringan saraf tiruan terdiri dari :

1. Lapisan masukan (*Input Layer*)

Berisi node – node yang masing – masing menyimpan nilai masukan yang tidak berubah pada fase pelatihan dan hanya bisa berubah jika diberikan nilai masukan baru.

2. Lapisan tersembunyi atau lapisan dalam (*Hidden Layer*)

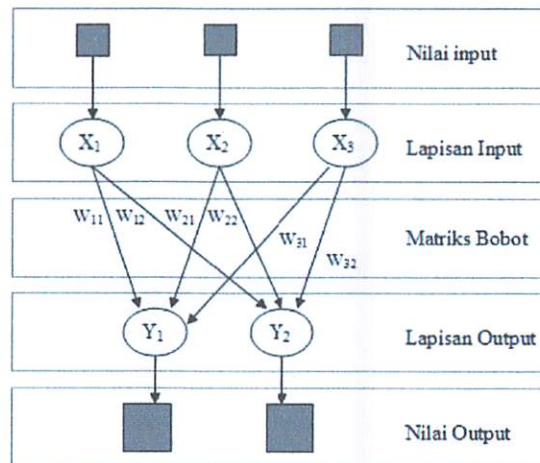
Lapisan ini terletak di antara *input layer* dan *output layer*. Lapisan ini tidak pernah muncul sehingga dinamakan *hidden layer*. Akan tetapi semua proses pada fase pelatihan dan fase pengenalan dijalankan pada lapisan ini. Jumlahnya melebihi 2 lapisan tergantung keinginan pembuat program untuk menetapkan jumlah *hidden layer* maupun jumlah nodenya. Konsekuensi dari adanya lapisan ini adalah pelatihan menjadi makin sulit atau lama.

3. Lapisan keluaran (*Output Layer*)

Berfungsi untuk menyalurkan sinyal – sinyal keluaran hasil pemrosesan jaringan.

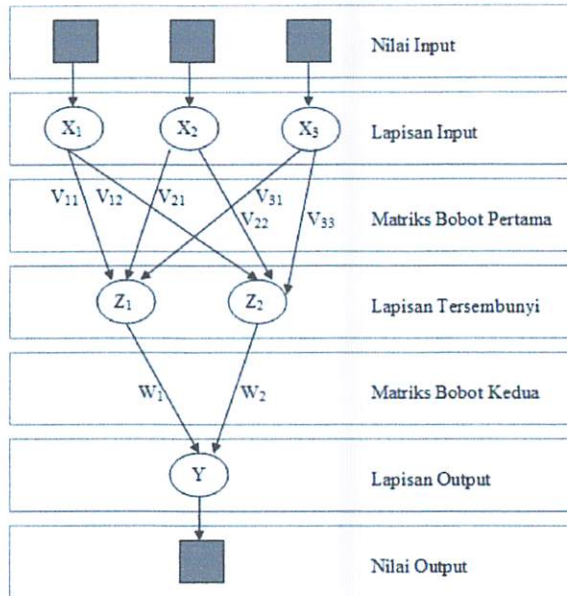
2.4.2 Arsitektur Jaringan

- a. Jaringan dengan lapisan tunggal (*single layer net*)
- Hanya memiliki satu lapisan dengan bobot-bobot terhubung
 - Jaringan ini hanya menerima *input* kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi.



Gambar 2.6
Jaringan dengan Lapisan Tunggal (*single layer net*)

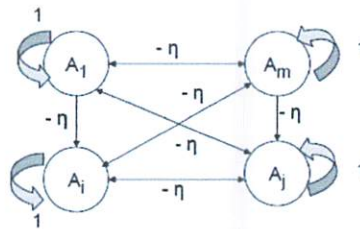
- b. Jaringan dengan banyak lapisan (*multilayer net*)
- Memiliki satu atau lebih lapisan yang terletak diantara lapisan *input* dan lapisan *output*
 - Ada lapisan yang berbobot yang terletak antara 2 lapisan yang bersebelahan



Gambar 2.7 Jaringan dengan Banyak Lapisan (*multilayer net*)

c. Jaringan dengan lapisan kompetitif (*competitive layer net*)

Hubungan antar neuron pada lapisan kompetitif tidak diperlihatkan pada diagram arsitektur.

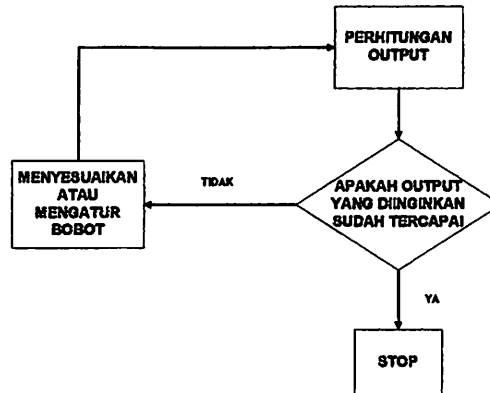


Gambar 2.8 Jaringan dengan Lapisan Kompetitif (*competitive layer net*)

2.4.3 Metode Pelatihan / Pembelajaran

Pelatihan/pembelajaran Jaringan Saraf Tiruan (JST) dimaksudkan untuk mencari bobot- bobot yang terdapat dalam layer. Proses umum dari pembelajaran, meliputi:

1. Perhitungan output
2. Membandingkan output dengan target yang diinginkan
3. Menyesuaikan bobot dan mengulangi prosesnya



Gambar 2.9
Diagram Alir Proses Pembelajaran Jaringan Saraf Tiruan

Cara berlangsungnya pembelajaran atau pelatihan JST dikelompokkan menjadi 3, yaitu :

a. *Supervised learning* (pembelajaran terawasi)

Setiap pola yang diberikan ke dalam JST, telah diketahui *output* - nya. Selisih antara pola *output actual* (*output* yang dihasilkan) dengan pola *output* yang dikehendaki (*output target*) yang disebut *error* digunakan untuk mengoreksi bobot JST sehingga JST mampu menghasilkan *output* sedekat mungkin dengan pola target yang telah diketahui oleh JST. Contoh algoritma JST yang menggunakan metode ini adalah : *Hebb Rule*, *Perceptron*, *Delta Rule*, *Backpropagation*, dll.

b. *Unsupervised learning* (pembelajaran tak terawasi)

Pada metode ini, tidak memerlukan target *output*. Pada metode ini tidak dapat ditentukan hasil seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu *range* tertentu tergantung pada nilai *input* yang diberikan. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran ini biasanya sangat cocok untuk klasifikasi pola. Contoh algoritma JST yang menggunakan metode ini adalah : *Competitive, Kohonen, Neocognization*, dll.

c. *Hybrid Learning* (pembelajaran hibrida)

Merupakan kombinasi dari metode pembelajaran *supervised learning* dan *unsupervised learning*. Sebagian dari bobot-bobotnya ditentukan melalui pembelajaran terawasi dan sebagian lainnya melalui pembelajaran tak terawasi. Contoh algoritma JST yang menggunakan metode ini, yaitu algoritma RBF.

2.4.4 Fungsi Aktivasi

Fungsi aktivasi adalah fungsi yang mengolah data *input* menjadi data *output*. Fungsi ini biasanya berupa fungsi pemampat (*Squashing Function*).

1. Fungsi Sigmoid Biner

Fungsi ini digunakan untuk jaringan saraf yang dilatih dengan menggunakan metode *backpropagation*. Fungsi *sigmoid biner* memiliki nilai pada *range* 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan saraf yang membutuhkan nilai *output* yang terletak pada *interval* 0 sampai 1. Namun, fungsi ini bisa juga digunakan oleh jaringan saraf yang nilai *output*-nya 0 dan 1. Fungsi *sigmoid biner* dirumuskan :

$$y = f(x) = \frac{1}{(1 + e^{-ax})} \quad (2.1)$$

Dengan turunan pertama dari persamaan 2.1 :

$$f'(x) = af(x)[1 - f(x)] \quad (2.2)$$

2. Fungsi Sigmoid Bipolar

Fungsi *sigmoid bipolar* hampir sama dengan fungsi *sigmoid biner*, hanya saja *output* dari fungsi ini memiliki *range* antara 1 sampai -1. Fungsi tersebut dirumuskan :

$$y = f(x) = \frac{1 - e^{-ax}}{(1 + e^{-ax})} \quad (2.3)$$

Dengan turunan pertama dari persamaan 2.3 :

$$f'(x) = \frac{\sigma}{2} [1 + f(x)][1 - f(x)] \quad (2.4)$$

2.4.5 Propagasi Balik (*Backpropagation*)

Propagasi balik atau *backpropagation* merupakan suatu teknik pembelajaran/pelatihan *supervised learning*. Metode ini merupakan suatu metode pelatihan yang meminimalisasikan total *error* pada keluaran yang dihitung oleh jaringan. Pada jaringan propagasi balik, selain terdapat *input layer*, *hidden layer*, dan *output layer* juga terdapat bias yang diberikan pada unit-unit tersembunyi dan output

Pada jaringan propagasi balik, sebuah sampel tanda tangan disebarkan dan menghasilkan sebuah keluaran. Keluaran ini dibandingkan dengan target yang diinginkan. Jika keluaran tidak sesuai dengan target, maka akan dihasilkan nilai *error*. Nilai *error* ini akan digunakan dalam komputansi untuk memperbarui nilai bobot pada *hidden layer* dan *input layer*. Tiap *neuron* akan menyesuaikan besar bobot untuk menghasilkan nilai *error* yang lebih kecil. Proses ini dinamakan proses *backforward* dan dilakukan berulang-ulang sampai keluaran sesuai dengan target yang diharapkan.

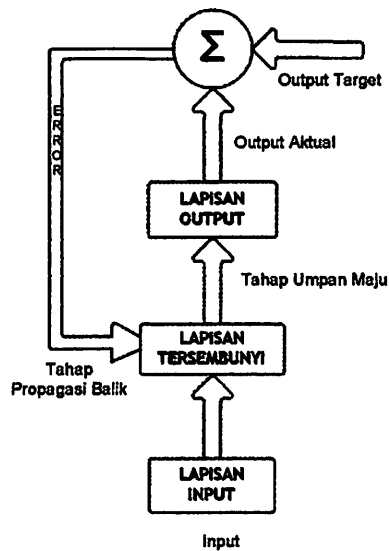
Algoritma propagasi balik dapat di bagi ke dalam 2 bagian :

1. Algoritma pelatihan

Terdiri dari 3 tahap : tahap umpan maju (*feedforward*) pola pelatihan input, tahap perhitungan dan backpropagasi kembali dari kumpulan kesalahan (*error backpropagation*), dan tahap pengaturan *error* / pembaharuan (*adjustment*) bobot dan bias.

2. Algoritma aplikasi

Yang digunakan hanyalah tahap umpan maju (*feedforward*).



Gambar 2.10
Metode Propagasi Balik

Cara kerja propagasi balik seperti berikut :

Mula – mula, jaringan diinisialisasi dengan bobot yang diset dengan bilangan acak. Kemudian, contoh - contoh pelatihan dimasukkan ke dalam jaringan. Contoh pelatihan terdiri dari pasangan vektor *input* dan vektor *output* target. Keluaran dari jaringan berupa sebuah vektor *output* aktual. Selanjutnya vektor *output* aktual jaringan dibandingkan dengan vektor *output* target untuk mengetahui apakah *output* jaringan sudah sesuai dengan harapan (*output* aktual sudah sama dengan output target).

Error yang timbul akibat perbedaan antara *output* aktual dengan *output* target tersebut kemudian dihitung dan digunakan

untuk meng-*update* bobot – bobot yang relevan dengan jalan mempropagasikan kembali *error*. Setiap perubahan bobot yang terjadi diharapkan dapat mengurangi besaran *error*. Epoch (siklus setiap pola pelatihan) seperti ini dilakukan pada semua set pelatihan sampai unjuk kerja jaringan mencapai tingkat yang diinginkan atau sampai kondisi berhenti di sini, misalnya : pelatihan akan dihentikan setelah epoch mencapai 1000 kali, atau pelatihan akan dihentikan sampai sebuah nilai ambang yang ditetapkan terlampaui. Setelah proses pelatihan selesai, barulah diterapkan algoritma aplikasi. Biasanya sebelum digunakan untuk aplikasi yang sebenarnya, pengujian unjuk kerja jaringan dilakukan dengan cara memasukkan set pengujian (set tes) ke dalam jaringan. Karena bersifat untuk menguji, set pengujian hanya berupa *input* saja. Dari respons jaringan dapat dinilai kemampuan memorisasi dan generalisasi jaringan dalam menebak *output* berdasarkan pada apa yang telah dipelajarinya selama ini.

2.4.6 Algoritma Propagasi Balik (*Backpropagation*)

Backpropagation biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot – bobot yang terhubung dengan *neuron-neuron* yang ada pada bagian tersembunyi. Pada dasarnya, *perceptron* pada jaringan saraf tiruan dengan satu lapisan memiliki bobot yang bisa diatur dalam suatu

nilai ambang (*threshold*). Algoritma yang digunakan oleh aturan *perceptron* ini akan mengatur parameter- parameter bebasnya melalui proses pembelajaran.

Algoritma *backpropagation* menggunakan *error output* untuk mengubah nilai bobot – bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, *neuron – neuron* diaktifkan dengan menggunakan fungsi aktivasi *sigmoid* atau biner *sigmoid*.

Pada dasarnya algoritma *backpropagation* terbagi menjadi tiga tahap, yaitu tahap umpan maju (*feedforward*) pola pelatihan *input*, tahap perhitungan dan backpropagasi kembali dari kumpulan kesalahan (*error backpropagation*), dan tahap pengaturan *error* / perubahan nilai bobot. Pada saat *feedforward*, perhitungan nilai bobot *neuron* hanya didasarkan pada vektor masukan, sedangkan pada *backpropagation*, nilai bobot diperhalus dengan memperhitungkan nilai target atau keluaran.

Algoritma:

- *Input* : data pelatihan, jumlah *layer*, jumlah *neuron*, target *error*, *learning rate* (α)
- *Output*: model JST yang siap mengolah data baru

1. Inisialisasi bobot dan bias.

Baik bobot maupun bias dapat di-set dengan sembarang angka (acak) dan biasanya angka di sekitar 0 dan 1 atau -1 (bias positif atau negatif). Dan tentukan angka pembelajaran (α), nilai target *error* atau nilai ambang (*threshold*), bila menggunakan nilai ambang sebagai kondisi berhenti. Atau set maksimal epoch (bila menggunakan banyaknya epoch sebagai kondisi berhenti).

2. Jika kondisi berhenti (*stop condition*) masih belum terpenuhi (Epoch < maksimum Epoch dan MSE > target *error*), jalankan langkah ke-3 sampai langkah ke-10.
3. Untuk setiap data pelatihan (*training*), jalankan langkah ke-4 sampai langkah ke-9.

Tahap *Feedforward*

4. Tiap-tiap unit *input* ($X_i, i = 1,2,3,\dots,n$) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan tersembunyi (*Hidden layer*). Input x_i yang dipakai adalah input training data.
5. Tiap-tiap unit tersembunyi ($Z_j, j = 1,2,3,\dots,p$) menjumlahkan sinyal – sinyal *input* berbobot, termasuk biasnya :

$$Z_{in\ j} = V_{o\ j} + \sum_{i=1}^n x_i v_{ij} \quad (2.6)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output* dari *hidden* unit yang bersangkutan :

$$Z_j = f(z_{in_j}) \quad (2.7)$$

Dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit – unit *output*).

6. Tiap – tiap unit output (Y_k , $k = 1,2,3,\dots,m$) menjumlahkan sinyal – sinyal *input* berbobot, termasuk biasanya.

$$y_{in_k} = w_{o_k} + \sum_{i=1}^p Z_i w_{jk} \quad (2.8)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output* dari unit output yang bersangkutan :

$$y_k = f(y_{in_k}) \quad (2.9)$$

Dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit – unit *output*).

Tahap *Backpropagation of Error*

7. Tiap – tiap unit output (Y_k , $k = 1,2,3,\dots,m$) menerima target pola (*desired output*) yang berhubungan dengan pola *input* pelatihan, kemudian hitung informasi *error*nya, antara target dengan output yang dihasilkan jaringan :

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (2.10)$$

Faktor δ_k ini digunakan untuk menghitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai W_{jk}):

$$\Delta W_{jk} = \alpha \delta_k Z_j \quad (2.11)$$

Hitung juga koreksi bias ΔW_{ok} (yang nantinya akan digunakan untuk memperbaiki nilai W_{ok}) :

$$\Delta W_{ok} = \alpha \delta_k \quad (2.12)$$

Kirimkan δ_k ini ke unit – unit yang ada di lapisan bawahnya:

8. Tiap – tiap unit tersembunyi (Z_j , $j = 1,2,3,\dots,p$) menjumlahkan delta *inputnya* (dari unit- unit yang berada pada lapisan di atasnya) yang sudah berbobot :

$$\delta_{in_j} = \sum_{k=1}^m \delta_k W_{jk} \quad (2.13)$$

Kalikan nilai ini dengan turunan dari fungsi aktifasinya untuk menghitung faktor koreksi *error* δ_j :

$$\delta_j = \delta_{in_j} f'(Z_{in_j}) \quad (2.14)$$

Kemudian hitung koreksi bobot ΔV_{ij} (yang nantinya akan digunakan untuk memperbaiki nilai V_{ij}) :

$$\Delta V_{ij} = \alpha \delta_j x_i \quad (2.15)$$

Hitung juga koreksi bias ΔV_{oj} (yang nantinya akan digunakan untuk memperbaiki nilai V_{oj}) :

$$\Delta V_{oj} = \alpha \delta_j \quad (2.16)$$

Tahap *Adjustment* (Pembaharuan Bobot dan Bias)

9. Tiap – tiap unit *output* (Y_k , $k = 1,2,3,\dots,m$) memperbaharui bias dan bobotnya dari setiap *hidden* unit ($j = 0,1,2,\dots,p$) :

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad (2.17)$$

Tiap – tiap unit tersembunyi (Z_j , $j = 1,2,3,\dots,p$) memperbaharui bias dan bobotnya dari setiap unit *input* ($i = 0,1,\dots,n$) :

$$V_{ij} (\text{baru}) = V_{ij} (\text{lama}) + \Delta V_{ij} \quad (2.18)$$

10. Periksa *stopping condition*

Jika *stopping condition* terpenuhi, maka pelatihan jaringan dapat dihentikan.

Untuk menentukan *stopping condition*, yaitu :

- a. Pertama, dengan membatasi iterasi yang dilakukan. Misalnya, jaringan akan dilatih sampai pada iterasi yang ke-500. Yang dimaksud satu iterasi di sini adalah perulangan langkah ke-4 sampai dengan langkah ke-9 untuk semua training data yang ada.
- b. Cara kedua adalah membatasi *error*. Untuk metode *backpropagation*, dipakai metode *Mean Square Error* untuk menghitung rata-rata *error* antara output yang dikehendaki pada *training* data dengan *output* yang dihasilkan oleh jaringan. Misalnya, *error* telah mencapai 0,01 (1%), pelatihan dihentikan. Besarnya persen *error* ini tergantung kepresisian yang dibutuhkan oleh sistem yang bersangkutan.

Selain kedua cara di atas, sebuah pertimbangan lain yang dapat dipakai untuk menghentikan pelatihan. Sebelum mencapai kondisi yang diinginkan, *error* justru semakin besar. Kejadian seperti ini disebut *overtraining*. Kondisi *error* yang diperhatikan

di sini tidak hanya *error* dari training set, tapi juga dari test set. Jika salah satu saja dari training set error atau test set error bertambah besar, maka pelatihan harus dihentikan.

Cara memeriksa *stopping condition* dengan *Mean Square Error* (MSE):

- a. Dengan bobot yang ada saat itu, jalankan langkah ke-4 sampai langkah ke-6), di mana inputnya diambil dari input training set jika yang ingin dihitung adalah *training set error* dan atau input *test* set jika yang ingin dihitung adalah test set *error*. Langkah ini diambil untuk semua data *training/test* yang ada.
- b. Kemudian dicari selisih antara target *output* (t_k) dengan *output* jaringan (y_k) dan diimplementasikan pada persamaan *Mean Square Error*. Jika terdapat m training data, maka:

$$\text{Mean Square Error} = E = 0.5 \times \{(t_{k1} - y_{k1})^2 + (t_{k2} - y_{k2})^2 + \dots + (t_{kt} - y_{kt})^2\} \quad (2.19)$$

Algoritma Aplikasi

1. Inisialisasi bobot. Bobot ini diambil dari bobot-bobot terakhir yang diperoleh dari algoritma pelatihan di atas.
2. Untuk setiap input x_i ($i = 1, \dots, n$) menerima sinyal input pengujian x_i dan meneruskan sinyal x_i ke semua unit pada lapisan di atasnya (unit-unit tersembunyi):

3. Tiap-tiap unit tersembunyi (Z_j , $j = 1,2,3,\dots,p$) menjumlahkan sinyal– inyal *input* berbobot, termasuk biasnya :

$$Z_{in_j} = V_{o_j} + \sum_{i=1}^n x_i v_{ij} \quad (2.6)$$

Gunakan fingsi aktifasi untuk menghitung sinyal *output* dari *hidden* unit yang bersangkutan :

$$Z_j = f(z_{in_j}) \quad (2.7)$$

Dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit– nit *output*).

4. Tiap – tiap unit output (Y_k , $k = 1,2,3,\dots,m$) menjumlahkan sinyal – sinyal *input* berbobot, termasuk biasnya.

$$y_{in_k} = W_{o_k} + \sum_{i=1}^p Z_i W_{jk} \quad (2.8)$$

Gunakan fungsi aktifasi untuk menghitung simnyal *output* dari unit output yang bersangkutan :

$$y_k = f(y_{in_k}) \quad (2.9)$$

Variabel y_k adalah output yang masih dalam skala menurut range fungsi aktivasi.

Keterangan simbol – simbol pada algoritma jaringan saraf tiruan *backpropagation* dijelaskan pada tabel di bawah ini :

Tabel 2.1
Simbol – Simbol Algoritma *Backpropagation*

SIMBOL	KETERANGAN
x	Data training, $x = (x_1, \dots, x_i, \dots, x_n)$
t	Data training untuk output(target/desired output), $t = (t_1, \dots, t_k, \dots, t_m)$
α	Learning rate, yaitu parameter yang mengontrol perubahan bobot selama pelatihan. Jika learning rate besar, jaringan semakin cepat belajar, tetapi hasilnya kurang akurat. Learning biasanya dipilih antara 0 sampai 1
x_i	Unit <i>input</i> ke-i. untuk unit <i>input</i> , sinyal yang masuk dan keluar pada suatu unit dilambangkan dengan variabel yang sama, yaitu x_i .
Z_j	<i>Hidden</i> unit ke-j. sinyal input pada Z_j , dilambangkan dengan z_in_j . Sinyal <i>output</i> (aktivasi) untuk Z_j dilambangkan dengan z_j .
v_{oj}	Bias untuk <i>hidden</i> unit ke-j.
v_{ij}	Bobot antara unit <i>input</i> ke-i dan <i>hidden</i> unit ke-j.
Y_k	Unit <i>output</i> ke-k. Sinyal input ke Y_k dilambangkan y_in_k . Sinyal <i>output</i> (aktivasi) untuk Y_k dilambangkan dengan y_k .
w_{ok}	Bias untuk unit <i>output</i> ke-k.
w_{jk}	Bias pada unit tersembunyi k
δ_k	Faktor koreksi <i>error</i> untuk bobot w_{jk} .
δ_j	Faktor koreksi <i>error</i> untuk bobot v_{ij} .

Pelatihan jaringan saraf tiruan dikatakan berhasil jika pelatihan *konvergen*, dan gagal jika pelatihan *divergen*. Suatu pelatihan dikatakan *konvergen*, jika kesalahan pada setiap iterasi pelatihan selalu mengecil, sampai pada titik di mana nilai bobot pada setiap *neuron* telah mencapai nilai yang paling baik untuk data pelatihan yang diberikan. Sebaliknya, pelatihan dikatakan *divergen*, jika

kesalahan pada pelatihan tidak cenderung mengecil menuju sebuah titik tertentu.

2.5 Citra Digital

Citra (gambar) mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Ukuran citra dapat juga dinyatakan secara fisik dalam satuan panjang (misalnya mm atau *inch*). Sehingga harus ada hubungan antara ukuran titik (*pixel*) penyusun citra dengan satuan panjang. Hal tersebut dinyatakan dengan resolusi yang merupakan ukuran banyaknya titik untuk setiap satuan panjang. Biasanya satuan yang digunakan adalah dpi (*dot per inch*).

Citra sebagai keluaran suatu sistem perekaman data dapat bersifat *optic* berupa foto, bersifat analog berupa sinyal-sinyal video seperti citra pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu pita *magnetic*.

Citra digital merupakan suatu array dua dimensi atau suatu matriks yang elemen-elemennya menyatakan tingkat keabuan dari elemen citra. Citra digital tidak selalu merupakan hasil langsung data rekaman suatu sistem. Terkadang hasil rekaman data bersifat kontinu seperti foto sinar-X, citra hasil pemindaian dll. Untuk mendapatkan suatu citra digital diperlukan suatu proses konversi sehingga citra tersebut selanjutnya dapat diproses komputer.

Sebuah citra diubah ke bentuk digital agar dapat disimpan dalam memori komputer atau media lain. Proses mengubah citra ke bentuk digital bisa dilakukan dengan beberapa perangkat, misalnya *scanner*, kamera digital, dan *handycam*. Ketika sebuah citra sudah diubah ke dalam bentuk digital (selanjutnya disebut citra digital), bermacam-macam proses pengolahan citra dapat diperlakukan terhadap citra tersebut.

2.6 Pengolahan Citra (*Image Processing*)

Pengolahan citra (*image processing*) merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual, yang masukannya berupa citra dan keluarannya juga citra tapi dengan kualitas lebih baik daripada citra masukan. Misal citra berwarna kurang tajam, kabur (*blurring*), mengandung *noise* (misal bintik-bintik putih), dll sehingga perlu ada pemrosesan untuk memperbaiki citra karena citra tersebut menjadi sulit diinterpretasikan sebab informasi yang disampaikan menjadi berkurang. Tujuan dari pengolahan citra adalah memperbaiki informasi pada gambar sehingga mudah terbaca atau memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia/mesin (komputer).

2.6.1 Teknik – Teknik Pengolahan Citra Digital

Secara umum, teknik pengolahan citra dibedakan menjadi tiga tingkat pengolahan, yaitu :

- Tahap 1 *Low Level Preprocessing* merupakan pengolahan citra paling dasar dalam pengolahan citra, sebagai contohnya: pengurangan *noise* (*noise reduction*), perbaikan citra (*image enhancement*) dan resolusi citra (*image restoration*).
- Tahap 2 *Mid Level Processing* pengolahan citra ini meliputi segmentasi pada citra, deteksi objek, dan klasifikasi objek secara terpisah.
- Tahap 3 *High Level Processing* merupakan tahap analisa citra.

Dari ketiga tahap diatas, dapat dinyatakan suatu gambaran mengenai teknik-teknik pengolahan citra digital.

- Image Enhancement* (perbaikan kualitas citra), berupa proses perbaikan citra dengan meningkatkan kualitas citra. Misal, perbaikan kontras gelap/terang, perbaikan tepian objek (*edge enhancement*), penajaman (*sharpening*), pemberian warna semu (*pseudocoloring*), penapisan derau (*noise filtering*).
- Image Restoration* (pemugaran citra), dengan tujuan menghilangkan cacat pada citra. Misal, penghilangan kesamaran (*deblurring*), penghilangan derau (*noise*).
- Color Image Processing*, berhubungan dengan citra berwarna baik itu *image enhancement*, *image restoration*, atau yang lain.
- Wavelet dan Multiresolution Processing*, proses yang menyatakan citra dalam beberapa resolusi.

- e. *Image Compression* (pemampatan citra), merupakan proses yang digunakan untuk mengubah ukuran data pada citra. Tujuannya adalah citra direpresentasikan dalam bentuk yang lebih kompak, sehingga keperluan memori lebih sedikit namun dengan tetap mempertahankan kualitas gambar (misal dari .BMP menjadi .JPG).
- f. *Morphological Processing*, proses untuk memperoleh informasi yang menyatakan deskripsi dari suatu bentuk pada citra.
- g. *Segmentation*, proses untuk membedakan atau memisahkan objek-objek yang ada dalam suatu citra, seperti memisahkan objek dengan *background*-nya. Bagian ini bertujuan untuk memecah suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu. Jenis operasi ini berkaitan erat dengan pengenalan pola.

Ada 2 macam segmentasi, yaitu *full segmentation* dan *partial segmentation*. *Full segmentation* adalah pemisahan suatu *object* secara individu dari *background*. *Partial segmentation* adalah pemisahan sejumlah data dari *background* dimana data yang disimpan hanya data yang dipisahkan saja untuk mempercepat proses selanjutnya. Ada 3 tipe dari segmentasi yaitu:

- *classification-based*: segmentasi berdasarkan kesamaan suatu ukuran dari nilai *pixel*. Salah satu cara paling mudah adalah *thresholding*. *Thresholding* ada 2 macam yaitu global dan lokal. Pada *thresholding* global, segmentasi berdasarkan pada sejenis histogram. Pada *thresholding* lokal, segmentasi dilakukan

berdasarkan posisi pada gambar, gambar dibagi menjadi bagian-bagian yang saling melengkapi, jadi sifatnya dinamis.

- *edge-based*: mencari garis yang ada pada gambar dan garis tersebut digunakan sebagai pembatas dari tiap segmen.
- *region-based*: segmentasi dilakukan berdasarkan kumpulan *pixel* yang memiliki kesamaan (tekstur, warna atau tingkat warna abu-abu) dimulai dari suatu titik ke titik-titik lain yang ada disekitarnya.

h. *Objek Recognition*, proses untuk mengetahui ada tidaknya objek dalam citra.

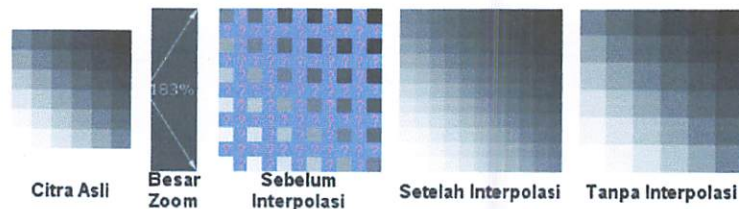
2.6.2 Interpolasi Citra Digital

Interpolasi berkaitan erat dengan proses pemetaan piksel-piksel baik secara *forward* maupun *reverse*. Algoritma-algoritma pengolahan citra yang menerapkan proses interpolasi antara lain adalah algoritma penskalaan (pembesaran atau *digital zoom*), rotasi citra serta proses-proses geometris dan kreatif lainnya

Interpolasi (kadang-kadang disebut *resampling*) merupakan sebuah metode pencitraan untuk meningkatkan (atau mengurangi) jumlah piksel dalam gambar digital. Beberapa kamera digital menggunakan interpolasi untuk menghasilkan gambar yang lebih besar daripada sensor ditangkap atau untuk membuat zoom digital. Hampir semua perangkat lunak *editing* gambar mendukung

satu atau lebih metode interpolasi. Bagaimana gambar yang diperbesar bisa halus tanpa meninggalkan kesan pecah sangat tergantung pada kecanggihan algoritma interpolasi tersebut.

Interpolasi citra digital bekerja secara dua arah. Proses ini berusaha untuk mendapatkan hasil perkiraan nilai piksel warna dan intensitas yang terbaik berdasarkan nilai pada piksel-piksel di sekitarnya (*pixel surrounding/pixer neighborhood*). Citra asli adalah citra *gray-level* 8-bit yang akan diperbesar 183%. Sebelum interpolasi dilakukan, proses pembesaran akan menghasilkan *hole* (ditandai dengan ?). Setelah dilakukan interpolasi maka *hole* akan diisi dengan nilai warna dan intensitas sesuai dengan algoritma interpolasi yang digunakan. Sebagai pembanding, gambar di paling kanan adalah hasil citra jika tidak dilakukan interpolasi, tetapi hanya dilakukan perbesaran dengan menggunakan teknik replikasi piksel. Contoh berikut menggambarkan cara mengubah ukuran / pembesaran bekerja:



Gambar 2.11
Ilustrasi Interpolasi Citra Digital

Nilai-nilai *pixel* dapat berubah lebih jauh dari satu lokasi ke lokasi berikutnya. Semakin banyak yang diketahui tentang informasi

di sekitar *pixel*, semakin baik dalam proses interpolasi. Oleh karena itu semakin diperbesar, semakin banyak informasi yang harus diketahui.

Terdapat empat metode interpolasi yang umum dan dapat dimanfaatkan untuk pengolahan citra digital, yaitu metode *nearest-neighbour*, metode *bilinier*, metode *bicubic* dan metode *fractal*.

- Interpolasi Bilinear

Algoritma Bilinear Interpolation membuat pixel baru dengan cara mengambil rata-rata 4 pixel (2 x 2) tetangganya yang terdekat dari original image. Hasil rata-rata ini mengaplikasikan efek anti-aliasing yang hasil pembesaran tersebut terlihat lebih halus dan hampir tidak terlihat adanya *jaggies*.

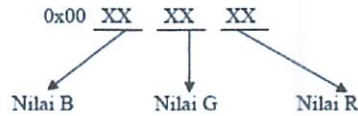
2.6.3 Cropping

Cropping adalah memotong satu bagian dari citra sehingga diperoleh citra yang berukuran lebih kecil. Operasi ini pada dasarnya adalah operasi translasi, yaitu menggeser koordinat titik citra.

2.6.4 Warna

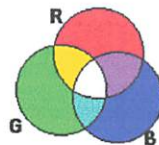
Dasar dari pengolahan citra adalah pengolahan warna RGB pada posisi tertentu. Sebuah citra berwarna umumnya lebih sulit untuk diproses karena sebuah warna tidak dapat diwakilkan oleh sebuah angka saja. Selain itu terkadang manusia memiliki persepsi

warna yang rumit. Dalam pengolahan citra warna dipresentasikan dengan nilai hexadesimal dari 0x00000000 sampai 0x00ffffff. Warna hitam adalah 0x00000000 dan warna putih adalah 0x00ffffff. Definisi nilai warna di atas seperti gambar di bawah, variabel 0x00 menyatakan angka dibelakangnya adalah *hexadecimal*.



Gambar 2.12
Nilai Warna RGB dalam *Hexadecimal*

Terlihat bahwa setiap warna mempunyai range nilai 00 (angka desimalnya adalah 0) dan ff (angka desimalnya adalah 255), atau mempunyai nilai derajat keabuan $256 = 2^8$. Dengan demikian range warna yang digunakan adalah $(2^8)(2^8)(2^8) = 2^{24}$ (atau yang dikenal dengan istilah *True Colour* pada Windows). Nilai warna yang digunakan di atas merupakan gabungan warna cahaya merah, hijau dan biru seperti yang terlihat pada gambar 2.21. Sehingga untuk menentukan nilai dari suatu warna yang bukan warna dasar digunakan gabungan skala kecerahan dari setiap warnanya.



Gambar 2.13
Komposisi Warna RGB

Dari definisi diatas untuk menyajikan warna tertentu dapat dilakukan dengan mencampurkan ketiga warna dasar RGB.

Tabel 2.2
Contoh-Contoh Warna dalam Hexadesimal

NILAI	WARNA	NILAI	WARNA
0x00000000	Hitam	0x0000A0FF	Orange
0x000000FF	Merah	0x00888888	Abu-abu
0x0000FF00	Hijau	0x00FF00AA	Ungu
0x00FF0000	Biru	0x00A0FF00	Hijau Muda
0x0000FFFF	Kuning	0x00AA00FF	Merah Muda
0x00FF00FF	Magenta	0x00A0FFFF	Kuning Muda
0x00FFFF00	Cyan	0x000088AA	Coklat
0x00FFFFFF	Putih	0x00AA0088	Ungu

2.6.5 Ekstrasi Ciri

Ekstrasi merupakan proses untuk mengubah citra menjadi kode tertentu yang nantinya akan dimasukkan dalam proses jaringan saraf tiruan. Pengkodean ini berdasarkan nilai dari setiap *pixel* dalam citra. Jika *pixel* bernilai hitam , maka dikodekan menjadi 1. Apabila tidak, maka dikodekan menjadi angka 0.

2.6.6 Pixel

Pixel adalah unsur gambar atau representasi sebuah titik terkecil dalam sebuah gambar grafis yang dihitung per inci. *Pixel* sendiri berasal dari akronim bahasa Inggris *Picture Element* yang disingkat menjadi *Pixel*.

Salah satu komponen gambar yang menentukan resolusi dari gambar adalah *pixel*. Sebagai contoh suatu gambar memiliki

spesifikasi resolusi sebesar 400x300 hal ini berarti gambar ini memiliki panjang *pixel* horizontalnya 400 dan panjang *pixel* vertikal 300 dan jumlah total keseluruhan *pixel* gambar tersebut adalah 120000.

Pada ujung tertinggi skala resolusi, mesin cetak gambar berwarna dapat menghasilkan hasil cetak yang memiliki lebih dari 2.500 titik per inci dengan pilihan 16 juta warna lebih untuk setiap inci, dalam istilah komputer berarti gambar seluas satu inci persegi yang bisa ditampilkan pada tingkat resolusi tersebut sepadan dengan 150 juta bit informasi. Makin tinggi jumlahnya, maka makin tinggi juga resolusi gambarnya.

2.6.7 Dasar – Dasar Pengolahan Citra

2.6.7.1 Mengubah Citra Berwarna Menjadi *Grayscale*

Proses awal yang banyak dilakukan dalam *image processing* adalah mengubah citra berwarna menjadi citra *grayscale*. Digunakan untuk menyederhanakan model citra. Citra berwarna terdiri dari 3 *layer* matrik yaitu *R-layer*, *G-layer* dan *B-layer*. Untuk melakukan proses-proses selanjutnya tetap diperhatikan tiga *layer* di atas. Bila setiap proses perhitungan dilakukan menggunakan tiga *layer*, berarti dilakukan tiga perhitungan yang sama. Konsep itu disederhanakan, dengan mengubah 3 *layer* di atas menjadi 1

layer matrik *grayscale* Hasilnya adalah citra *grayscale*. Dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan. Mengubah citra berwarna yang mempunyai nilai matrik *r*, *g* dan *b* menjadi citra *grayscale* dengan nilai *s*, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai *r*, *g* dan *b*:

$$s = \frac{r + g + b}{3}$$

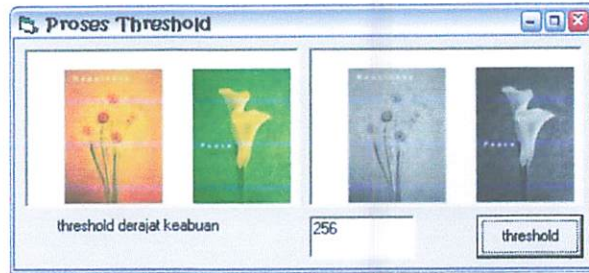
(2.19)



Gambar 2.14
Proses *Grayscale*

2.6.7.2 *Thresholding*

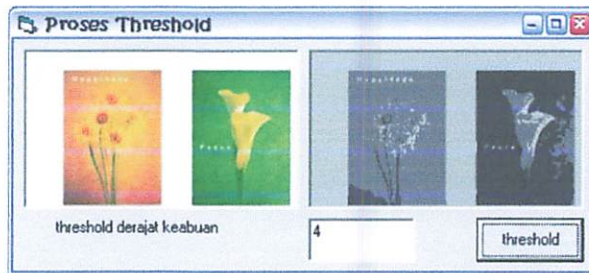
Thresholding digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. Pada Visual Basic adalah 256 atau 2^8 . Dengan *thresholding* maka derajat keabuan dapat diubah sesuai keinginan. Pada *thresholding* yang tinggi tidak tampak perbedaan karena keterbatasan mata, tetapi untuk *thresholding* tingkat rendah seperti 2, 4, 8, 16 tampak sekali perbedaannya.



Gambar 2.15
Proses *Thresholding* dengan Derajat Keabuan 256



Gambar 2.16
Proses *Thresholding* dengan Derajat Keabuan 16



Gambar 2.17
Proses *Thresholding* dengan Derajat Keabuan 4

2.6.7.3 Konversi ke Citra *Biner*

Citra *biner* (hitam-putih) merupakan citra yang banyak dimanfaatkan untuk keperluan *pattern recognition* yang sederhana. Misalnya, pengenalan angka atau pengenalan huruf. Untuk mengubah suatu citra *grayscale* menjadi citra *biner*, sebetulnya prosesnya sama dengan

threshold yaitu mengubah kuantisasi citra. Untuk citra dengan derajat keabuan 256, nilai tengahnya adalah 128, sehingga untuk mengubah menjadi citra *biner* dapat dituliskan:

Jika $x < 128$ maka $x = 0$, jika tidak maka $x = 255$



Gambar 2.18
Konversi ke *Biner*

BAB III

METODOLOGI DAN PERANCANGAN SISTEM

3.1 Gambaran Umum

Sistem terbagi atas dua macam tahap, yaitu tahap pelatihan dan tahap pengenalan. Pada tahap pelatihan, 3 (tiga) buah citra tanda tangan milik seseorang dipelajari dan hasil pembelajaran disimpan dalam tempat penyimpanan data (*data storage*) yang kemudian digunakan sebagai dasar pembandingan dalam tahap pengenalan. Sebuah citra tanda tangan akan diubah ke dalam bentuk biner yang akan dikonversi dalam bentuk matriks. Kemudian nilai bobot akan diberikan sebagai aktivasi awal untuk menyesuaikan nilai bobot agar nilai keluaran mencapai target yang ditentukan. Target yang dimaksudkan adalah nilai fungsi *sigmoid bipolar*.

Perangkat lunak yang digunakan dalam sistem ini adalah :

1. Microsoft Visual Basic 6.0
2. Microsoft Access XP

Sedangkan perangkat keras yang digunakan adalah:

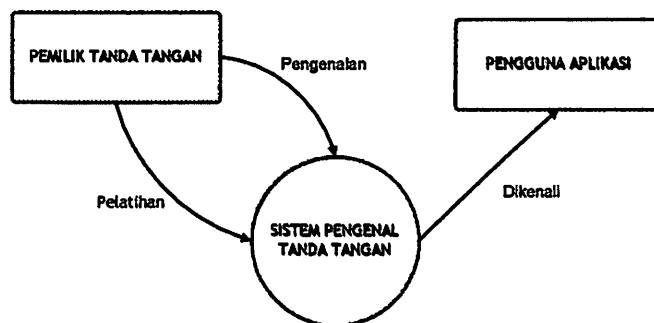
Prosesor	: Core 2 Duo 2 GHz
Memori	: DDR2 1 GB
Video Card	: ATI Rodeon HIS
Monitor	: 17" flat LG

3.2 Perancangan Sistem

Berdasarkan analisa yang akan dilakukan, berikut akan dibahas mengenai arsitektur dan proses yang terjadi pada perangkat lunak serta perancangan antarmuka sistem.

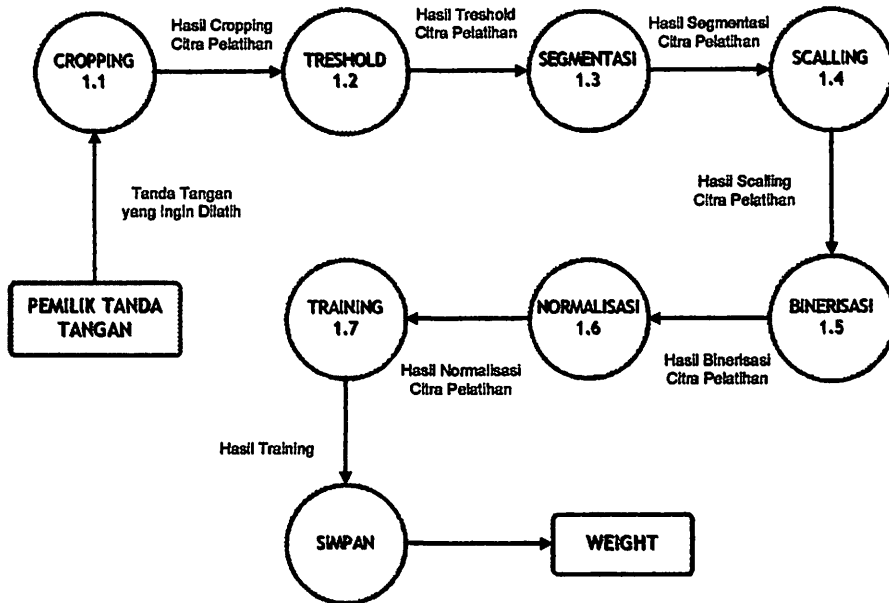
3.2.1 Perancangan Arsitektur dan Proses

1. Secara garis besar, alur sistem dapat digambarkan sebagai berikut:

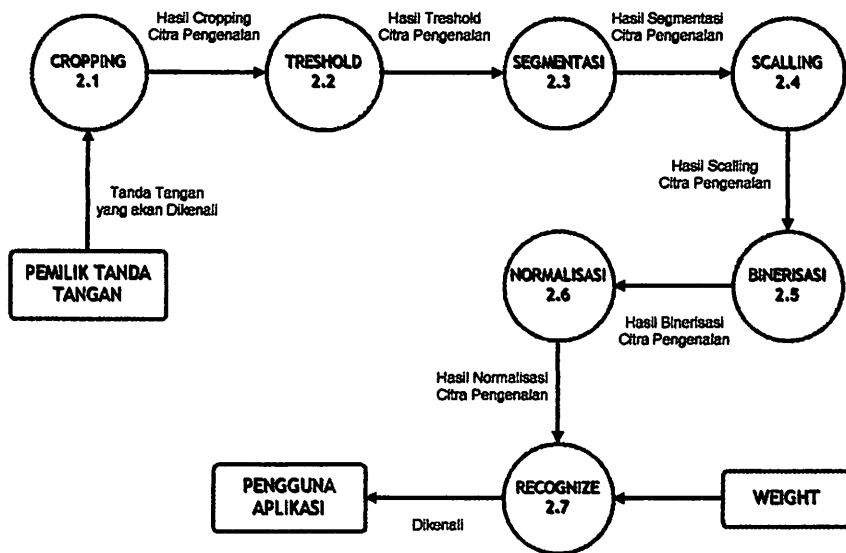


Gambar 3.1
Diagram Konteks

Dari gambar di atas, dapat dijelaskan bahwa sistem pertama kali akan menerima *input*-an dari user (pemilik tanda tangan). Input di sini berupa citra tanda tangan. Citra tersebut kemudian diolah sehingga citra dapat diproses. Citra yang telah diolah berupa biner yang selanjutnya dapat dihitung dengan jaringan saraf tiruan metode *backpropagation*. Kemudian, sistem akan menerima *input* yang kedua sebagai pembanding dari *input* pertama. Keluaran dari sistem ini adalah bobot hasil pelatihan dan pengenalan/pengujian.



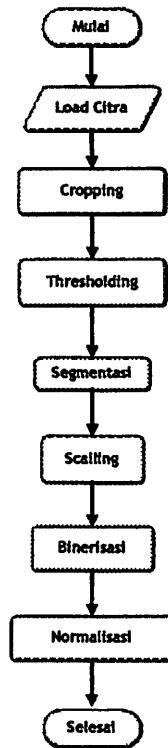
Gambar 3.2
Diagram Rinci Pelatihan



Gambar 3.3
Diagram Rinci Pengenalan

2. Pengolahan Citra

Pengolahan citra merupakan suatu proses yang sangat berpengaruh pada proses pelatihan maupun pembelajaran.



Gambar 3.4
Diagram Alir Pengolahan Citra

Dari gambar di atas dapat dijelaskan sebagai berikut:

1. *Load Citra*

Citra berupa tanda tangan di *load* atau diambil dari data *storage*. Citra tersebut merupakan dari *user/pemilik* tanda tangan yang membubuhkan tanda tangan pada sebuah kertas dan kemudian di *scan* untuk disimpan sehingga menghasilkan file berupa citra dengan format *.bmp (bitmap)*. Dengan tujuan, citra tersebut dapat diproses dalam tahap selanjutnya.

2. *Cropping*

Citra hasil *scan* kemudian di *cropping*, yaitu memotong citra kosong yang terlalu banyak sehingga nantinya citra tanda tangan yang kosong menjadi berkurang.

3. *Thresholding*

Kemudian dilakukan *thresholding*, yaitu digunakan untuk memisahkan tanda tangan dari latar belakangnya.

4. Segmentasi

Proses segmentasi merupakan proses pembatasan area gambar, sehingga akan menghasilkan sebuah citra yang menampilkan area yang terdapat objek tanda tangannya saja.

5. *Scalling*

Scalling merupakan proses perbesaran citra ke dalam bentuk normal atau bentuk asli pertama citra sama seperti citra sebelum diolah, yaitu ke ukuran 100x100 piksel. Proses *scalling* ini menggunakan interpolasi *bilinear*.

6. *Binerisasi*

Citra *biner* (hitam-putih) merupakan citra yang digunakan untuk keperluan *pattern recognition* yang sederhana. Citra diubah menjadi angka-angka biner tiap pikselnya melalui proses *binerisasi*.

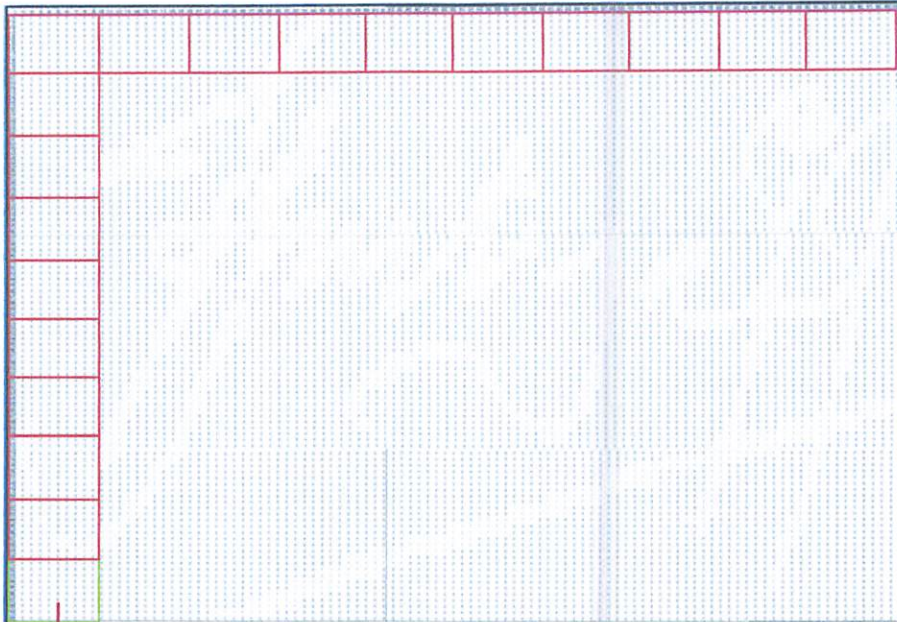
7. Normalisasi

Proses normalisasi merupakan proses yang menjadi $M \times N$ ukuran untuk setiap karakternya. Hasil perkalian $M \times N$ ini nantinya menjadi jumlah neuron yang ada pada input layer. Pada percobaan

yang nantinya akan dilakukan, diujikan beberapa $M \times N$ ukuran normalisasi dengan satuan *pixel*, dari matriks 100×100 kemudian dijadikan menjadi 10×100 dan diubah menjadi matriks 10×10 yang kemudian dijadikan menjadi 100×1 . Ukuran matriks 100×1 ini digunakan pada proses pengenalan. Matriks 100×1 ini yang akan dijadikan sebagai *input-an*, dengan *input* sebanyak 100 buah. Proses ini merupakan bentuk data matriks citra yang terdiri dari nilai 0 dan 1, yang akan mempresentasikan data pada matriks citra 100×100 yang nantinya akan menjadi 100×1 . Proses ini membantu sistem menjadi lebih cepat menyimpan data ke dalam storage. Proses pengubahan ukuran matriks ini tidak menyebabkan hilangnya informasi.

- Konversi Nilai

Digunakan pada sistem bertujuan untuk menganalisa karakteristik tanda tangan. Konversi dilakukan dengan mempresentasikan setiap sepuluh satuan data secara horizontal dari baris dan kolom terkecil sampai dengan baris dan kolom terbesar dengan nilai konversi yang telah dihitung (biner menjadi desimal). Setelah dikonversi, didapatkan matriks 100×10 . Sebagai contoh, diambil area matriks baris terakhir kolom pertama.



Gambar 3.5
 Pengelompokan Matriks 100x100 dengan Setiap Sepuluh Satuan Data secara Horizontal dan Vertikal

	1	2	3	4	5	6	7	8	9	10	11	12	13
84	1	0	0	0	0	0	0	0	0	0	0	0	1
85	1	0	0	0	0	0	0	0	0	0	0	0	1
86	1	0	0	0	0	0	0	0	0	0	0	1	1
87	1	0	0	0	0	0	0	0	0	0	0	1	1
88	1	0	0	0	0	0	0	0	0	0	1	1	1
89	1	1	0	0	0	0	0	0	0	1	1	1	0
90	1	1	0	0	0	0	0	0	0	1	1	1	0
91	0	1	0	0	0	0	0	0	0	1	1	0	0
92	0	1	0	0	0	0	0	1	1	0	0	0	0
93	0	1	0	0	0	0	0	1	1	0	0	0	0
94	0	1	1	0	0	1	1	1	1	0	0	0	0
95	0	1	1	0	0	1	1	1	1	0	0	0	0
96	0	0	1	1	1	1	1	0	0	0	0	0	0
97	0	0	0	1	1	0	0	0	0	0	0	0	0
98	0	0	0	1	1	0	0	0	0	0	0	0	0
99	0	0	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 3.6
 Matriks Kolom Pertama Baris Terakhir

Cara konversi nilai dari contoh di atas:

- Untuk baris ke-1 = $0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1$

$$= (2^0 \times 0) + (2^1 \times 1) + (2^2 \times 0) + (2^3 \times 0) + (2^4 \times 0) + (2^5 \times 0) +$$

$$(2^6 \times 0) + (2^7 \times 0) + (2^8 \times 0) + (2^9 \times 1)$$

$$= 0 + 2 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 512$$

$$= 514$$

$$= 514 \times \frac{1}{10.000}$$

$$= 0.0514$$

- Untuk baris ke-2 = 0 1 0 0 0 0 0 1 1 0

$$= (2^0 \times 0) + (2^1 \times 1) + (2^2 \times 0) + (2^3 \times 0) + (2^4 \times 0) + (2^5 \times 0)$$

$$+ (2^6 \times 1) + (2^7 \times 1) + (2^8 \times 0) + (2^9 \times 0)$$

$$= 0 + 2 + 0 + 0 + 0 + 0 + 0 + 0 + 128 + 256 + 0$$

$$= 386$$

$$= 386 \times \frac{1}{10.000}$$

$$= 0.0386$$

- Untuk baris ke-3 = 0 1 0 0 0 0 0 1 1 0

$$= (2^0 \times 0) + (2^1 \times 1) + (2^2 \times 0) + (2^3 \times 0) + (2^4 \times 0) + (2^5 \times 0)$$

$$+ (2^6 \times 1) + (2^7 \times 1) + (2^8 \times 0) + (2^9 \times 0)$$

$$= 0 + 2 + 0 + 0 + 0 + 0 + 0 + 0 + 128 + 256 + 0$$

$$= 386$$

$$= 386 \times \frac{1}{10.000}$$

$$= 0.0386$$

- Untuk baris ke-4 = 0 1 1 0 0 1 1 1 1 0

$$= (2^0 \times 0) + (2^1 \times 1) + (2^2 \times 1) + (2^3 \times 0) + (2^4 \times 0) + (2^5 \times 1)$$

$$+ (2^6 \times 1) + (2^7 \times 1) + (2^8 \times 1) + (2^9 \times 0)$$

$$= 0 + 2 + 4 + 0 + 0 + 32 + 64 + 128 + 256 + 0$$

$$= 486$$

$$= 406 \times \frac{1}{10.000}$$

$$= 0.0486$$

- Untuk baris ke-5 = 0 1 1 0 0 1 1 1 1 0

$$= (2^0 \times 0) + (2^1 \times 1) + (2^2 \times 1) + (2^3 \times 0) + (2^4 \times 0) + (2^5 \times 1)$$

$$+ (2^6 \times 1) + (2^7 \times 1) + (2^8 \times 1) + (2^9 \times 0)$$

$$= 0 + 2 + 4 + 0 + 0 + 32 + 64 + 128 + 256 + 0$$

$$= 486$$

$$= 486 \times \frac{1}{10.000}$$

$$= 0.0486$$

- Untuk baris ke-6 = 0 0 1 1 1 1 1 0 0 0

$$= (2^0 \times 0) + (2^1 \times 0) + (2^2 \times 1) + (2^3 \times 1) + (2^4 \times 1) + (2^5 \times 1)$$

$$+ (2^6 \times 1) + (2^7 \times 0) + (2^8 \times 0) + (2^9 \times 0)$$

$$= 0 + 0 + 4 + 8 + 16 + 32 + 64 + 0 + 0 + 0$$

$$= 124$$

$$= 124 \times \frac{1}{10.000}$$

$$= 0.0124$$

- Untuk baris ke-7 = 0 0 0 1 1 0 0 0 0 0

$$= (2^0 \times 0) + (2^1 \times 0) + (2^2 \times 0) + (2^3 \times 1) + (2^4 \times 1) + (2^5 \times 0)$$

$$+ (2^6 \times 0) + (2^7 \times 0) + (2^8 \times 0) + (2^9 \times 0)$$

$$= 0 + 0 + 0 + 8 + 16 + 0 + 0 + 0 + 0 + 0$$

$$= 24$$

$$= 24 \times \frac{1}{10.000}$$

$$= 0.0024$$

- Untuk baris ke-8 = 0 0 0 1 1 0 0 0 0

$$= (2^0 \times 0) + (2^1 \times 0) + (2^2 \times 0) + (2^3 \times 1) + (2^4 \times 1) + (2^5 \times 0)$$

$$+ (2^6 \times 0) + (2^7 \times 0) + (2^8 \times 0) + (2^9 \times 0)$$

$$= 0 + 0 + 0 + 8 + 16 + 0 + 0 + 0 + 0 + 0$$

$$= 24$$

$$= 24 \times \frac{1}{10.000}$$

$$= 0.0024$$

- Untuk baris ke-9 = 0 0 0 0 0 0 0 0 0

$$= (2^0 \times 0) + (2^1 \times 0) + (2^2 \times 0) + (2^3 \times 0) + (2^4 \times 0) + (2^5 \times 0)$$

$$+ (2^6 \times 0) + (2^7 \times 0) + (2^8 \times 0) + (2^9 \times 0)$$

$$= 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0$$

$$= 0$$

- Untuk baris ke-9 = 0 0 0 0 0 0 0 0 0

$$= (2^0 \times 0) + (2^1 \times 0) + (2^2 \times 0) + (2^3 \times 0) + (2^4 \times 0) + (2^5 \times 0)$$

$$+ (2^6 \times 0) + (2^7 \times 0) + (2^8 \times 0) + (2^9 \times 0)$$

$$= 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0$$

$$= 0$$

1. Matriks 10x100 diubah menjadi matriks 10x10

Matriks yang telah berukuran 100x10 dikecilkan kembali dengan cara menjumlahkan tiap nilai vertical dari tiap-tiap kolom, sehingga menjadi matriks 10x10.

78	0.0002	0.0224	0.0000	0.0000	0.0000	0.0902	0.0896	0.0495	0.0064
79	0.0002	0.0096	0.0000	0.0000	0.0000	0.1022	0.1016	0.0481	0.0096
80	0.0002	0.0096	0.0000	0.0000	0.0000	0.1022	0.1016	0.0481	0.0096
81	0.0002	0.0048	0.0000	0.0000	0.0000	0.1016	0.0255	0.0224	0.0032
82	0.0002	0.0048	0.0000	0.0000	0.0000	0.1016	0.0255	0.0224	0.0032
83	0.0002	0.0024	0.0000	0.0000	0.0000	0.1016	0.0015	0.0112	0.0048
84	0.0001	0.0028	0.0000	0.0000	0.0512	0.0511	0.0000	0.0112	0.0048
85	0.0001	0.0028	0.0000	0.0000	0.0512	0.0511	0.0000	0.0112	0.0048
86	0.0001	0.0014	0.0000	0.0000	0.0992	0.0015	0.0000	0.0016	0.0056
87	0.0001	0.0014	0.0000	0.0000	0.0992	0.0015	0.0000	0.0016	0.0056
88	0.0001	0.0007	0.0000	0.0512	0.1023	0.0001	0.0000	0.0016	0.0024
89	0.0515	0.0003	0.0000	0.0992	0.0031	0.0000	0.0000	0.0000	0.0028
90	0.0515	0.0003	0.0000	0.0992	0.0031	0.0000	0.0000	0.0000	0.0028
91	0.0514	0.0001	0.0000	0.1022	0.0003	0.0000	0.0000	0.0000	0.0012
92	0.0386	0.0000	0.0896	0.0255	0.0000	0.0000	0.0000	0.0000	0.0014
93	0.0386	0.0000	0.0896	0.0255	0.0000	0.0000	0.0000	0.0000	0.0014
94	0.0486	0.0000	0.1008	0.0007	0.0000	0.0000	0.0000	0.0000	0.0006
95	0.0486	0.0000	0.1008	0.0007	0.0000	0.0000	0.0000	0.0000	0.0006
96	0.0124	0.0000	0.1022	0.0000	0.0000	0.0000	0.0000	0.0000	0.0007
97	0.0024	0.0512	0.0063	0.0000	0.0000	0.0000	0.0000	0.0000	0.0003
98	0.0024	0.0512	0.0063	0.0000	0.0000	0.0000	0.0000	0.0000	0.0003
99	0.0000	0.0960	0.0003	0.0000	0.0000	0.0000	0.0000	0.0512	0.0001
100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Gambar 3.8
Konversi ke Matriks 10x10

Berikut perhitungannya:

$$\begin{aligned}
 &= 0.0514 + 0.0386 + 0.0386 + 0.0486 + 0.0486 + 0.0124 + 0.0024 \\
 &\quad + 0.0024 + 0.0000 + 0.0000 \\
 &= 0.2430
 \end{aligned}$$

	1	2	3	4	5	6	7	8	9	10
1	0.0000	0.0000	0.0000	0.6320	0.0002	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.0512	0.6253	0.7547	0.0106	0.0000	0.0000	0.0000	0.1952	0.0028
3	0.0000	0.5216	0.6536	0.5406	0.0005	0.0000	0.0000	0.0000	0.2868	0.5401
4	0.0000	0.2048	1.0202	0.0234	0.0784	0.4589	0.0012	0.0000	0.8088	0.5305
5	0.0000	0.0000	0.6981	0.8626	0.5775	0.6350	0.1717	0.3932	0.1040	0.1116
6	0.0000	0.6112	0.0561	0.0796	0.0152	0.0068	0.0442	0.3176	0.3377	0.0120
7	0.0000	0.1356	0.0051	0.4960	0.1116	0.0000	0.0057	0.0512	0.5542	0.0024
8	0.0020	0.3704	0.0002	0.0768	0.1792	0.6278	0.2941	0.7323	0.3806	0.0006
9	0.1041	0.0217	0.0000	0.2496	0.4093	0.4101	0.0525	0.0832	0.0400	0.0000
10	0.2430	0.1985	0.4959	0.1546	0.0003	0.0000	0.0000	0.0512	0.0066	0.0000

Gambar 3.9
Matriks 10x10

2. Matriks 10x10 diubah menjadi matriks 100x1

Matriks yang telah berukuran 10x10 dikecilkan lagi menjadi 100x1. Konversi matriks ini dilakukan dengan memindahkan tiap-tiap kolom menjadi satu kolom saja. Citra yang telah diolah menjadi matriks 100x1 ini digunakan JST sebagai *input layer*.

	1	2	3	4	5	6	7	8	9	10
1	0.0000	0.0000	0.0000	0.6320	0.0002	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.0512	0.6253	0.7547	0.0106	0.0000	0.0000	0.0000	0.1952	0.0028
3	0.0000	0.5216	0.6536	0.5406	0.0005	0.0000	0.0000	0.0000	0.2868	0.5401
4	0.0000	0.2048	1.0202	0.0234	0.0794	0.4589	0.0012	0.0000	0.8088	0.5305
5	0.0000	0.0000	0.6981	0.8626	0.5775	0.6350	0.1717	0.3932	0.1040	0.1116
6	0.0000	0.6112	0.0561	0.0796	0.0152	0.0068	0.0442	0.3176	0.3377	0.0120
7	0.0000	0.1356	0.0051	0.4960	0.1116	0.0000	0.0057	0.0512	0.5542	0.0024
8	0.0020	0.3704	0.0002	0.0768	0.1792	0.6278	0.2941	0.7323	0.3806	0.0006
9	0.1041	0.0217	0.0000	0.2496	0.4093	0.4101	0.0525	0.0832	0.0400	0.0000
10	0.2430	0.1995	0.4959	0.1546	0.0003	0.0000	0.0000	0.0512	0.0066	0.0000

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.0000	0.0000	0.0000	0.632	0.0002	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0512	0.6253

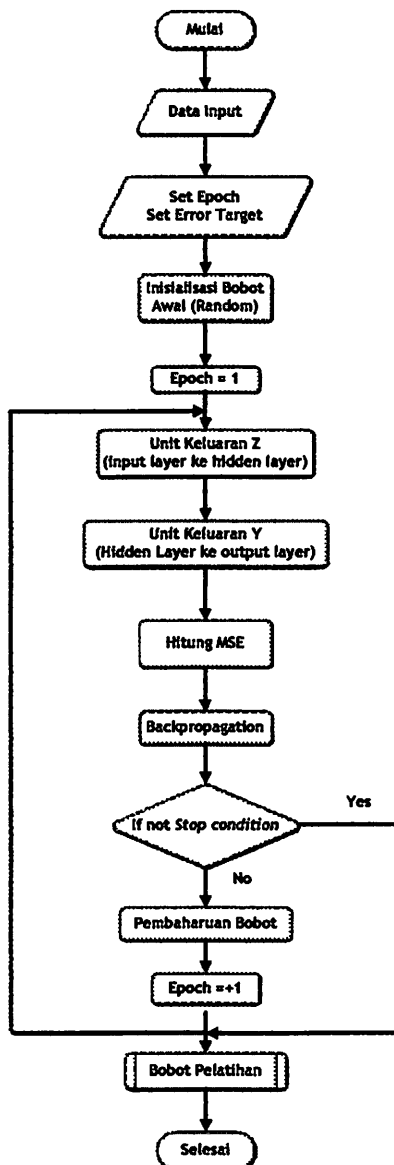
..dst sampai dengan 100x1.

Gambar 3.10
Matriks 100x1

3.2.1.1 Proses Pelatihan

Citra yang telah dipersiapkan dari proses pengolahan citra menjadi matriks 100x1 memasuki tahap pelatihan untuk dipelajari dengan sistem JST seperti yang telah diilustrasikan. Tahap pelatihan dimulai dengan proses *feedforward* yang dipakai untuk mencari nilai *hidden layer* dan nilai *output layer*. Jika hasil *output layer* tidak sesuai dengan target, maka dilakukan proses *backward/backpropagation* untuk mencari nilai *error* yang ada di *output layer* dan *hidden layer* yang nantinya untuk memperbaiki nilai bobot yang ada di *hidden*

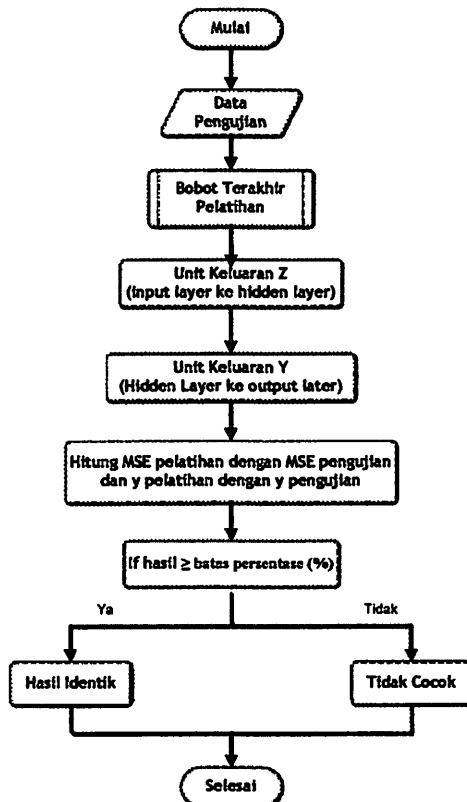
layer dan *input* layer, setelah itu dilakukan kembali *feedforward* jika nilai keluaran tidak sesuai dengan target.



Gambar 3.11
Diagram Alir Proses Pelatihan

3.2.1.2 Proses Pengenalan

Setelah proses pelatihan selesai dilakukan, maka proses pengenalan dapat dilakukan. Tahap pengenalan memiliki cara kerja yang hampir sama dengan tahap pelatihan, di mana citra hasil *scan* disiapkan terlebih dahulu melalui proses pengolahan citra. Perbedaan tahapan kerja pelatihan dengan pengenalan adalah hanya pada proses setelah proses normalisasi. Tahap pengenalan hanya menggunakan proses *feedforwad* yang dipakai untuk mencari nilai *hidden layer* dan nilai *output layer*. Nilai pada semua node *output layer* dibandingkan untuk mendapatkan node yang memiliki nilai terbesar. Node yang memiliki nilai terbesar ini diverifikasi, apakah sesuai dengan identitas pengguna yang tanda tangannya ingin dikenali.



Gambar 3.12
Diagram Alir Proses Pengenalan/Pengujian

Berikut rumus pada proses pengenalan/pengujian:

- Untuk perhitungan MSE

$$MSE = 0.5 * (t_{pi} - y_{pi})^2 + (t_u - y_u)^2$$

Jika hasil MSE \leq (kurang dari sama dengan) target *error* (0.01),

maka:

Total MSE yang dikenali = +1

Di mana: t_{pi} = target pelatihan ke-i

y_{pi} = y pelatihan ke-i

t_u = target pengujian

y_u = y pengujian

- Untuk perhitungan y (*output*)

Jika y pelatihan lebih besar dari y pengujian, maka :

$$\text{Hasil} = y_{\text{pelatihan}} - Y_{\text{pengujian}}$$

Namun, jika y pelatihan kurang dari y pengujian, maka:

$$\text{Hasil} = y_{\text{pengujian}} - Y_{\text{pelatihan}}$$

Jika hasil perhitungan y tersebut \leq (kurang dari sama dengan) target $\text{error}_1(0.02)$, maka:

$$\text{Total } y \text{ yang dikenali} = + 1$$

- Untuk perhitungan persentase:

$$\text{persentase} = \frac{\text{total MSE yang dikenali}}{\text{total pelatihan}} \times 100\%$$

Jika hasil persentase tersebut \geq (lebih besar sama dengan) batas persentase (90%), maka:

$$\frac{\text{total } y \text{ (output) yang dikenali}}{\text{total pelatihan}} \times 100\%$$

3.2.2 Tempat Penyimpanan Data (*Data Storage*)

Tempat penyimpanan data sistem terdiri dari 6 (enam) macam tabel. Nilai bobot-bobot yang telah disimpan di sini akan dipakai dalam tahap pengenalan untuk membandingkan sebuah tanda tangan yang telah diproses dalam tahap pelatihan. Media penyimpanan yang digunakan adalah perangkat lunak Microsoft Access XP.

a. Tabel_User

Nama Field	Tipe	Size	Keterangan
No_User	Text	6	Nomer urut user
Nama	Text	30	Nama user terdaftar

Tabel 3.1
Tabel User

b. T_Pelatihan

Nama Field	Tipe	Size	Keterangan
No_Pelatihan	Text	9	Nomer pelatihan
No_User	Text	6	Nomer user
Nilai_Output	Number	-	Nilai keluaran dari perhitungan
Nilai_Target	Number	-	Nilai target dari perhitungan

Tabel 3.2
Tabel Pelatihan

c. T_BobotBias2Hidden (V_0)

Nama Field	Tipe	Size	Keterangan
No_Bobot_V ₀	Text	9	Nomer bobot awal bias ke hidden
No_Pelatihan	Text	6	Nomer pelatihan
Nilai_Bobot_V ₀	Number	-	Nilai bobot awal bias ke hidden

Tabel 3.3
Tabel Bobot Awal Bias *To Hidden*

d. T_BobotInput2Hidden (V)

Nama Field	Tipe	Size	Keterangan
No_Bobot_V	Text	11	Nomer bobot awal input ke hidden
No_Pelatihan	Text	6	Nomer pelatihan
Nilai_Bobot_V	Number	-	Nilai bobot awal input ke hidden

Tabel 3.4

Tabel Bobot Awal *Input To Hidden*

e. T_BobotAwalBias2Output (W_0)

Nama Field	Tipe	Size	Keterangan
No_Bobot_ W_0	Text	8	Nomer bobot awal bias ke output
No_Pelatihan	Text	6	Nomer pelatihan
Nilai_Bobot_ W_0	Number	-	Nilai bobot awal bias ke output

Tabel 3.5

Tabel Bobot Awal Bias *To Output*

f. T_BobotHidden2Output (W)

Nama Field	Tipe	Size	Keterangan
No_Bobot_W	Text	10	Nomer bobot awal hidden ke output
No_Pelatihan	Text	6	Nomer pelatihan
Nilai_Bobot_W	Number	-	Nilai bobot awal hidden ke output

Tabel 3.6

Tabel Bobot Awal *Hidden To Output*

3.2.3 Perancangan Antarmuka

Sesuai dengan spesifikasi sistem diatas, berikut perancangan form-form yang digunakan dalam sistem ini:

a. Desain Form Menu Utama

Pelatihan Pengujian Exit

Gambar 3.13
Desain Form Menu Utama

b. Desain Form User

No User	:						
Nama	:						
Total Pelatihan	:						
<table border="1"><tr><td>Pelatihan</td></tr></table>			Pelatihan				
Pelatihan							
<table border="1"><tr><td>Update</td></tr></table>	Update	<table border="1"><tr><td>Delete</td></tr></table>	Delete	<table border="1"><tr><td>Cancel</td></tr></table>	Cancel	<table border="1"><tr><td>Exit</td></tr></table>	Exit
Update							
Delete							
Cancel							
Exit							
No User	Nama	Total Pelatihan					

Gambar 3.14
Desain Form User

c. Desain Form Pelatihan

Open File		Save Pelatihan									
Mulai Pelatihan		Hasil Biner									
<input type="text"/>		<table border="1"><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>									
MSE	<input type="text"/>										
Cropping		Matriks 10x100 Matriks 10x10									
<input type="text"/>		<table border="1"><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table> <table border="1"><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>									
Threshold	Segmentasi	Scalling									
<input type="text"/>	<input type="text"/>	<input type="text"/>									
		Matriks 100x1									
		<table border="1"><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>									

Gambar 3.15
Desain Form Pelatihan

d. Desain Form Pengujian

No User :		Data Biner				
Nama :		<table border="1"><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>				
Input Jaringan :						
<input type="text"/>						
Load Signature						
<input type="text"/>						
Pengujian Tanda Tangan						
Comparison Process						
% Kemiripan :						
Keterangan :						

Gambar 3.16
Desain Form Pengujian

3.3 Contoh Perhitungan Matematis

Misalkan, terdapat suatu hasil dari semua proses yang sudah dijelaskan di atas, didapatkan suatu matriks 2x1 pada suatu citra tanda tangan sebagai berikut:

I. *Input* 1

	1	2
1	0.2688	0.5561

Dari *input*-an di atas, maka didapatkan:

$$t \text{ (target)} = 0.2688 + 0.5561 = 0.8249$$

$$x = (0.2688 ; 0.5561)$$

Dengan ketentuan :

Jumlah *input layer* : 2

Jumlah *hidden layer* : 2

Jumlah *output layer* : 1

Learning rate (α) : 0.5

Maksimum Epoch : 2

Target *Error* : 0.01

Target *Error*₁ : 0.02

Bobot Awal (ditentukan secara acak) :

a. Bobot Awal input ke hidden (v):

$$v_{11} = 0.1632 \quad v_{12} = 0.2886$$

$$v_{21} = 0.0133 \quad v_{22} = 0.3711$$

b. Bobot Awal bias ke hidden (v_0):

$$v_{01} = 0.3496$$

$$v_{02} = 0.1628$$

c. Bobot Awal hidden ke output (w):

$$w_1 = 0.4585$$

$$w_2 = 0.0550$$

d. Bobot Awal bias ke output (w):

$$w_0 = 0.4505$$

- Pembelajaran :

1. Iterasi I pada *input* 1

Berikut akan dijabarkan perhitungan yang terjadi pada Epoch I (iterasi I) dengan menggunakan *input* $x = (0.2688 ; 0.5561)$:

a. Operasi pada Hidden Layer:

$$\begin{aligned} z_{in1} &= v_{01} + v_{11} * x_1 + v_{21} * x_2 \\ &= 0.3496 + (0.1632 * 0.2688) + (0.0133 * 0.5561) \\ &= 0.3496 + 0.4320 + 0.0073 \\ &= 0.7889 \end{aligned}$$

$$\begin{aligned} z_{in2} &= v_{02} + v_{12} * x_1 + v_{22} * x_2 \\ &= 0.1628 + (0.2886 * 0.2688) + (0.3711 * 0.5561) \\ &= 0.1628 + 0.0775 + 0.2063 \\ &= 0.4466 \end{aligned}$$

Menghitung sinyal output:

$$z_1 = \frac{1}{1 + e^{-z_{in1}}}$$

$$= \frac{1 - e^{-0.7889}}{1 + e^{-0.7889}}$$

$$= \frac{0.5457}{1.4543}$$

$$= 0.3752$$

$$z_2 = \frac{1 - e^{-z_{in_2}}}{1 + e^{-z_{in_2}}}$$

$$= \frac{1 - e^{-0.4406}}{1 + e^{-0.4406}}$$

$$= \frac{0.3603}{1.6397}$$

$$= 0.2197$$

b. Operasi pada Output Layer:

$$y_{in} = w_0 + w_1 * z_1 + w_2 * z_2$$

$$= 0.4505 + (0.4585 * 0.3752) + (0.0550 * 0.2191)$$

$$= 0.4505 + 0.8337 + 0.2741$$

$$= 1.5583$$

Menghitung sinyal output:

$$y = \frac{1 - e^{-y_{in}}}{1 + e^{-y_{in}}}$$

$$= \frac{1 - e^{-1.5583}}{1 + e^{-1.5583}}$$

$$= \frac{0.7896}{1.2104}$$

$$= 0.6523$$

$$\delta = \frac{(t - y)}{2} * \left[1 + \left(\frac{1 - e^{-y_{in}}}{1 + e^{-y_{in}}} \right) \right] * \left[1 - \left(\frac{1 - e^{-y_{in}}}{1 + e^{-y_{in}}} \right) \right]$$

$$- \frac{(0.8249 - 0.6523)}{2} \cdot \left[1 + \left(\frac{1 - e^{-1.5583}}{1 + e^{-1.5583}} \right) \right] \cdot \left[1 - \left(\frac{1 - e^{-1.5583}}{1 + e^{-1.5583}} \right) \right]$$

$$= \frac{0.1726}{2} * [1 + (0.6523)] * [1 - (0.6523)]$$

$$= 0.0863 * 1.6523 * 0.3477$$

$$= 0.0495$$

$$\Delta w_1 = \alpha * \delta * z_1$$

$$= 0.5 * 0.0495 * 0.3752$$

$$= 0.0092$$

$$\Delta w_2 = \alpha * \delta * z_2$$

$$= 0.5 * 0.0495 * 0.2197$$

$$= 0.0054$$

$$\Delta w_0 = \alpha * \delta$$

$$= 0.5 * 0.0495$$

$$= 0.0247$$

$$\delta in_1 = \delta * w_1$$

$$= 0.0495 * 0.4585$$

$$= 0.0226$$

$$\delta in_2 = \delta * w_2$$

$$= 0.0495 * 0.0550$$

$$= 0.0027$$

$$\delta_1 = - \frac{\delta in_1}{2} \cdot \left[1 + \left(\frac{1 - e^{-0.7839}}{1 + e^{-0.7839}} \right) \right] \cdot \left[1 - \left(\frac{1 - e^{-0.7839}}{1 + e^{-0.7839}} \right) \right]$$

$$= \frac{0.0226}{2} * \left[1 + \left(\frac{1 - e^{-0.7839}}{1 + e^{-0.7839}} \right) \right] * \left[1 - \left(\frac{1 - e^{-0.7839}}{1 + e^{-0.7839}} \right) \right]$$

$$= 0.0113 * (1 + 0.3752) * (1 - 0.3752)$$

$$= 0.0113 * 1.3752 * 0.6248$$

$$= 0.0097$$

$$\delta_2 = -\frac{\delta i n_2}{2} * \left[1 + \left(\frac{1 - e^{-\sigma i n_2}}{1 + e^{\sigma i n_2}} \right) \right] * \left[1 - \left(\frac{1 - e^{-\sigma i n_2}}{1 + e^{\sigma i n_2}} \right) \right]$$

$$= -\frac{0.0027}{2} * [1 + (0.2179)] * [1 - (0.2179)]$$

$$= 0.0013 * 1.2179 * 0.7821$$

$$= 0.0012$$

$$\Delta v_{11} = \alpha * \delta_1 * x_1$$

$$= 0.5 * 0.0097 * 0.2688$$

$$= 0.0013$$

$$\Delta v_{12} = \alpha * \delta_2 * x_1$$

$$= 0.5 * 0.0012 * 0.2688$$

$$= 0.0016$$

$$\Delta v_{21} = \alpha * \delta_1 * x_2$$

$$= 0.5 * 0.0097 * 0.5561$$

$$= 0.0026$$

$$\Delta v_{22} = \alpha * \delta_2 * x_2$$

$$= 0.5 * 0.0012 * 0.5561$$

$$= 0.0033$$

$$\Delta v_{01} = \alpha * \delta_1$$

$$= 0.5 * 0.0097$$

$$= 0.0048$$

$$\begin{aligned}
\Delta v_0 &= \alpha * \delta_2 \\
&= 0.5 * 0.0012 \\
&= 0.0006 \\
w_1 &= w_1 + \Delta w_1 \\
&= 0.4585 + 0.0092 \\
&= 0.4677 \\
w_2 &= w_2 + \Delta w_2 \\
&= 0.0550 + 0.0054 \\
&= 0.0604 \\
w_0 &= w_0 + \Delta w_0 \\
&= 0.4505 + 0.0247 \\
&= 0.4752 \\
v_{11} &= v_{11} + \Delta v_{11} \\
&= 0.1623 + 0.0013 \\
&= 0.1636 \\
v_{12} &= v_{12} + \Delta v_{12} \\
&= 0.2886 + 0.0016 \\
&= 0.2902 \\
v_{21} &= v_{21} + \Delta v_{21} \\
&= 0.0133 + 0.0026 \\
&= 0.0159 \\
v_{22} &= v_{22} + \Delta v_{22} \\
&= 0.3711 + 0.0033
\end{aligned}$$

$$= 0.3744$$

$$\begin{aligned}v_{0_1} &= v_{0_1} + \Delta v_{0_1} \\ &= 0.3496 + 0.0048 \\ &= 0.3544\end{aligned}$$

$$\begin{aligned}v_{0_2} &= v_{0_2} + \Delta v_{0_2} \\ &= 0.1628 + 0.0006 \\ &= 0.1634\end{aligned}$$

$$\begin{aligned}\text{Jumlah Kuadrat Error (MSE)} &= 0.5 * (t - y)^2 \\ &= 0.5 * (0.8249 - 0.6523)^2 \\ &= 0.0148\end{aligned}$$

2. Iterasi II pada input 1

Berikut akan dijabarkan perhitungan yang terjadi pada Epoch I (iterasi I)

dengan menggunakan *input* $x = (0.2688 ; 0.5561)$:

$$\begin{aligned}z_{in_1} &= v_{0_1} + v_{11} * x_1 + v_{21} * x_2 \\ &= 0.3544 + (0.1636 * 0.2688) + (0.0159 * 0.5561) \\ &= 0.3544 + 0.0439 + 0.0089 \\ &= 0.4072\end{aligned}$$

$$\begin{aligned}z_{in_2} &= v_{0_2} + v_{12} * x_1 + v_{22} * x_2 \\ &= 0.1634 + (0.2902 * 0.2688) + (0.3744 * 0.5561) \\ &= 0.1634 + 0.0780 + 0.2082 \\ &= 0.4496\end{aligned}$$

Menghitung sinyal output:

$$\begin{aligned}
 z_1 &= \frac{1 - e^{-z_{in1}}}{1 + e^{-z_{in1}}} \\
 &= \frac{1 - e^{-0.9703}}{1 + e^{-0.9703}} \\
 &= \frac{0.7896}{1.3789} \\
 &= 0.5726
 \end{aligned}$$

$$\begin{aligned}
 z_2 &= \frac{1 - e^{-z_{in2}}}{1 + e^{-z_{in2}}} \\
 &= \frac{1 - e^{-0.4496}}{1 + e^{-0.4496}} \\
 &= \frac{0.3922}{1.6378} \\
 &= 0.2394
 \end{aligned}$$

c. Operasi pada Output Layer:

$$\begin{aligned}
 y_{in} &= w_0 + w_1 * z_1 + w_2 * z_2 \\
 &= 0.4752 + (0.4677 * 0.5726) + (0.0604 * 0.2394) \\
 &= 0.4752 + 0.2678 + 0.0144 \\
 &= 0.7574
 \end{aligned}$$

Menghitung sinyal output:

$$\begin{aligned}
 y &= \frac{1 - e^{-y_{in}}}{1 + e^{-y_{in}}} \\
 &= \frac{0.5312}{1.4688} \\
 &= 0.3616
 \end{aligned}$$

$$\delta = \frac{(t - y)}{2} * \left[1 + \left(\frac{1 - e^{-y_{in}}}{1 + e^{-y_{in}}} \right) \right] * \left[1 - \left(\frac{1 - e^{-y_{in}}}{1 + e^{-y_{in}}} \right) \right]$$

$$= \frac{0.3002}{2} * [1 + (1.1331)] * [1 - (1.1331)]$$

$$= 0.1541 * 2.1331 * (-0.1331)$$

$$= -0.0437$$

$$\Delta w_1 = \alpha * \delta * z_1$$

$$= 0.5 * (-0.0437) * 0.5726$$

$$= -0.0125$$

$$\Delta w_2 = \alpha * \delta * z_2$$

$$= 0.5 * (-0.0437) * 0.2394$$

$$= -0.0052$$

$$\Delta w_0 = \alpha * \delta$$

$$= 0.5 * (-0.0437)$$

$$= -0.0218$$

$$\delta in_1 = \delta * w_1$$

$$= (-0.0437) * 0.4677$$

$$= -0.0204$$

$$\delta in_2 = \delta * w_2$$

$$= (-0.0437) * 0.0604$$

$$= -0.0026$$

$$\delta_1 = \frac{\delta in_1}{2} * \left[1 + \left(\frac{1 - e^{-z_1 in_1}}{1 + e^{-z_1 in_1}} \right) \right] * \left[1 + \left(\frac{1 - e^{-z_1 in_2}}{1 + e^{-z_1 in_2}} \right) \right]$$

$$= \frac{-0.0204}{2} * [1 + (0.5726)] * [1 + (0.5726)]$$

$$= -0.0102 * 1.5726 * 0.4274$$

$$= -0.0068$$

$$\delta_2 = \frac{\delta \ln_2}{\lambda} * \left[1 + \left(\frac{1 - e^{-\lambda_1 x_1}}{1 + e^{\lambda_1 x_1}} \right) \right] * \left[1 - \left(\frac{1 - e^{-\lambda_2 x_2}}{1 + e^{\lambda_2 x_2}} \right) \right]$$

$$= \frac{-0.0026}{2} * [1 + (0.2394)] * [1 - (0.2394)]$$

$$= -0.0013 * 1.2394 * 0.7606$$

$$= -0.0012$$

$$\Delta v_{11} = \alpha * \delta_1 * x_1$$

$$= 0.5 * -0.0068 * 0.2688$$

$$= -0.0091$$

$$\Delta v_{12} = \alpha * \delta_2 * x_1$$

$$= 0.5 * -0.0012 * 0.2688$$

$$= -0.0016$$

$$\Delta v_{21} = \alpha * \delta_1 * x_2$$

$$= 0.5 * -0.0068 * 0.5561$$

$$= -0.0018$$

$$\Delta v_{22} = \alpha * \delta_2 * x_2$$

$$= 0.5 * -0.0012 * 0.5561$$

$$= -0.0003$$

$$\Delta v_{01} = \alpha * \delta_1$$

$$= 0.5 * -0.0068$$

$$= -0.0034$$

$$\Delta v_{02} = \alpha * \delta_2$$

$$= 0.5 * -0.0012$$

$$= -0.0006$$

$$\begin{aligned}w_1 &= w_1 + \Delta w_1 \\ &= 0.4677 + (-0.0125) \\ &= 0.4552\end{aligned}$$

$$\begin{aligned}w_2 &= w_2 + \Delta w_2 \\ &= 0.0604 + (-0.0052) \\ &= 0.0552\end{aligned}$$

$$\begin{aligned}w_0 &= w_0 + \Delta w_0 \\ &= 0.4752 + (-0.0218) \\ &= 0.4534\end{aligned}$$

$$\begin{aligned}v_{11} &= v_{11} + \Delta v_{11} \\ &= 0.1639 + (-0.0091) \\ &= 0.1548\end{aligned}$$

$$\begin{aligned}v_{12} &= v_{12} + \Delta v_{12} \\ &= 0.2902 + (-0.0016) \\ &= 0.2886\end{aligned}$$

$$\begin{aligned}v_{21} &= v_{21} + \Delta v_{21} \\ &= 0.0159 + (-0.0018) \\ &= 0.0141\end{aligned}$$

$$\begin{aligned}v_{22} &= v_{22} + \Delta v_{22} \\ &= 0.3744 + (-0.0003) \\ &= 0.3741\end{aligned}$$

$$\begin{aligned}v_{0_1} &= v_{0_1} + \Delta v_{0_1} \\ &= 0.3544 + (-0.0034)\end{aligned}$$

$$\begin{aligned}
&= 0.3510 \\
v_{0_2} &= v_{0_2} + \Delta v_{0_2} \\
&= 0.1634 + (-0.0006) \\
&= 0.1628
\end{aligned}$$

$$\begin{aligned}
\text{Jumlah Kuadrat Error (MSE)} &= 0.5 * (t - y)^2 \\
&= 0.5 * (0.8249 - 1.1331)^2 \\
&= 0.0474
\end{aligned}$$

II. Input 2

	1	2
1	0,0000	0.719

Dari *input*-an di atas, maka didapatkan:

$$t \text{ (target)} = 0.0000 + 0.7190 = 0.7190$$

$$x = (0.0000 ; 0.7190)$$

1. Iterasi I pada *input* 2

Berikut akan dijabarkan perhitungan yang terjadi pada Epoch I (iterasi I)

dengan menggunakan *input* $x = (0.0000 ; 0.7190)$:

a. Operasi pada Hidden Layer:

$$\begin{aligned}
z_{in_1} &= v_{0_1} + v_{11} * x_1 + v_{21} * x_2 \\
&= 0.3510 + (0.1548 * 0.0000) + (0.0141 * 0.7190) \\
&= 0.3510 + 0 + 0.0101 \\
&= 0.3611
\end{aligned}$$

$$z_{in_2} = v_{0_2} + v_{12} * x_1 + v_{22} * x_2$$

$$\begin{aligned}
&= 0.1628 + (0.2886 * 0.0000) + (0.3741 * 0.7190) \\
&= 0.1628 + 0 + 0.2689 \\
&= 0.4317
\end{aligned}$$

Menghitung sinyal output:

$$\begin{aligned}
z_1 &= \frac{1 - e^{-z_{in1}}}{1 + e^{-z_{in1}}} \\
&= \frac{0.3031}{1.6969} \\
&= 0.1786
\end{aligned}$$

$$\begin{aligned}
z_2 &= \frac{1 - e^{-z_{in2}}}{1 + e^{-z_{in2}}} \\
&= \frac{0.3506}{1.6494} \\
&= 0.2125
\end{aligned}$$

b. Operasi pada Output Layer:

$$\begin{aligned}
y_{in} &= w_0 + w_1 * z_1 + w_2 * z_2 \\
&= 0.4534 + (0.4552 * 0.1786) + (0.0552 * 0.2125) \\
&= 0.4534 + 0.0812 + 0.0117 \\
&= 0.5463
\end{aligned}$$

Menghitung sinyal output:

$$\begin{aligned}
y &= \frac{1 - e^{-y_{in}}}{1 + e^{-y_{in}}} \\
&= \frac{0.4210}{1.5790} \\
&= 0.2666
\end{aligned}$$

$$\delta = \frac{(x - y)}{2} * \left[1 + \left(\frac{1 - e^{-y \cdot \pi}}{1 + e^{-y \cdot \pi}} \right) \right] * \left[1 - \left(\frac{1 - e^{-y \cdot \pi}}{1 + e^{-y \cdot \pi}} \right) \right]$$

$$= \frac{0.4524}{2} * [1 + (0.2666)] * [1 - (0.2666)]$$

$$= 0.2262 * 1.2666 * 0.7334$$

$$= 0.2101$$

$$\Delta w_1 = \alpha * \delta * z_1$$

$$= 0.5 * 0.2101 * 0.1786$$

$$= 0.0187$$

$$\Delta w_2 = \alpha * \delta * z_2$$

$$= 0.5 * 0.2101 * 0.2125$$

$$= 0.0223$$

$$\Delta w_0 = \alpha * \delta$$

$$= 0.5 * 0.2101$$

$$= 0.1050$$

$$\delta_{in_1} = \delta * w_1$$

$$= 0.2101 * 0.4552$$

$$= 0.0956$$

$$\delta_{in_2} = \delta * w_2$$

$$= 0.2101 * 0.0552$$

$$= 0.0115$$

$$\delta_1 = \frac{\delta_{in_1}}{2} * \left[1 + \left(\frac{1 - e^{-\sigma \cdot \pi}}{1 + e^{\sigma \cdot \pi}} \right) \right] * \left[1 - \left(\frac{1 - e^{-\sigma \cdot \pi}}{1 + e^{\sigma \cdot \pi}} \right) \right]$$

$$= \frac{0.0956}{2} * [1 + 0.1786] * [1 - 0.1786]$$

$$= 0.0478 * 1.1786 * 0.8214$$

$$= 0.0462$$

$$\delta_2 = \frac{\delta \ln_2}{2} * \left[1 + \left(\frac{1 - e^{-x_2 \ln_2}}{1 + e^{x_2 \ln_2}} \right) \right] * \left[1 - \left(\frac{1 - e^{-x_2 \ln_2}}{1 + e^{x_2 \ln_2}} \right) \right]$$

$$= \frac{0.0115}{2} * [1 + (0.2125)] * [1 - (0.2125)]$$

$$= 0.0057 * 1.2125 * 0.7875$$

$$= 0.0054$$

$$\Delta v_{11} = \alpha * \delta_1 * x_1$$

$$= 0.5 * 0.0462 * 0.0000$$

$$= 0.0000$$

$$\Delta v_{12} = \alpha * \delta_2 * x_1$$

$$= 0.5 * 0.0054 * 0.0000$$

$$= 0.0000$$

$$\Delta v_{21} = \alpha * \delta_1 * x_2$$

$$= 0.5 * 0.0462 * 0.7190$$

$$= 0.0166$$

$$\Delta v_{22} = \alpha * \delta_2 * x_2$$

$$= 0.5 * 0.0054 * 0.7190$$

$$= 0.0019$$

$$\Delta v_{01} = \alpha * \delta_1$$

$$= 0.5 * 0.0462$$

$$= 0.0231$$

$$\Delta v_{02} = \alpha * \delta_2$$

$$= 0.5 * 0.0054$$

$$= 0.0027$$

$$\begin{aligned}w_1 &= w_1 + \Delta w_1 \\ &= 0.4552 + 0.0187 \\ &= 0.4739\end{aligned}$$

$$\begin{aligned}w_2 &= w_2 + \Delta w_2 \\ &= 0.0552 + 0.0223 \\ &= 0.0775\end{aligned}$$

$$\begin{aligned}w_0 &= w_0 + \Delta w_0 \\ &= 0.4534 + 0.1050 \\ &= 0.5584\end{aligned}$$

$$\begin{aligned}v_{11} &= v_{11} + \Delta v_{11} \\ &= 0.1548 + 0.0000 \\ &= 0.1548\end{aligned}$$

$$\begin{aligned}v_{12} &= v_{12} + \Delta v_{12} \\ &= 0.2886 + 0.0000 \\ &= 0.2886\end{aligned}$$

$$\begin{aligned}v_{21} &= v_{21} + \Delta v_{21} \\ &= 0.0141 + 0.0166 \\ &= 0.0307\end{aligned}$$

$$\begin{aligned}v_{22} &= v_{22} + \Delta v_{22} \\ &= 0.3741 + 0.0019 \\ &= 0.3760\end{aligned}$$

$$\begin{aligned}
v_{0_1} &= v_{0_1} + \Delta v_{0_1} \\
&= 0.3510 + 0.0231 \\
&= 0.3741
\end{aligned}$$

$$\begin{aligned}
v_{0_2} &= v_{0_2} + \Delta v_{0_2} \\
&= 0.1628 + 0.0027 \\
&= 0.1655
\end{aligned}$$

$$\begin{aligned}
\text{Jumlah Kuadrat Error (MSE)} &= 0.5 * (t - y)^2 \\
&= 0.5 * (0.7190 - 0.2666)^2 \\
&= 0.5 * 0.2046 \\
&= 0.1023
\end{aligned}$$

2. Iterasi II pada *input* 2

Berikut akan dijabarkan perhitungan yang terjadi pada Epoch I (iterasi I) dengan menggunakan *input* $x = (0.0000 ; 0.7190)$:

a. Operasi pada Hidden Layer:

$$\begin{aligned}
z_{in_1} &= v_{0_1} + v_{11} * x_1 + v_{21} * x_2 \\
&= 0.3741 + (0.1548 * 0.0000) + (0.0307 * 0.7190) \\
&= 0.3741 + 0 + 0.0220 \\
&= 0.3961
\end{aligned}$$

$$\begin{aligned}
z_{in_2} &= v_{0_2} + v_{12} * x_1 + v_{22} * x_2 \\
&= 0.1655 + (0.2886 * 0.0000) + (0.3760 * 0.7190) \\
&= 0.1655 + 0 + 0.2703 \\
&= 0.4358
\end{aligned}$$

Menghitung sinyal output:

$$\begin{aligned} z_1 &= \frac{1 - e^{-z_{in_1}}}{1 + e^{-z_{in_1}}} \\ &= \frac{0.3271}{1.6729} \\ &= 0.1955 \end{aligned}$$

$$\begin{aligned} z_2 &= \frac{1 - e^{-z_{in_2}}}{1 + e^{-z_{in_2}}} \\ &= \frac{0.3533}{1.6467} \\ &= 0.2145 \end{aligned}$$

b. Operasi pada Output Layer:

$$\begin{aligned} y_{in} &= w_0 + w_1 * z_1 + w_2 * z_2 \\ &= 0.5584 + (0.4739 * 0.1955) + (0.0775 * 0.2145) \\ &= 0.5584 + 0.0926 + 0.0166 \\ &= 0.6932 \end{aligned}$$

Menghitung sinyal output:

$$\begin{aligned} y &= \frac{1 - e^{-y_{in}}}{1 + e^{-y_{in}}} \\ &= \frac{0.5001}{1.4999} \\ &= 0.3334 \end{aligned}$$

$$\begin{aligned} \delta &= \frac{(r - y)}{2} * \left[1 + \left(\frac{1 - e^{-y_{in}}}{1 + e^{-y_{in}}} \right) \right] * \left[1 - \left(\frac{1 - e^{-y_{in}}}{1 + e^{-y_{in}}} \right) \right] \\ &= \frac{0.3856}{2} * [1 + 0.3334] * [1 - 0.3334] \\ &= 0.1928 * 1.3334 * 0.6666 \end{aligned}$$

$$\begin{aligned}
&= 0.1713 \\
\Delta w_1 &= \alpha * \delta * z_1 \\
&= 0.5 * 0.1713 * 0.1955 \\
&= 0.0167 \\
\Delta w_2 &= \alpha * \delta * z_2 \\
&= 0.5 * 0.1713 * 0.2145 \\
&= 0.0183 \\
\Delta w_0 &= \alpha * \delta \\
&= 0.5 * 0.1713 \\
&= 0.0856 \\
\delta_{in_1} &= \delta * w_1 \\
&= 0.1713 * 0.4739 \\
&= 0.0811 \\
\delta_{in_2} &= \delta * w_2 \\
&= 0.1713 * 0.0775 \\
&= 0.0132 \\
\delta_1 &= \frac{\delta_{in_1}}{2} * \left[1 + \left(\frac{1 - e^{-z_{in_1}}}{1 + e^{z_{in_1}}} \right) \right] * \left[1 - \left(\frac{1 - e^{-z_{in_1}}}{1 + e^{z_{in_1}}} \right) \right] \\
&= \frac{0.0811}{2} * [1 + 0.1955] * [1 - 0.1955] \\
&= 0.0405 * 1.1955 * 0.8045 \\
&= 0.0389 \\
\delta_2 &= \frac{\delta_{in_2}}{2} * \left[1 + \left(\frac{1 - e^{-z_{in_2}}}{1 + e^{z_{in_2}}} \right) \right] * \left[1 - \left(\frac{1 - e^{-z_{in_2}}}{1 + e^{z_{in_2}}} \right) \right]
\end{aligned}$$

$$= \frac{0.0132}{2} * [1 + 0.2145] * [1 - 0.2145]$$

$$= 0.0066 * 1.2145 * 0.7855$$

$$= 0.0062$$

$$\Delta v_{11} = \alpha * \delta_1 * x_1$$

$$= 0.5 * 0.0389 * 0.0000$$

$$= 0.0194$$

$$\Delta v_{12} = \alpha * \delta_2 * x_1$$

$$= 0.5 * 0.0062 * 0.0000$$

$$= 0.0031$$

$$\Delta v_{21} = \alpha * \delta_1 * x_2$$

$$= 0.5 * 0.0389 * 0.7190$$

$$= 0.0139$$

$$\Delta v_{22} = \alpha * \delta_2 * x_2$$

$$= 0.5 * 0.0062 * 0.7190$$

$$= 0.0022$$

$$\Delta v_{01} = \alpha * \delta_1$$

$$= 0.5 * 0.0389$$

$$= 0.0194$$

$$\Delta v_{02} = \alpha * \delta_2$$

$$= 0.5 * 0.0062$$

$$= 0.0021$$

$$w_1 = w_1 + \Delta w_1$$

$$= 0.4739 + 0.0167$$

$$\begin{aligned}
&= 0.4906 \\
w_2 &= w_2 + \Delta w_2 \\
&= 0.0775 + 0.0183 \\
&= 0.0958 \\
w_0 &= w_0 + \Delta w_0 \\
&= 0.5584 + 0.0856 \\
&= 0.6440 \\
v_{11} &= v_{11} + \Delta v_{11} \\
&= 0.1548 + 0.0194 \\
&= 0.1742 \\
v_{12} &= v_{12} + \Delta v_{12} \\
&= 0.2886 + 0.0031 \\
&= 0.2917 \\
v_{21} &= v_{21} + \Delta v_{21} \\
&= 0.0307 + 0.0139 \\
&= 0.0446 \\
v_{22} &= v_{22} + \Delta v_{22} \\
&= 0.3760 + 0.0022 \\
&= 0.3782 \\
v_{01} &= v_{01} + \Delta v_{01} \\
&= 0.3741 + 0.0194 \\
&= 0.3935 \\
v_{02} &= v_{02} + \Delta v_{02}
\end{aligned}$$

$$= 0.1655 + 0.0021$$

$$= 0.1676$$

$$\begin{aligned} \text{Jumlah Kuadrat Error (MSE)} &= 0.5 * (t - y)^2 \\ &= 0.5 * (0.7190 - 0.3334)^2 \\ &= 0.0743 \end{aligned}$$

- **Pengujian**

Setelah perhitungan pelatihan selesai dilakukan atau telah mencapai nilai maksimum iterasi, maka dilakukan pengujian dimisalkan inputnya adalah :

	1	2
1	0,0512	0.6253

Dari *input*-an di atas, maka didapatkan:

$$t \text{ (target)} = 0.0512 + 0.6253 = 0.6765$$

$$x = (0.0512 ; 0.6253)$$

a. Operasi pada Hidden Layer:

$$\begin{aligned} z_{in1} &= v_{01} + v_{11} * x_1 + v_{21} * x_2 \\ &= 0.3935 + (0.1742 * 0.0512) + (0.0446 * 0.6253) \\ &= 0.3935 + 0.0089 + 0.0278 \\ &= 0.4302 \end{aligned}$$

$$\begin{aligned} z_{in2} &= v_{02} + v_{12} * x_1 + v_{22} * x_2 \\ &= 0.1676 + (0.2917 * 0.0512) + (0.3782 * 0.6253) \\ &= 0.1676 + 0.0149 + 0.2364 \end{aligned}$$

$$\begin{aligned}
 &= 0.4189 \\
 z_1 &= \frac{1 - e^{-z_1 \cdot m_1}}{1 + e^{-z_1 \cdot m_1}} \\
 &= \frac{0.3497}{1.6503} \\
 &= 0.2119 \\
 z_2 &= \frac{1 - e^{-z_2 \cdot m_2}}{1 + e^{-z_2 \cdot m_2}} \\
 &= \frac{0.3423}{1.6577} \\
 &= 0.2064 \\
 y_{in} &= w_0 + w_1 * z_1 + w_2 * z_2 \\
 &= 0.6440 + (0.4906 * 0.2119) + (0.0958 * 0.2064) \\
 &= 0.6440 + 0.1039 + 0.3022 \\
 &= 1.0501 \\
 y &= \frac{1 - e^{-y_{in}}}{1 + e^{-y_{in}}} \\
 &= \frac{0.6501}{1.3499} \\
 &= 0.4815
 \end{aligned}$$

Dari hasil semua perhitungan di atas, bandingkan MSE pelatihan dengan MSE terakhir.

Di mana, target error yang telah ditentukan sebesar 0.01

$$\begin{aligned}
 MSE_1 &= 0.5 * (t_{p1} - y_{p1})^2 + (t_u - y_u)^2 \\
 &= 0.5 * (0.8249 - 0.3616)^2 + (0.6765 - 0.4815)^2 \\
 &= 0.1263
 \end{aligned}$$

$$\begin{aligned}
 \text{MSE}_2 &= 0.5 * (t_{p2} - y_{p2})^2 + (t_u - y_u)^2 \\
 &= 0.5 * (0.7190 - 0.3334)^2 + (0.6765 - 0.4815)^2 \\
 &= 0.0933
 \end{aligned}$$

Dari perhitungan di atas, maka 0.1263 dan $0.0933 \geq 0.01$. Sehingga jumlah MSE yang dikenali tidak ada atau 0.

$$y_{\text{pelatihan}} = 0.3334$$

$$y_{\text{pengujian}} = 0.4815$$

Maka, $y_{\text{pelatihan}} \leq y_{\text{pengujian}}$. Sehingga:

$$\begin{aligned}
 \text{Hasil} &= y_{\text{pengujian}} - y_{\text{pelatihan}} \\
 &= 0.4815 - 0.3334 \\
 &= 0.1481
 \end{aligned}$$

Dari perhitungan di atas, $0,1481 \geq 0.02$ (target error₁).

$$\begin{aligned}
 \text{persentase} &= \frac{\text{total MSE yang dikenali}}{\text{total pelatihan}} \times 100\% \\
 &= \frac{0}{2} \times 100\% \\
 &= 0\%
 \end{aligned}$$

Hasil persentase tersebut sebesar 0%. Sehingga $0\% \leq 90\%$ (batas persentase). Berarti citra tanda tangan tidak dikenali oleh sistem.

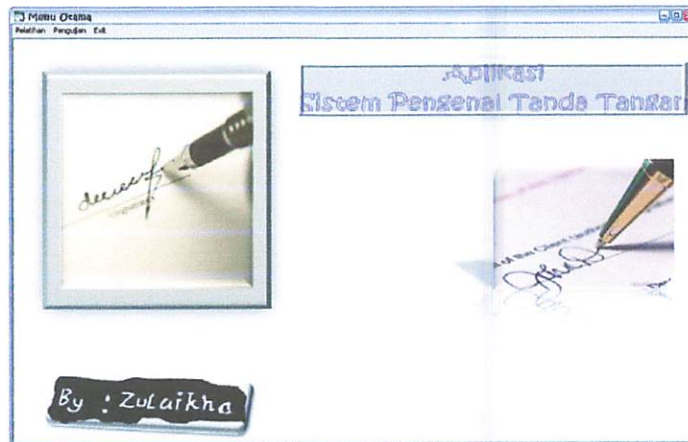
BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Implementasi merupakan kegiatan akhir dari proses penerapan sistem dimana sistem akan dioperasikan. Pada tahap implementasi sistem, merupakan hasil dari tahap analisis dan perancangan sistem pengenalan tanda tangan dengan jaringan saraf tiruan metode propagasi balik pada bab sebelumnya.

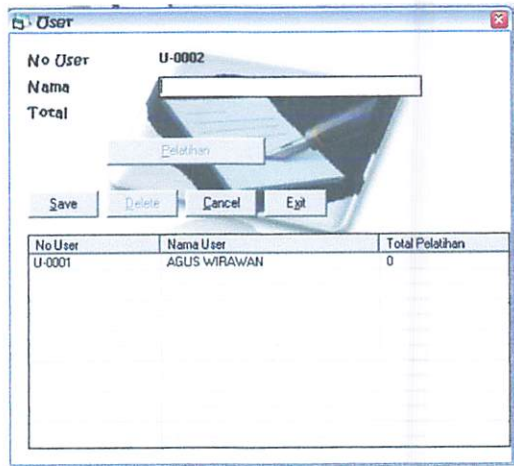
4.1.1 Form Menu Utama



Gambar 4.1
Form Menu Utama

Form di mana aplikasi pertama akan muncul. Pada form Menu Utama terdapat dua menu utama, yaitu Form Pelatihan dan Form Pengujian.

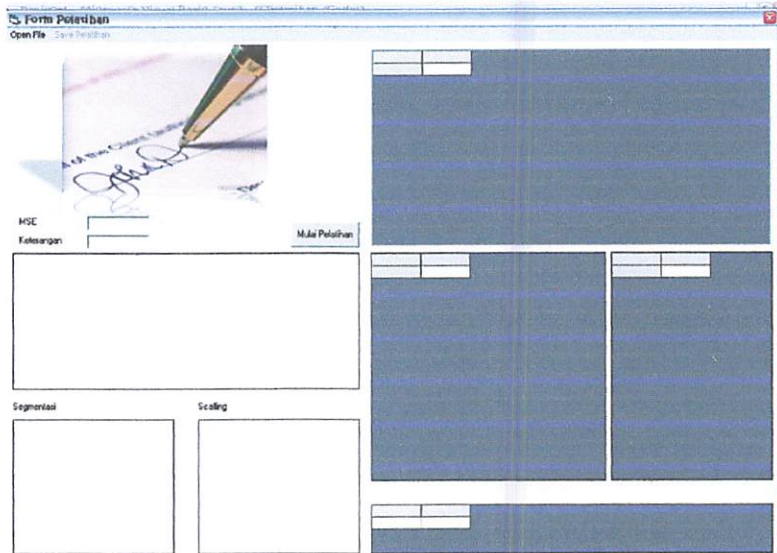
4.1.2 Form User



Gambar 4.2
Form User

Jika memilih menu Pelatihan, maka akan muncul form User. Pada form User, kita dapat menambah, meng-*update* dan menghapus user. Jika *user* telah terdaftar, maka akan muncul form Pelatihan.

4.1.3 Form Pelatihan



Gambar 4.3
Form Pelatihan

Form Pelatihan muncul setelah form User. Form ini berisi hasil pengolahan citra hingga menghasilkan matriks ukuran 100x1. Terdapat *text box*, untuk *jumlah hidden layer* yang digunakan. Selain itu terdapat tiga *picture box*, digunakan sebagai hasil pengolah citra tanda tangan, seperti *cropping*, segmentasi dan *scalling*. Dan empat tabel yang nantinya akan memunculkan nilai matriks ukuran 100x100, 10x100, 10x10 dan 100x1. Setelah proses pelatihan, maka disimpan pada menu Save Pelatihan untuk proses pengujian.

4.1.4 Form Pengujian

Gambar 4.4
Form Pengujian

Form Pengujian untuk mengetahui presentase kemiripan tanda tangan. Form ini dijalankan setelah form Pelatihan. Tabel input jaringan merupakan input an pada tahap pengenalan, yaitu dengan ukuran matriks 100x1. Tombol

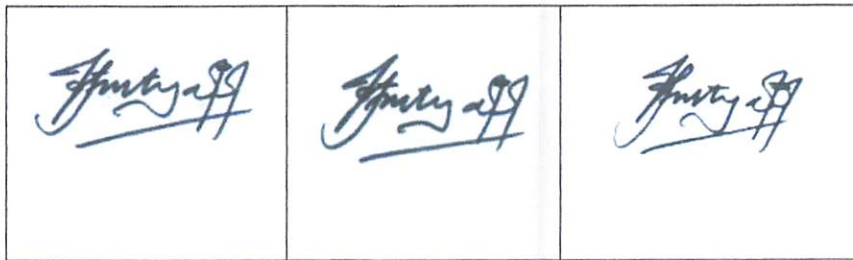
comparison process digunakan untuk menentukan hasil pengujian tanda tangan dalam bentuk persen.

4.2 Pengujian Sistem

Pada pengujian sistem pengenalan tanda tangan ini, sistem terbagi menjadi dua tahap, yaitu tahap pelatihan dan tahap pengenalan.

4.2.1 Tahap Pelatihan

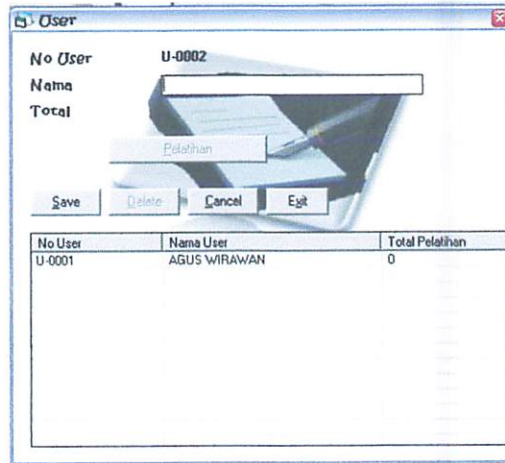
Pada tahap pelatihan, terdapat 3 (tiga) buah citra tanda tangan milik seseorang yang akan dilatih/dipelajari.



Gambar 4.5
Citra Tanda Tangan Tahap Pelatihan

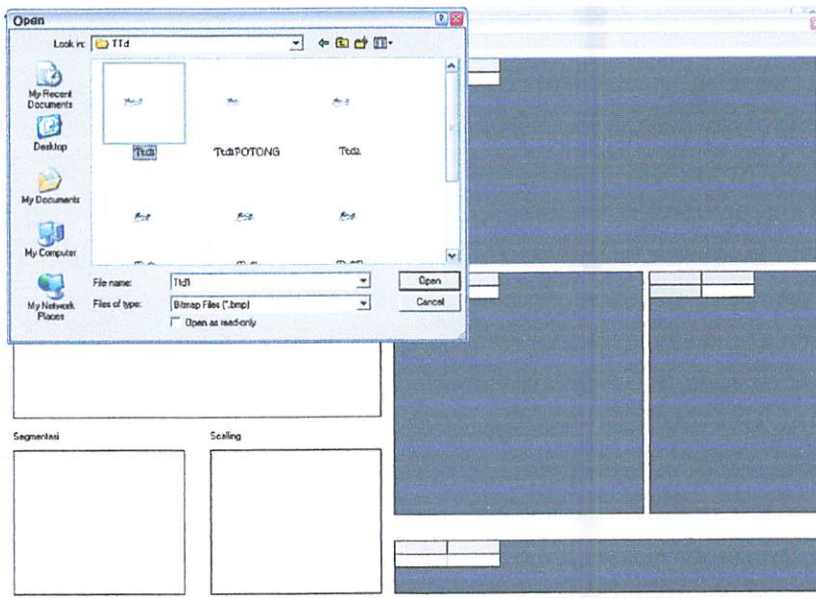
Adapun langkah-langkahnya sebagai berikut:

1. Jalankan program aplikasi pengenalan tanda tangan.
2. Pilih menu Pelatihan, dan muncul form User. Jika *user* belum terdaftar, maka isi kolom Nama kemudian pilih tombol Save. Jika *user* telah terdaftar, klik 2 kali nama *user* yang akan melakukan proses pelatihan. Lalu tekan tombol Pelatihan.



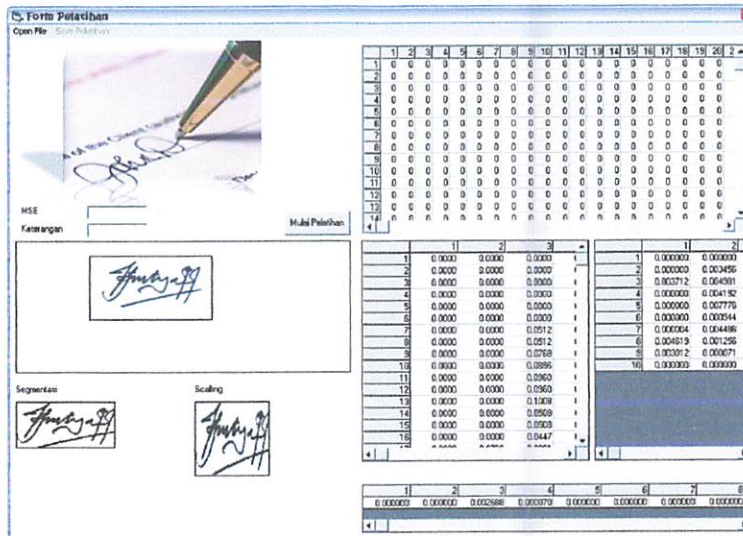
Gambar 4.6
Pengisian Nama *User*

3. Kemudian masuk ke form Pengenalan. Pada form ini, pilih menu Open File untuk mengambil citra tanda tangan.



Gambar 4.7
Mengambil File Citra Tanda Tangan

4. Kemudian tekan tombol Mulai Pelatihan. Lalu pilih menu Save Pelatihan, untuk disimpan ke database.

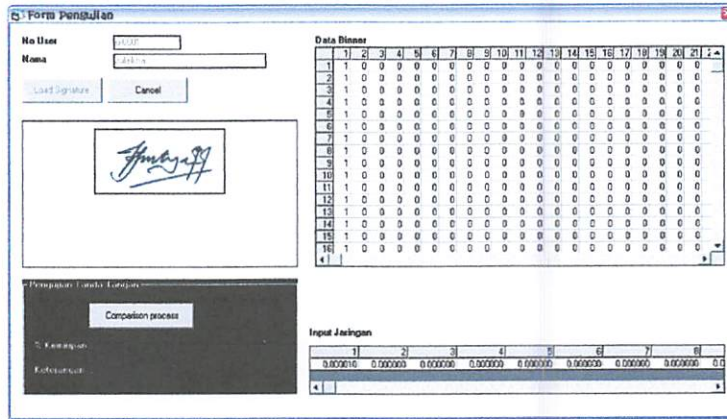


Gambar 4.8
Tampilan Hasil Pelatihan

4.2.2 Tahap Pengenalan

Untuk menjalankan tahap pengenalan, *user* harus sudah melakukan tahap pengenalan. Berikut langkah-langkah pada tahap pengenalan:

1. Jalankan program aplikasi sistem pengenalan tanda tangan.
2. Pada form Menu Utama, pilih menu Pengujian. Isi nomer *user* yang akan diuji, lalu pilih tombol Load Signature untuk mengambil citra tangan yang akan diuji. Kemudian tekan tombol *Comparison Process*, untuk mengetahui hasil kemiripan.



Gambar 4.9
Tampilan Hasil Pengujian

4.3 Analisa Hasil





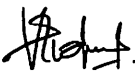



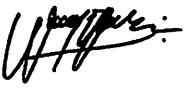



Dari hasil percobaan dengan Visual Basic 6.0 pada sistem ini, dilakukan 3 (tiga) kali pelatihan, dengan ketentuan konfigurasi sebagai berikut:

- Jumlah *input layer* : 100
- Jumlah *hidden layer* : 50
- Learning rate* (α) : 0.5
- Maksimum Epoch : 100
- Target Error* : 0.01
- Target Error*₁ : 0.02

Konfigurasi di atas didapat berdasarkan dari hasil percobaan. Pada tahap pengujian ini, digunakan batas persentase sebagai dasar penilaian dikenali sebuah tanda tangan atau tidak. Batas persentase yang digunakan sebesar $\geq 90\%$.

Tabel 4.1
 Hasil Persentase Citra Tanda Tangan

No.	Pelatihan		Pengenalan/Pengujian		Persentase (%)
	Citra Tanda Tangan	MSE	Citra Tanda Tangan	MSE	
1.		0.0000		0.0011	67
		0.0023			
		0.0038			
2.		0.0029		0.0006	33
		0.007			
		0.0124			
3.		0.0004		0.0005	50
		0.0005			
		0.0005			
4.		0.0000		0.0001	100
		0.0015			
		0.0002			
5.		0.0000		0.0004	100
		0.0002			
		0.0009			

No.	Pelatihan		Pengenalan/Pengujian		Persentase (%)
	Citra Tanda Tangan	MSE	Citra Tanda Tangan	MSE	
6.		0.0000		0.0008	100
		0.0002			
		0.0011			
7.		0.0009		0.0004	31
		0.0036			
		0.0060			
8.		0.0001		0.0010	19
		0.0004			
		0.0009			

BAB V

PENUTUP

5.1 Kesimpulan

Dari percobaan perancangan dan pembuatan aplikasi sistem pengenalan tanda tangan ini, dapat diambil kesimpulan sebagai berikut:

- a. Tidak konsistennya user dalam membubuhkan tanda tangan menyebabkan rendahnya pengenalan tanda tangan
- b. Dengan jaringan saraf tiruan propagasi balik ini, ditentukan maksimum epoch sebanyak 100, target *error* adalah 0.01, target *error*₁ 0.02. Konstanta alpha (*learning rate*) 0.5, jumlah sel lapisan tersembunyi (*hidden layer*) 50 buah, dan fungsi aktivasi yang dipakai adalah fungsi *sigmoid bipolar*.
- c. Dengan *hidden layer* sebanyak 50 buah, dibutuhkan waktu kurang lebih 1 menit, pada waktu melakukan tahap pelatihan.
- d. Semakin banyak jumlah *hidden layer*, pelatihan semakin lama namun hasil lebih akurat. Dan jika *learning rate* semakin besar, jaringan semakin cepat belajar tapi hasilnya kurang akurat.
- e. Hasil yang didapatkan dipengaruhi oleh beberapa faktor, yaitu bergantung pada jumlah *hidden layer*, epoch, target *error*, *learning rate* dan MSE (*Mean Square Error*).
- f. Kekurang akuratan hasil juga dipengaruhi oleh data pelatihan yang dimasukkan.

- g. Tahap pelatihan/pembelajaran dilakukan sebanyak 3 (tiga) kali pelatihan.
- h. Area kertas citra tanda tangan yang digunakan berukuran 10x6 cm.
- i. Berdasarkan percobaan yang dilakukan, batas persentase yang digunakan sebagai dasar penilaian dikenali sebuah tanda tangan atau tidak adalah $\geq 90\%$.
- j. Persentase yang tinggi disebabkan karena adanya kemiripan antara citra tanda tangan yang ingin dikenali dengan satu atau lebih citra tanda tangan yang telah dilatih, begitu pula sebaliknya dengan persentase yang rendah.
- k. Dengan menggunakan jaringan saraf tiruan metode propagasi balik, aplikasi sistem pengenalan tanda tangan ini belum dapat mengenali citra tanda tangan secara maksimal.

5.2 Saran

- a. Sistem akan lebih baik jika ada proses penghilangan noise.
- b. Perlu dilakukan penelitian lebih lanjut dengan model jaringan saraf tiruan yang berbeda.
- c. Perlu dikembangkan untuk mengurangi resiko pemalsuan tanda tangan dari pihak yang berbeda dengan pemilik tanda tangan tersebut.

DAFTAR PUSTAKA

- [1] Puspitaningrum, Diyah. 2006 . “Jaringan Saraf Tiruan” ANDI: Yogyakarta
- [2] Widodo, Mr. “Pengantar Pengolahan Citra”
- [3] Ismail Yusuf Nakhoda, Ir, MT. “ Komputasi Cerdas Jaringan Saraf Tiruan”
- [4] “*Jaringan Saraf Tiruan*”. http://www.wikipedia.com/Jaringan_saraf_tiruan.html
- [5] Dwi Cahyo, Edwin. 2009 . “*Sistem Pengenalan Nomer Kendaraan Berbasis Citra Menggunakan Metode Template Matching*” . Institut Teknologi Nasional Malang.
- [6] Wardani Kusuma Setiawa, Prof , MT. “*Paradigma Sistem Cerdas*” . Surabaya
- [7] Kusumadewi, Sri . 2003 . “*Artificial Intelligence*” . Graha Ilmu: Yogyakarta
- [8] Muis, Saludin. 2006 . “*Teknik Jaringan Saraf Tiruan*” . 2006. Graha Ilmu : Yogyakarta
- [9] Basuki, Ahmad . 2005. “ *Pengolahan Citra Digital Menggunakan Visual Basic*” . Graha Ilmu : Yogyakarta



- **FORM USER**

```

Option Explicit
Dim Sqlcmd As String
Private Type User_Rec
    No_User As String
    Nama As String
End Type
Dim User As User_Rec
Private Sub Setting_Awal()
Dim i As Integer
    LUser(0).Caption = vbNullString
    TxtUser(0).Enabled = True
    TxtUser(0).Text = vbNullString
    LUser(1).Caption = vbNullString
    For i = CmdButton.LBound To CmdButton.UBound
        If i = 1 Then
            CmdButton(i).Enabled = False
        Else
            CmdButton(i).Enabled = True
        End If
    Next i
    CmdButton(4).Enabled = False
    LUser(0).Caption = Get_No_User
    CmdButton(0).Caption = "&Save"
    Laporan_User
End Sub
Private Function Get_No_User() As String
    Sqlcmd = "Select Top 1 No_User From T_User order by No_User desc "
    Set MyRs = MyDb.Execute(Sqlcmd)
    If Not (MyRs.BOF And MyRs.EOF) Then
        Get_No_User = "U-" & Format(CLng(Right(MyRs!No_User, 4)) + 1, "0000")
    Else
        Get_No_User = "U-0001"
    End If
End Function
Private Function Find_No_User(No_User As String) As Boolean
    Find_No_User = False
    Sqlcmd = "Select * From T_User Where No_user= " & No_User & " "
    Set MyRs = MyDb.Execute(Sqlcmd)
    If Not (MyRs.BOF And MyRs.EOF) Then
        Find_No_User = True
        With User
            .No_User = MyRs.Fields(0)
            .Nama = MyRs.Fields(1)
        End With
    End If
End Function
Private Function Get_Total_Pelatihan(No_User As String) As Long

```

```

Get_Total_Pelatihan = 0
Sqlcmd = "Select Count(No_Pelatihan)as No_Pelatihan1 From T_Pelatihan Where
No_User=" & No_User & "" "
Set MyRs = MyDb.Execute(Sqlcmd)
If Not (MyRs.BOF And MyRs.EOF) Then
    If Not IsNull(MyRs.Fields(0)) Then
        Get_Total_Pelatihan = MyRs.Fields(0)
    End If
End If
End Function
Private Sub Laporan_User()
    Dim Ch As ColumnHeader
    Dim Temp As String
    Dim Temp1 As Byte
    Dim strItem As String
    Dim mItem As Variant
    Tabel_User.Width = 55
    Tabel_User.ColumnHeaders.Clear
    Set Ch = Tabel_User.ColumnHeaders.Add(, "No User", 15)
    Set Ch = Tabel_User.ColumnHeaders.Add(, "Nama User", 25)
    Set Ch = Tabel_User.ColumnHeaders.Add(, "Total Pelatihan", 15)
    Tabel_User.GridLines = True
    With Tabel_User
        .Sorted = False
        .ListItems.Clear
        Sqlcmd = "Select Top 1 * From T_User "
        Sqlcmd = Sqlcmd & "order by No_User "
        Set MyRs = MyDb.Execute(Sqlcmd)
        Do Until (MyRs.EOF)
            Temp = MyRs.Fields(0).Value

            Set mItem = .ListItems.Add(, MyRs.Fields(0).Value)
            mItem.SubItems(1) = MyRs.Fields(1).Value
            mItem.SubItems(2) = Get_Total_Pelatihan(Temp)
            'Refresh for appearance
            If .ListItems.Count > 15 Then
                DoEvents
                .Refresh
            End If
            Sqlcmd = "Select Top 1 * From T_User Where No_User > " & Temp & "" "
            Sqlcmd = Sqlcmd & "order by No_User "
            Set MyRs = MyDb.Execute(Sqlcmd)
        Loop
        .Refresh
    End With
End Sub
Private Sub Save_Data()
    With User
        .No_User = LUser(0).Caption
    End With
End Sub

```

```

.Nama = TxtUser(0).Text
If CmdButton(0).Caption = "&Save" Then
    Sqlcmd = "Insert into T_User Values('" & .No_User & "','" & .Nama & "')"
Else
    Sqlcmd = "Update T_User Set Nama = '" & .Nama & "' Where No_User=" &
.No_User & "' "
End If
End With
MyDb.BeginTrans
MyDb.Execute (Sqlcmd)
MyDb.CommitTrans
Setting_Awal
End Sub
Private Sub Delete_Data()
    With User
        .No_User = LUser(0).Caption
        MyDb.BeginTrans
        Sqlcmd = "Delete * From T_User Where No_User=" & .No_User & """"
        MyDb.Execute (Sqlcmd)
        Sqlcmd = "Delete * From T_Pelatihan Where No_User=" & .No_User & """"
        MyDb.Execute (Sqlcmd)
        Sqlcmd = "Delete * From T_BobotBias2Hidden Where No_Pelatihan Like '" &
.No_User & "%' "
        MyDb.Execute (Sqlcmd)
        Sqlcmd = "Delete * From T_BobotInput2Hidden Where No_Pelatihan Like '" &
.No_User & "%' "
        MyDb.Execute (Sqlcmd)
        Sqlcmd = "Delete * From T_BobotBias2Output Where No_Pelatihan Like '" &
.No_User & "%' "
        MyDb.Execute (Sqlcmd)
        Sqlcmd = "Delete * From T_BobotHidden2Output Where No_Pelatihan Like '" &
.No_User & "%' "
        MyDb.Execute (Sqlcmd)
        MyDb.CommitTrans
    End With
    Setting_Awal
End Sub

Private Sub CmdButton_Click(Index As Integer)
    Select Case Index
        Case 0: Save_Data
        Case 1: Delete_Data
        Case 2: Setting_Awal
        Case 3: Unload Me
        Case 4
            No_User1 = LUser(0).Caption
            FPelatihan.Show
    End Select
End Sub

```

```

Private Sub Form_Load()
    Setting_Awal
End Sub

```

```

Private Sub Tabel_User_DblClick()
    User.No_User = Tabel_User.SelectedItem.Text
    If Find_No_User(User.No_User) Then
        With User
            Setting_Awal
            LUser(0).Caption = .No_User
            TxtUser(0).Text = .Nama
            LUser(1).Caption = Get_Total_Pelatihan(.No_User)
            CmdButton(0).Caption = "Update"
            CmdButton(1).Enabled = True
            CmdButton(4).Enabled = True
        End With
    End If
End Sub

```

- **FORM PELATIHAN**

```

Option Explicit
Private Type Pelatihan_Rec
    No_Pelatihan As String
    No_User As String
    Nilai_Output As Double
    Nilai_Target As Double
End Type
Dim Pelatihan As Pelatihan_Rec
Private Type BobotBias2Hidden_Rec
    No_V0 As String
    No_Pelatihan As String
    Nilai_V0 As Double
End Type
Dim BobotBias2Hidden1 As BobotBias2Hidden_Rec
Private Type BobotInput2Hidden_Rec
    No_V As String
    No_Pelatihan As String
    Nilai_V As Double
End Type
Dim BobotInput2Hidden1 As BobotInput2Hidden_Rec

Private Type BobotBias2Output_Rec
    No_W0 As String
    No_Pelatihan As String
    Nilai_W0 As Double
End Type

```

```

Dim BobotBias2Output1 As BobotBias2Output_Rec
Private Type BobotHidden2Output_Rec
    No_W As String
    No_Pelatihan As String
    Nilai_W As Double
End Type
Dim BobotHidden2Output1 As BobotHidden2Output_Rec
Dim Sqlcmd As String
Dim Hasil_Binner(99, 99) As Byte
Dim Hasil_Pic(9, 99) As Double
Dim Hasil_Pic1(9, 9) As Double
Dim XX As Long, YY As Long
Dim XX2 As Long, YY2 As Long
Dim Status_Cropping As Boolean
Dim Status_OpenFile As Boolean

Private Function Get_No_Pelatihan(No_User) As String
    Sqlcmd = "Select Top 1 No_Pelatihan From T_Pelatihan Where No_User=" &
No_User & " order by No_Pelatihan Desc "
    Set MyRs = MyDb.Execute(Sqlcmd)
    If Not (MyRs.BOF And MyRs.EOF) Then
        Get_No_Pelatihan = No_User & Format(CLng(Right(MyRs.Fields(0), 1)) + 1, "0")
    Else
        Get_No_Pelatihan = No_User & "1"
    End If
End Function

Private Function Get_No_BobotV0(No_Pelatihan) As String
    Sqlcmd = "Select Top 1 No_Bobot_V0 From T_BobotBias2Hidden Where
No_Pelatihan=" & No_Pelatihan & " order by No_Bobot_V0 Desc "
    Set MyRs = MyDb.Execute(Sqlcmd)
    If Not (MyRs.BOF And MyRs.EOF) Then
        Get_No_BobotV0 = No_Pelatihan & Format(CLng(Right(MyRs.Fields(0), 3)) + 1,
"000")
    Else
        Get_No_BobotV0 = No_Pelatihan & "001"
    End If
End Function

Private Function Get_No_BobotV(No_Pelatihan) As String
    Sqlcmd = "Select Top 1 No_Bobot_V From T_BobotInput2Hidden Where
No_Pelatihan=" & No_Pelatihan & " order by No_Bobot_V Desc "
    Set MyRs = MyDb.Execute(Sqlcmd)
    If Not (MyRs.BOF And MyRs.EOF) Then
        Get_No_BobotV = No_Pelatihan & Format(CLng(Right(MyRs.Fields(0), 5)) + 1,
"00000")
    Else
        Get_No_BobotV = No_Pelatihan & "00001"
    End If
End Function

Private Function Get_No_BobotW0(No_Pelatihan) As String

```

```

    Sqlcmd = "Select Top 1 No_Bobot_W0 From T_BobotBias2Output Where
No_Pelatihan= '' & No_Pelatihan & '' order by No_Bobot_W0 Desc "
    Set MyRs = MyDb.Execute(Sqlcmd)
    If Not (MyRs.BOF And MyRs.EOF) Then
        Get_No_BobotW0 = No_Pelatihan & Format(CLng(Right(MyRs.Fields(0), 2)) + 1,
"00")
    Else
        Get_No_BobotW0 = No_Pelatihan & "01"
    End If
End Function
Private Function Get_No_BobotW(No_Pelatihan) As String
    Sqlcmd = "Select Top 1 No_Bobot_W From T_BobotHidden2Output Where
No_Pelatihan= '' & No_Pelatihan & '' order by No_Bobot_W Desc "
    Set MyRs = MyDb.Execute(Sqlcmd)
    If Not (MyRs.BOF And MyRs.EOF) Then
        Get_No_BobotW = No_Pelatihan & Format(CLng(Right(MyRs.Fields(0), 3)) + 1,
"000")
    Else
        Get_No_BobotW = No_Pelatihan & "001"
    End If
End Function
Private Function Find_Pelatihan(No_User As String) As Boolean
    Find_Pelatihan = False
    Sqlcmd = "Select Top 1 * From T_Pelatihan Where No_User= '' & No_User & '' Order
by No_Pelatihan desc "
    Set MyRs = MyDb.Execute(Sqlcmd)
    If Not (MyRs.BOF And MyRs.EOF) Then
        Find_Pelatihan = True
        With Pelatihan
            .No_Pelatihan = MyRs.Fields(0)
            .No_User = MyRs.Fields(1)
            .Nilai_Output = MyRs.Fields(2)
            .Nilai_Target = MyRs.Fields(3)
        End With
    End If
End Function
Private Function Juml_Target(No_User As String) As Long
    Juml_Target = 0
    Sqlcmd = "Select count(No_Pelatihan) From T_Pelatihan Where No_User= '' &
No_User & ''"
    Set MyRs = MyDb.Execute(Sqlcmd)
    If Not (MyRs.BOF And MyRs.EOF) Then
        Juml_Target = MyRs.Fields(0)
    End If
End Function
Private Sub Get_AllPelatihan(No_User As String)
Dim i As Long
    i = 0

```



```

Sqlcmd = "Select Top 1 * From T_Pelatihan Where No_User= '" & No_User & "' order
by No_Pelatihan "
Set MyRs = MyDb.Execute(Sqlcmd)
While Not (MyRs.BOF And MyRs.EOF)
    Pelatihan.No_Pelatihan = MyRs.Fields(0)
    Y(i) = MyRs.Fields(2)
    Target(i) = MyRs.Fields(3)
    i = i + 1
    Sqlcmd = "Select Top 1 * From T_Pelatihan Where No_User= '" & No_User & "'
And No_Pelatihan > '" & Pelatihan.No_Pelatihan & "' order by No_Pelatihan "
    Set MyRs = MyDb.Execute(Sqlcmd)
Wend
End Sub
Private Function Get_Last_No_Pelatihan(No_User As String) As String
    Get_Last_No_Pelatihan = ""
    Sqlcmd = "Select Top 1 * From T_Pelatihan Where No_User= '" & No_User & "' order
by No_Pelatihan Desc "
    Set MyRs = MyDb.Execute(Sqlcmd)
    If Not (MyRs.BOF And MyRs.EOF) Then
        Get_Last_No_Pelatihan = MyRs.Fields(0)
    End If
End Function
Private Sub Load_BobotBias2Hidden(No_Pelatihan As String)
    Dim i As Long
    i = 0
    Sqlcmd = "Select Top 1 * From T_BobotBias2Hidden Where No_Pelatihan= '" &
No_Pelatihan & "' order by No_Bobot_V0 "
    Set MyRs = MyDb.Execute(Sqlcmd)
    While Not (MyRs.BOF And MyRs.EOF)
        BobotBias2Hidden1.No_V0 = MyRs.Fields(0)
        BobotBias2Hidden(i) = MyRs.Fields(2)
        i = i + 1
        Sqlcmd = "Select Top 1 * From T_BobotBias2Hidden Where No_Pelatihan= '" &
No_Pelatihan & "' And No_Bobot_V0 > '" & BobotBias2Hidden1.No_V0 & "' order by
No_Bobot_V0 "
        Set MyRs = MyDb.Execute(Sqlcmd)
    Wend
End Sub
Private Sub Load_BobotInput2Hidden(No_Pelatihan As String)
    Dim i As Long
    i = 0
    Sqlcmd = "Select Top 1 * From T_BobotInput2Hidden Where No_Pelatihan= '" &
No_Pelatihan & "' order by No_Bobot_V "
    Set MyRs = MyDb.Execute(Sqlcmd)
    While Not (MyRs.BOF And MyRs.EOF)
        i = i + 1
        BobotInput2Hidden1.No_V = MyRs.Fields(0)
        If (i Mod 100) = 0 Then
            bobotinput2hidden(100 - 1, (i \ 100) - 1) = MyRs.Fields(2)
        End If
    Wend
End Sub

```

```

Else
    bobotinput2hidden((i Mod 100) - 1, i \ 100) = MyRs.Fields(2)
End If
DoEvents
Sqlcmd = "Select Top 1 * From T_BobotInput2Hidden Where No_Pelatihan= '" &
No_Pelatihan & "' And No_Bobot_V > '" & BobotInput2Hidden1.No_V & "' order by
No_Bobot_V "
Set MyRs = MyDb.Execute(Sqlcmd)
Wend
End Sub
Private Sub Load_BobotHidden2Output(No_Pelatihan As String)
Dim i As Long
i = 0
Sqlcmd = "Select Top 1 * From T_BobotHidden2Output Where No_Pelatihan= '" &
No_Pelatihan & "' order by No_Bobot_W "
Set MyRs = MyDb.Execute(Sqlcmd)
While Not (MyRs.BOF And MyRs.EOF)
    BobotHidden2Output1.No_W = MyRs.Fields(0)
    BobotHidden2Output(i) = MyRs.Fields(2)
    i = i + 1
    Sqlcmd = "Select Top 1 * From T_BobotHidden2Output Where No_Pelatihan= '" &
No_Pelatihan & "' And No_Bobot_W > '" & BobotHidden2Output1.No_W & "' order by
No_Bobot_W "
Set MyRs = MyDb.Execute(Sqlcmd)
Wend
End Sub
Private Sub Load_BobotBias2Output(No_Pelatihan As String)
Sqlcmd = "Select Top 1 * From T_BobotBias2Output Where No_Pelatihan= '" &
No_Pelatihan & "' "
Set MyRs = MyDb.Execute(Sqlcmd)
If Not (MyRs.BOF And MyRs.EOF) Then
    BobotBias2Output = MyRs.Fields(2)
End If
End Sub
Private Sub Save_Data_Pelatihan(No_Pelatihan As String, No_User As String,
Nilai_Output As Double, Nilai_Target As Double)
Sqlcmd = "Insert into T_Pelatihan Values('" & No_Pelatihan & "','" & No_User & "','"
& Nilai_Output & "','" & Nilai_Target & "'" "
MyDb.BeginTrans
MyDb.Execute (Sqlcmd)
MyDb.CommitTrans
End Sub
Private Sub Save_Data_BobotV0(No_Bobot_V0 As String, No_Pelatihan As String,
Nilai_Bobot_V0 As Double)
Sqlcmd = "Insert into T_BobotBias2Hidden Values('" & No_Bobot_V0 & "','" &
No_Pelatihan & "','" & Nilai_Bobot_V0 & "'" "
MyDb.BeginTrans
MyDb.Execute (Sqlcmd)

```

```

MyDb.CommitTrans
End Sub
Private Sub Delete_Data_BobotV0(No_Pelatihan As String)
    Sqlcmd = "Delete * From T_BobotBias2Hidden Where No_Pelatihan = " &
No_Pelatihan & " "
    MyDb.BeginTrans
    MyDb.Execute (Sqlcmd)
    MyDb.CommitTrans
End Sub

Private Sub Save_Data_BobotV(No_Bobot_V As String, No_Pelatihan As String,
Nilai_Bobot_V As Double)
    Sqlcmd = "Insert into T_BobotInput2Hidden Values(" & No_Bobot_V & "," &
No_Pelatihan & "," & Nilai_Bobot_V & ")" "
    MyDb.BeginTrans
    MyDb.Execute (Sqlcmd)
    MyDb.CommitTrans
End Sub
Private Sub Delete_Data_BobotV(No_Pelatihan As String)
    Sqlcmd = "Delete * From T_BobotInput2Hidden Where No_Pelatihan = " &
No_Pelatihan & " "
    MyDb.BeginTrans
    MyDb.Execute (Sqlcmd)
    MyDb.CommitTrans
End Sub
Private Sub SaVe_Data_BobotW0(No_Bobot_W0 As String, No_Pelatihan As String,
Nilai_Bobot_W0 As Double)
    Sqlcmd = "Insert into T_BobotBias2Output Values(" & No_Bobot_W0 & "," &
No_Pelatihan & "," & Nilai_Bobot_W0 & ")" "
    MyDb.BeginTrans
    MyDb.Execute (Sqlcmd)
    MyDb.CommitTrans
End Sub
Private Sub Delete_Data_BobotW0(No_Pelatihan As String)
    Sqlcmd = "Delete * From T_BobotBias2Output Where No_Pelatihan = " &
No_Pelatihan & " "
    MyDb.BeginTrans
    MyDb.Execute (Sqlcmd)
    MyDb.CommitTrans
End Sub

Private Sub Save_Data_BobotW(No_Bobot_W As String, No_Pelatihan As String,
Nilai_Bobot_W As Double)
    Sqlcmd = "Insert into T_BobotHidden2Output Values(" & No_Bobot_W & "," &
No_Pelatihan & "," & Nilai_Bobot_W & ")" "
    MyDb.BeginTrans
    MyDb.Execute (Sqlcmd)
    MyDb.CommitTrans
End Sub

```

```

Private Sub Delete_Data_BobotW(No_Pelatihan As String)
    Sqlcmd = "Delete * From T_BobotHidden2Output Where No_Pelatihan = "" &
No_Pelatihan & "" "
    MyDb.BeginTrans
    MyDb.Execute (Sqlcmd)
    MyDb.CommitTrans
End Sub

Private Sub Save_Data()
Dim i As Long, j As Long
If Find_Pelatihan(No_User1) Then
    Call Delete_Data_BobotV(Pelatihan.No_Pelatihan)
    Call Delete_Data_BobotV0(Pelatihan.No_Pelatihan)
    Call Delete_Data_BobotW0(Pelatihan.No_Pelatihan)
    Call Delete_Data_BobotW(Pelatihan.No_Pelatihan)
End If
With Pelatihan
.No_User = No_User1
.No_Pelatihan = Get_No_Pelatihan(.No_User)
.Nilai_Output = Round(Y(NumOutput - 1), 6)
.Nilai_Target = Round(CDbl(fg3.TextMatrix(1, 100)), 6)
Call Save_Data_Pelatihan(.No_Pelatihan, .No_User, .Nilai_Output, .Nilai_Target)
End With

With BobotInput2Hidden1
.No_Pelatihan = Pelatihan.No_Pelatihan
For j = 0 To Juml_HiddenLayer - 1
    For i = 0 To NumInput - 1
        .No_V = Get_No_BobotV(.No_Pelatihan)
        .Nilai_V = Round(bobotinput2hidden(i, j), 6)
        Call Save_Data_BobotV(.No_V, .No_Pelatihan, .Nilai_V)
    Next i
    With BobotBias2Hidden1
.No_Pelatihan = Pelatihan.No_Pelatihan
.No_V0 = Get_No_BobotV0(.No_Pelatihan)
.Nilai_V0 = Round(BobotBias2Hidden(j), 6)
Call Save_Data_BobotV0(.No_V0, .No_Pelatihan, .Nilai_V0)
    End With
    DoEvents
Next j
End With
With BobotBias2Output1
.No_Pelatihan = Pelatihan.No_Pelatihan
.No_W0 = Get_No_BobotW0(.No_Pelatihan)
.Nilai_W0 = Round(BobotBias2Output, 6)
Call SaVe_Data_BobotW0(.No_W0, .No_Pelatihan, .Nilai_W0)
End With
With BobotHidden2Output1
.No_Pelatihan = Pelatihan.No_Pelatihan

```

```

For i = 0 To Juml_HiddenLayer - 1
    .No_W = Get_No_BobotW(.No_Pelatihan)
    .Nilai_W = Round(BobotHidden2Output(i), 6)
    Call Save_Data_BobotW(.No_W, .No_Pelatihan, .Nilai_W)
Next i
End With

Text3.Text = "Selesai"
End Sub

Private Sub Open_File()
'On Error Resume Next
Dim sFile As String
With Cd1
    .DialogTitle = "Open"
    .CancelError = False

    .Filter = "Bitmap Files (*.bmp)" & "|*.bmp"
    .ShowOpen
    sFile = .FileName
    Status_OpenFile = True
End With
shp1.Visible = False
Gambar1.Picture = LoadPicture(Cd1.FileName)
End Sub

Private Sub Greyscale(ByVal Source As PictureBox, ByVal dest As PictureBox)
Dim i As Long
Dim j As Long
Dim W As Long, R As Long, G As Long, b As Long, X As Long
For i = 0 To Source.ScaleWidth - 1
    For j = 0 To Source.ScaleHeight - 1
        W = Source.Point(i, j)
        R = W And RGB(255, 0, 0)
        G = Int((W And RGB(0, 255, 0)) / 256)
        b = Int(Int((W And RGB(0, 0, 255)) / 256) / 256)
        X = 0.2 * R + 0.1 * G + 0.7 * b
        dest.PSet (i, j), RGB(X, X, X)
    Next j
Next i
End Sub

Private Sub monokromBiner(ByVal Source As PictureBox, ByVal dest As PictureBox,
NilaiBatas As Byte)
Dim i As Long
Dim j As Long
Dim W As Long, R As Long, G As Long, b As Long, X As Long
Dim n As Byte
For i = 0 To Source.ScaleWidth - 1
    For j = 0 To Source.ScaleHeight - 1

```

```

    W = Source.Point(i, j)
    R = W And RGB(255, 0, 0)
    G = Int((W And RGB(0, 255, 0)) / 256)
    b = Int(Int((W And RGB(0, 0, 255)) / 256) / 256)
    X = Int((R + G + b) / 3)
    If X < NilaiBatas Then n = 0 Else n = 255
    dest.PSet (i, j), RGB(n, n, n)
Next j
Next i
End Sub
Private Sub Tresholding(Source As PictureBox, dest As PictureBox, NilaiTresh As Byte)
Dim i As Long
Dim j As Long
Dim W As Long, R As Long, G As Long, b As Long, X As Long
Dim nw As Long, xth As Long
nw = 256 / Val(NilaiTresh)
For i = 0 To Gambar1.ScaleWidth - 1
    For j = 0 To Source.ScaleHeight - 1
        W = Source.Point(i, j)
        R = W And RGB(255, 0, 0)
        G = Int((W And RGB(0, 255, 0)) / 256)
        b = Int(Int((W And RGB(0, 0, 255)) / 256) / 256)
        X = Int((R + G + b) / 3)
        xth = nw * Int(X / nw)
        dest.PSet (i, j), RGB(xth, xth, xth)
    Next j
Next i
End Sub

Private Sub Command2_Click()
    Call monokromBiner(Gambar1, Gambar1, 128)
    Call Segmentasi_gb(Gambar1, 1, 1, Gambar1.ScaleWidth - 1, Gambar1.ScaleHeight - 1,
Gambar2)
    Gambar3.Picture = Gambar2.Picture
End Sub
Private Sub Binnarisasi(Source As PictureBox, NilaiBatas As Byte)
Dim i As Long
Dim j As Long
Dim W As Long, R As Long, G As Long, b As Long, X As Long
Dim n As Byte
For j = 0 To Source.ScaleHeight - 1
    For i = 0 To Source.ScaleWidth - 1
        W = Source.Point(i, j)
        R = W And RGB(255, 0, 0)
        G = Int((W And RGB(0, 255, 0)) / 256)
        b = Int(Int((W And RGB(0, 0, 255)) / 256) / 256)
        X = Int((R + G + b) / 3)
        If X < NilaiBatas Then Hasil_Binner(i, j) = 1 Else Hasil_Binner(i, j) = 0
    Next i

```

```

Next j
End Sub
Private Sub Resize_Horisontal()
Dim i As Integer, j As Integer, k As Integer, l As Integer
Dim Temp As String
For j = 0 To 99
    l = 0
    For i = 0 To 99 Step 10
        k = i
        Temp = ""
        While k < i + 10
            Temp = Temp + Trim(Str(Hasil_Binner(k, j)))
            k = k + 1
        Wend
        Hasil_Pic(l, j) = BinToDec(Temp)
        l = l + 1
    Next i
Next j
End Sub
Private Sub Resize_Vertikal()
Dim i As Integer, j As Integer, k As Integer, l As Integer
Dim Temp As Double
For i = 0 To 9
    l = 0
    For j = 0 To 99 Step 10
        k = j
        Temp = 0
        While k < j + 10
            Temp = Temp + Hasil_Pic(i, k)
            k = k + 1
        Wend
        Hasil_Pic1(i, l) = Temp
        l = l + 1
    Next j
Next i
End Sub
Private Sub Ekstrasi_Ciri()
Call Binmarisasi(Gambar3, 128)
Dim i As Integer, j As Integer
Fg0.Cols = 101
Fg0.Rows = 101
Fg0.ColWidth(0) = 350
For i = 1 To 100
    Fg0.TextMatrix(0, i) = Trim(Str(i))
    Fg0.ColWidth(i) = 350
Next
For j = 1 To 100
    Fg0.TextMatrix(j, 0) = Trim(Str(j))
Next

```

```

For j = 1 To 100
  For i = 1 To 100
    Fg0.TextMatrix(j, i) = Trim(Str(Hasil_Binner(i - 1, j - 1)))
  Next i
Next j
Call Resize_Horisontal
Call Resize_Vertikal
Call Input_Jaringan
End Sub
Private Sub Input_Jaringan()
  Dim i As Integer, j As Integer, k As Integer
  Fg1.Cols = 11
  Fg1.Rows = 101
  For i = 1 To 10
    Fg1.TextMatrix(0, i) = Trim(Str(i))
  Next
  For j = 1 To 100
    Fg1.TextMatrix(j, 0) = Trim(Str(j))
  Next
  For j = 1 To 100
    For i = 1 To 10
      Fg1.TextMatrix(j, i) = Format(Hasil_Pic(i - 1, j - 1), "#0.0000")
    Next i
  Next j

  fg2.Cols = 11
  fg2.Rows = 11
  For i = 1 To 10
    fg2.TextMatrix(0, i) = Trim(Str(i))
  Next
  For j = 1 To 10
    fg2.TextMatrix(j, 0) = Trim(Str(j))
  Next
  fg3.Cols = 101
  fg3.Rows = 2
  For i = 0 To 99
    fg3.TextMatrix(0, i) = Trim(Str(i + 1))
  Next i
  fg3.TextMatrix(0, 100) = "Target"
  k = 0
  For j = 1 To 10
    For i = 1 To 10
      fg2.TextMatrix(j, i) = Format(Hasil_Pic1(i - 1, j - 1) / 100, "#0.000000")
      fg3.TextMatrix(1, k) = fg2.TextMatrix(j, i)
      k = k + 1
    Next i
  Next j
  fg3.TextMatrix(1, 100) = 0
  For i = 0 To 99

```



```

    fg3.TextMatrix(1, 100) = Cdbl(fg3.TextMatrix(1, 100)) + Cdbl(fg3.TextMatrix(1, i))
Next i
End Sub

```

```

Private Sub Command5_Click()
Dim inputA(100) As Double
Dim i As Integer
    Juml_HiddenLayer = 50
    NumInput = 100
    Command5.Enabled = False
If Not Find_Pelatihan(No_User1) Then
    NumOutput = 1
    InitBackpro
    Init_Bobot
Else
    NumOutput = Juml_Target(No_User1) + 1
    InitBackpro
    Get_AllPelatihan No_User1
    Pelatihan.No_Pelatihan = Get_Last_No_Pelatihan(No_User1)
    Call Load_BobotBias2Hidden(Pelatihan.No_Pelatihan)
    Call Load_BobotInput2Hidden(Pelatihan.No_Pelatihan)
    Call Load_BobotBias2Output(Pelatihan.No_Pelatihan)
    Call Load_BobotHidden2Output(Pelatihan.No_Pelatihan)
End If
Target(NumOutput - 1) = 0
For i = 0 To 99
    inputA(i) = Cdbl(fg3.TextMatrix(1, i))
Next
Target(NumOutput - 1) = Round(Cdbl(fg3.TextMatrix(1, 100)), 6)

```

```

List1.Clear
For i = 1 To 101
    Call FeedForward(inputA, NumOutput - 1)
    Text2.Text = Round(Total_MSE(Target, Y), 6)
    If Cdbl(Text2.Text) <= Error_Rate Then
        Text3.Text = "berhenti"
        Exit For
    End If
    If i < 101 Then
        Call BackPropagasi(inputA, NumOutput - 1)
    End If
    Exit For
Next
Command5.Enabled = True
Menu(1).Enabled = True
End Sub

```

```

Private Sub Form_Load()
    Menu(1).Enabled = False

```

```
Status_Cropping = False
Status_OpenFile = False
End Sub
```

```
Private Sub Gambar1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y
As Single)
```

```
    On Error Resume Next
    If Button = 1 Then
        If Status_OpenFile Then
            XX = X
            YY = Y
            XX2 = X
            YY2 = Y
            shp1.Shape = 0
            shp1.Visible = True
            shp1.Left = X
            shp1.Top = Y
            shp1.Width = 0
            shp1.Height = 0
            Status_Cropping = True
            Status_OpenFile = False
        End If
    End If
    Err.Clear
End Sub
```

```
Private Sub Gambar1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
```

```
    On Error Resume Next
    If Button = 1 And Status_Cropping Then
        XX2 = X
        YY2 = Y
        shp1.Left = IIf(X > XX, XX, X)
        shp1.Top = IIf(Y > YY, YY, Y)
        shp1.Width = Abs(X - XX)
        shp1.Height = Abs(Y - YY)
    End If
    Err.Clear
End Sub
```

```
Private Sub Gambar1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
```

```
    On Error Resume Next
    If Button = 1 And Status_Cropping Then
        Gambar1.Line (XX, YY)-(XX2, YY2), &H0, B
        shp1.Visible = False
        Set Picture1.Picture = Nothing
        Picture1.Height = shp1.Height - 1
        Picture1.Width = shp1.Width - 1
    End If
End Sub
```

```

Picture1.Refresh
Picture1.PaintPicture Gambar1.Image, 0, 0, shp1.Width, shp1.Height, shp1.Left + 1,
shp1.Top + 1, shp1.Width - 1, shp1.Height - 1, vbSrcCopy
Call Segmentasi_gb(Picture1, 0, 0, Picture1.ScaleWidth - 1, Picture1.ScaleHeight - 1,
Gambar2)
Gambar3.Width = Gambar2.Width
Gambar3.Height = Gambar2.Height

Call Scalling(Gambar2, Gambar3, 100 / Gambar2.ScaleWidth, 100 /
Gambar2.ScaleHeight)
Status_Cropping = False
Ekstrasi_Ciri
End If
Err.Clear
End Sub

```

```

Private Sub Menu_Click(Index As Integer)
Select Case Index
Case 0: Open_File
Case 1: Save_Data
End Select
End Sub

```

- **FORM PENGUJIAN**

```

Option Explicit
Private Type Pelatihan_Rec
No_Pelatihan As String
No_User As String
Nilai_Output As Double
Nilai_Target As Double
End Type
Dim Pelatihan As Pelatihan_Rec
Private Type BobotBias2Hidden_Rec
No_V0 As String
No_Pelatihan As String
Nilai_V0 As Double
End Type
Dim BobotBias2Hidden1 As BobotBias2Hidden_Rec
Private Type BobotInput2Hidden_Rec
No_V As String
No_Pelatihan As String
Nilai_V As Double
End Type
Dim BobotInput2Hidden1 As BobotInput2Hidden_Rec

Private Type BobotBias2Output_Rec
No_W0 As String
No_Pelatihan As String

```

```

    Nilai_W0 As Double
End Type
Private Type User_Rec
    No_User As String
    Nama As String
End Type
Dim User As User_Rec
Dim XX As Long, YY As Long
Dim XX2 As Long, YY2 As Long
Dim Status_Cropping As Boolean
Dim Status_OpenFile As Boolean
Dim Hasil_Binner(99, 99) As Byte
Dim Hasil_Pic(9, 99) As Double
Dim Hasil_Pic1(9, 9) As Double
Dim No_User As String

Dim BobotBias2Output1 As BobotBias2Output_Rec
Private Type BobotHidden2Output_Rec
    No_W As String
    No_Pelatihan As String
    Nilai_W As Double
End Type
Dim BobotHidden2Output1 As BobotHidden2Output_Rec
Dim Sqlcmd As String
Private Function Find_Pelatihan(No_User As String) As Boolean
    Find_Pelatihan = False
    Sqlcmd = "Select Top 1 * From T_Pelatihan Where No_User=" & No_User & " Order
by No_Pelatihan desc "
    Set MyRs = MyDb.Execute(Sqlcmd)
    If Not (MyRs.BOF And MyRs.EOF) Then
        Find_Pelatihan = True
        With Pelatihan
            .No_Pelatihan = MyRs.Fields(0)
            .No_User = MyRs.Fields(1)
            .Nilai_Output = MyRs.Fields(2)
            .Nilai_Target = MyRs.Fields(3)
        End With
    End If
End Function
Private Function Find_No_User(No_User As String) As Boolean
Dim Sqlcmd As String
    Find_No_User = False
    Sqlcmd = "Select * From T_User Where No_user=" & No_User & " "
    Set MyRs = MyDb.Execute(Sqlcmd)
    If Not (MyRs.BOF And MyRs.EOF) Then
        Find_No_User = True
        With User
            .No_User = MyRs.Fields(0)
            .Nama = MyRs.Fields(1)
        End With
    End If
End Function

```

```

    End With
End If
End Function
Private Function Juml_Target(No_User As String) As Long
    Juml_Target = 0
    Sqlcmd = "Select count(No_Pelatihan) From T_Pelatihan Where No_User= '" &
No_User & "'"
    Set MyRs = MyDb.Execute(Sqlcmd)
    If Not (MyRs.BOF And MyRs.EOF) Then
        Juml_Target = MyRs.Fields(0)
    End If
End Function
Private Sub Get_AllPelatihan(No_User As String)
Dim i As Long
    i = 0
    Sqlcmd = "Select Top 1 * From T_Pelatihan Where No_User= '" & No_User & "'" order
by No_Pelatihan "
    Set MyRs = MyDb.Execute(Sqlcmd)
    While Not (MyRs.BOF And MyRs.EOF)
        Pelatihan.No_Pelatihan = MyRs.Fields(0)
        Y(i) = MyRs.Fields(2)
        Target(i) = MyRs.Fields(3)
        i = i + 1
        Sqlcmd = "Select Top 1 * From T_Pelatihan Where No_User= '" & No_User & "'"
And No_Pelatihan > '" & Pelatihan.No_Pelatihan & "'" order by No_Pelatihan "
        Set MyRs = MyDb.Execute(Sqlcmd)
    Wend
End Sub
Private Function Get_Last_No_Pelatihan(No_User As String) As String
    Get_Last_No_Pelatihan = ""
    Sqlcmd = "Select Top 1 * From T_Pelatihan Where No_User= '" & No_User & "'" order
by No_Pelatihan Desc "
    Set MyRs = MyDb.Execute(Sqlcmd)
    If Not (MyRs.BOF And MyRs.EOF) Then
        Get_Last_No_Pelatihan = MyRs.Fields(0)
    End If
End Function
Private Sub Load_BobotBias2Hidden(No_Pelatihan As String)
    Dim i As Long
    i = 0
    Sqlcmd = "Select Top 1 * From T_BobotBias2Hidden Where No_Pelatihan= '" &
No_Pelatihan & "'" order by No_Bobot_V0 "
    Set MyRs = MyDb.Execute(Sqlcmd)
    While Not (MyRs.BOF And MyRs.EOF)
        BobotBias2Hidden1.No_V0 = MyRs.Fields(0)
        BobotBias2Hidden(i) = MyRs.Fields(2)
        i = i + 1
    End While
End Sub

```

```

    Sqlcmd = "Select Top 1 * From T_BobotBias2Hidden Where No_Pelatihan= "" &
No_Pelatihan & "" And No_Bobot_V0 > "" & BobotBias2Hidden1.No_V0 & "" order by
No_Bobot_V0 "
    Set MyRs = MyDb.Execute(Sqlcmd)
Wend
End Sub
Private Sub Load_BobotInput2Hidden(No_Pelatihan As String)
    Dim i As Long
    i = 0
    Sqlcmd = "Select Top 1 * From T_BobotInput2Hidden Where No_Pelatihan= "" &
No_Pelatihan & "" order by No_Bobot_V "
    Set MyRs = MyDb.Execute(Sqlcmd)
    While Not (MyRs.BOF And MyRs.EOF)
        i = i + 1
        BobotInput2Hidden1.No_V = MyRs.Fields(0)
        If (i Mod 100) = 0 Then
            bobotinput2hidden(100 - 1, (i \ 100) - 1) = MyRs.Fields(2)
        Else
            bobotinput2hidden((i Mod 100) - 1, i \ 100) = MyRs.Fields(2)
        End If
        DoEvents
        Sqlcmd = "Select Top 1 * From T_BobotInput2Hidden Where No_Pelatihan= "" &
No_Pelatihan & "" And No_Bobot_V > "" & BobotInput2Hidden1.No_V & "" order by
No_Bobot_V "
        Set MyRs = MyDb.Execute(Sqlcmd)
    Wend
End Sub
Private Sub Load_BobotHidden2Output(No_Pelatihan As String)
    Dim i As Long
    i = 0
    Sqlcmd = "Select Top 1 * From T_BobotHidden2Output Where No_Pelatihan= "" &
No_Pelatihan & "" order by No_Bobot_W "
    Set MyRs = MyDb.Execute(Sqlcmd)
    While Not (MyRs.BOF And MyRs.EOF)
        BobotHidden2Output1.No_W = MyRs.Fields(0)
        BobotHidden2Output(i) = MyRs.Fields(2)
        i = i + 1
        Sqlcmd = "Select Top 1 * From T_BobotHidden2Output Where No_Pelatihan= "" &
No_Pelatihan & "" And No_Bobot_W > "" & BobotHidden2Output1.No_W & "" order by
No_Bobot_W "
        Set MyRs = MyDb.Execute(Sqlcmd)
    Wend
End Sub
Private Sub Load_BobotBias2Output(No_Pelatihan As String)
    Sqlcmd = "Select Top 1 * From T_BobotBias2Output Where No_Pelatihan= "" &
No_Pelatihan & "" "
    Set MyRs = MyDb.Execute(Sqlcmd)
    If Not (MyRs.BOF And MyRs.EOF) Then
        BobotBias2Output = MyRs.Fields(2)
    End If
End Sub

```

```

End If
End Sub

Private Sub Open_File()
'On Error Resume Next
Dim sFile As String
With Cd1
.DialogTitle = "Open"
.CancelError = False

.Filter = "Bitmap Files (*.bmp)" & "|*.bmp"
.ShowOpen
sFile = .FileName
Status_OpenFile = True
End With
shp1.Visible = False
Gambar1.Picture = LoadPicture(Cd1.FileName)
End Sub
Private Sub Resize_Horisonal()
Dim i As Integer, j As Integer, k As Integer, l As Integer
Dim Temp As String
For j = 0 To 99
l = 0
For i = 0 To 99 Step 10
k = i
Temp = ""
While k < i + 10
Temp = Temp + Trim(Str(Hasil_Binner(k, j)))
k = k + 1
Wend
Hasil_Pic(l, j) = BinToDec(Temp)
l = l + 1
Next i
Next j
End Sub
Private Sub Resize_Vertikal()
Dim i As Integer, j As Integer, k As Integer, l As Integer
Dim Temp As Double
For i = 0 To 9
l = 0
For j = 0 To 99 Step 10
k = j
Temp = 0
While k < j + 10
Temp = Temp + Hasil_Pic(i, k)
k = k + 1
Wend
Hasil_Pic1(i, l) = Temp
l = l + 1

```

```

    Next j
  Next i
End Sub
Private Sub Binnarisasi(Source As PictureBox, NilaiBatas As Byte)
  Dim i As Long
  Dim j As Long
  Dim W As Long, R As Long, G As Long, b As Long, X As Long
  Dim n As Byte
  For j = 0 To Source.ScaleHeight - 1
    For i = 0 To Source.ScaleWidth - 1
      W = Source.Point(i, j)
      R = W And RGB(255, 0, 0)
      G = Int((W And RGB(0, 255, 0)) / 256)
      b = Int(Int((W And RGB(0, 0, 255)) / 256) / 256)
      X = Int((R + G + b) / 3)
      If X < NilaiBatas Then Hasil_Binner(i, j) = 1 Else Hasil_Binner(i, j) = 0
    Next i
  Next j
End Sub
Private Sub Ekstrasi_Ciri()
  Call Binnarisasi(Picture1, 128)
  Dim i As Integer, j As Integer
  Fg0.Cols = 101
  Fg0.Rows = 101
  Fg0.ColWidth(0) = 350
  For i = 1 To 100
    Fg0.TextMatrix(0, i) = Trim(Str(i))
    Fg0.ColWidth(i) = 350
  Next
  For j = 1 To 100
    Fg0.TextMatrix(j, 0) = Trim(Str(j))
  Next
  For j = 1 To 100
    For i = 1 To 100
      Fg0.TextMatrix(j, i) = Trim(Str(Hasil_Binner(i - 1, j - 1)))
    Next i
  Next j
  Call Resize_Horisontal
  Call Resize_Vertikal
  Call Input_Jaringan
End Sub
Private Sub Input_Jaringan()
  Dim i As Integer, j As Integer, k As Integer

  fg3.Cols = 101
  fg3.Rows = 2
  For i = 0 To 99
    fg3.TextMatrix(0, i) = Trim(Str(i + 1))
  Next i

```



```

fg3.TextMatrix(0, 100) = "Target"
k = 0
For j = 1 To 10
  For i = 1 To 10
    fg3.TextMatrix(1, k) = Format(Hasil_Pic1(i - 1, j - 1) / 100, "#0.000000")
    k = k + 1
  Next i
Next j
fg3.TextMatrix(1, 100) = 0
For i = 0 To 99
  fg3.TextMatrix(1, 100) = CDb1(fg3.TextMatrix(1, 100)) + CDb1(fg3.TextMatrix(1, i))
Next i
End Sub
Private Sub Pengujian()
Dim inputA(100) As Double
Dim AoA, BoB, AoB, jarak As Double
Dim i As Integer
Dim Total_Pelatihan As Double, Total_Dikenali As Double
Dim Total_Dikenali_Y As Double

Dim Temp As String
Dim Target1 As Double
  Jum1_HiddenLayer = 50
  NumInput = 100
  No_User = TxtUser(0).Text
  If Find_Pelatihan(No_User) Then
    NumOutput = Jum1_Target(No_User1) + 1
    InitBackpro
    Get_AllPelatihan No_User1
    Pelatihan.No_Pelatihan = Get_Last_No_Pelatihan(No_User)
    Call Load_BobotBias2Hidden(Pelatihan.No_Pelatihan)
    Call Load_BobotInput2Hidden(Pelatihan.No_Pelatihan)
    Call Load_BobotBias2Output(Pelatihan.No_Pelatihan)
    Call Load_BobotHidden2Output(Pelatihan.No_Pelatihan)
    For i = 0 To 99
      inputA(i) = CDb1(fg3.TextMatrix(1, i))
    Next
    Call FeedForward(inputA, NumOutput - 1)

  End If
Dim hasil As Double
Total_Dikenali = 0
Total_Dikenali_Y = 0
Total_Pelatihan = 0
Target1 = Round(CDb1(fg3.TextMatrix(1, 100)), 6)
Sqlcmd = "Select Top 1 * From T_Pelatihan order by No_Pelatihan "
Set MyRs = MyDb.Execute(Sqlcmd)
While Not (MyRs.EOF And MyRs.EOF)
  Temp = MyRs.No_Pelatihan

```

```

Total_Pelatihan = Total_Pelatihan + 1
Pelatihan.Nilai_Output = MyRs!Nilai_Output
Pelatihan.Nilai_Target = MyRs!Nilai_Target
hasil = 0.5 * ((Pelatihan.Nilai_Target - Pelatihan.Nilai_Output) ^ 2 + (Target1 -
Y(NumOutput - 1)) ^ 2)
If (hasil <= Error_Rate) Then
    Total_Dikenali = Total_Dikenali + 1
End If
If Pelatihan.Nilai_Output >= Y(NumOutput - 1) Then
    hasil = Pelatihan.Nilai_Output - Y(NumOutput - 1)
Else
    hasil = Y(NumOutput - 1) - Pelatihan.Nilai_Output
End If
If (hasil <= Error_Rate1) Then
    Total_Dikenali_Y = Total_Dikenali_Y + 1
End If
Sqlcmd = "Select Top 1 * From T_Pelatihan Where No_Pelatihan > " & Temp & "
order by No_Pelatihan "
Set MyRs = MyDb.Execute(Sqlcmd)
Wend
Dim Persentase As Double
Persentase = Round(Total_Dikenali / Total_Pelatihan, 2) * 100
If Persentase >= 90 Then
    ' bandingkan nilai Y nya
    Persentase = Round(Total_Dikenali_Y / Total_Pelatihan, 2) * 100
End If

Label7.Caption = Persentase
If Persentase >= 90 Then
    Label8.Caption = "IDENTIK"
Else
    Label8.Caption = "TIDAK DIKENALI"
End If
End Sub
Private Sub Command1_Click(Index As Integer)
    Select Case Index
    Case 0: Open_File
    Case 1: Pengujian
    Case 2: Setting_Awal
    End Select
End Sub

Private Sub Setting_Awal()
Dim i As Integer, j As Integer

TxtUser(0).Text = ""
TxtUser(1).Text = ""
TxtUser(0).Enabled = True
TxtUser(1).Enabled = False

```

```

Gambar1.Cls
Set Gambar1.Picture = Nothing
Command1(0).Enabled = False
Command1(1).Enabled = False
Fg0.Clear
fg3.Clear
fg3.Cols = 101
fg3.Rows = 2
For i = 0 To 99
    fg3.TextMatrix(0, i) = Trim(Str(i + 1))
Next i
fg3.TextMatrix(0, 100) = "Target"

```

```

Fg0.Cols = 101
Fg0.Rows = 101
Fg0.ColWidth(0) = 350
For i = 1 To 100
    Fg0.TextMatrix(0, i) = Trim(Str(i))
    Fg0.ColWidth(i) = 350
Next
For j = 1 To 100
    Fg0.TextMatrix(j, 0) = Trim(Str(j))
Next
For j = 1 To 100
    For i = 1 To 100
        Fg0.TextMatrix(j, i) = ""
    Next i
Next j

```

End Sub

```

Private Sub Form_Load()
    Setting_Awal
End Sub

```

```

Private Sub Gambar1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    On Error Resume Next
    If Button = 1 Then
        If Status_OpenFile Then
            XX = X
            YY = Y
            XX2 = X
            YY2 = Y
            shp1.Shape = 0
            shp1.Visible = True
            shp1.Left = X
            shp1.Top = Y

```

```

        shp1.Width = 0
        shp1.Height = 0
        Status_Cropping = True
        Status_OpenFile = False
    End If
End If
Err.Clear
End Sub
Private Sub Gambar1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    On Error Resume Next
    If Button = 1 And Status_Cropping Then
        XX2 = X
        YY2 = Y
        shp1.Left = IIf(X > XX, XX, X)
        shp1.Top = IIf(Y > YY, YY, Y)
        shp1.Width = Abs(X - XX)
        shp1.Height = Abs(Y - YY)
    End If
    Err.Clear
End Sub

Private Sub Gambar1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    On Error Resume Next
    If Button = 1 And Status_Cropping Then
        Gambar1.Line (XX, YY)-(XX2, YY2), &H0, B
        shp1.Visible = False
        Set Picture1.Picture = Nothing
        Picture1.Height = shp1.Height - 1
        Picture1.Width = shp1.Width - 1
        Picture1.Refresh
        Picture1.PaintPicture Gambar1.Image, 0, 0, shp1.Width, shp1.Height, shp1.Left + 1,
shp1.Top + 1, shp1.Width - 1, shp1.Height - 1, vbSrcCopy
        Picture1.PaintPicture Gambar1.Image, 0, 0, shp1.Width, shp1.Height, shp1.Left + 1,
shp1.Top + 1, shp1.Width - 1, shp1.Height - 1, vbSrcCopy
        Call Segmentasi_gb(Picture1, 0, 0, Picture1.ScaleWidth - 1, Picture1.ScaleHeight - 1,
Gambar2)
        Call Scalling(Gambar2, Picture1, 100 / Gambar2.ScaleWidth, 100 /
Gambar2.ScaleHeight)
        Status_Cropping = False
        Ekstrasi_Ciri
        Command1(0).Enabled = False
        Command1(1).Enabled = True
    End If
    Err.Clear
End Sub

Private Sub TxtUser_KeyPress(Index As Integer, KeyAscii As Integer)

```

```
If Index = 0 Then
  If KeyAscii = 13 Then
    User.No_User = TxtUser(0).Text
    If Find_No_User(User.No_User) Then
      Command1(0).Enabled = True
      TxtUser(0).Enabled = False
      TxtUser(1).Text = User>Nama
    Else
      MsgBox "User tidak ditemukan !!!", vbCritical
      Setting_Awal
    End If
    KeyAscii = 0
  End If
End If
End Sub
```



DAFTAR PRESTASI AKADEMIK PRAKTIKUM
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA

Nama Mahasiswa	:	ZULAIKHA PRABAWATI
NIM	:	0512696
Tempat, Tanggal Lahir	:	TRENGGALEK, 23 JUNI 1986
Jenjang	:	Strata I (S1)
Fakultas	:	Teknologi Industri
Jurusan / Program Studi	:	Teknik Elektro
Konsentrasi	:	Teknik Komputer dan Informatika

Praktikum Laboratorium	Kode	Nama Praktikum	SKS	Nilai
I	EL-2215	Fisika	1	B
		Rangkaian Listrik		B
		Rangkaian Logika dan Digital		B+
		Dasar Komputer dan Pemrograman		B+
II	EL-4216	Dasar Elektronika	1	B+
		Dasar Sistem Telekomunikasi		B+
		Mikrokontroler		B
		Sistem Pengukuran		A
III	EL-5316	Dasar Sistem Kendali	1	A
		Basis Data		B
		Administrasi Jaringan		B+
IV	EL-6317	Sistem Operasi	1	B
		Pemrograman Internet		C
		Pemrograman Objek		B+
V	EL-7318	Rekayasa Perangkat Lunak Sistem Informasi	1	B
		Peripheral dan Antar Muka		B
		Pemrosesan Sinyal Digital		B
		Multimedia		B
		Pemrograman Jaringan		B+

Malang, _____

Recording
Jurusan Teknik Elektro S1

Puji Handayani



INSTITUT TEKNOLOGI NASIONAL
 Jl. Raya Karanglo, Km. 2
 MALANG

PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama : ZULAIKHA . P
 NIM : 0512696
 Semester : 9
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro S-1
 Konsentrasi : **TEKNIK ELEKTRONIKA**
TEKNIK ENERGI LISTRIK
TEKNIK KOMPUTER DAN INFORMATIKA
 Alamat : JL. PERUM TIRTASANI BLOK A2, NO 2A, KARANGLO

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat **SKRIPSI Tingkat Sarjana**. Untuk melengkapi permohonan tersebut, bersama kami lampirkan persyaratan-persyaratan yang harus dipenuhi.

Adapun persyaratan-persyaratan pengambilan **SKRIPSI** adalah sebagai berikut :

1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
2. Telah lulus dan menyerahkan Laporan Praktek Kerja (.....)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (.....)
4. Telah menempuh mata kuliah ≥ 134 sks dengan IPK ≥ 2 dan tidak ada nilai E (.....)
5. Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenaran data tersebut diatas
 Recording Teknik Elektro

Malang,^{JULI}.....2009
 Pemohon

[Signature]

[Signature]

(.....) b.



LEMBAR PENGAJUAN JUDUL SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika/Teknik Komputer & Informatika*)

1.	Nama Mahasiswa: ZULAIKHA . P	Nim: 0512696
2.	Waktu Pengajuan	Tanggal: 5 Bulan: AGUSTUS Tahun: 2009
3.	Spesifikasi Judul (berilah tanda silang)**)	
	a. Sistem Tenaga Elektrik	e. Elektronika & Komponen
	b. Energi & Konversi Energi	f. Elektronika Digital & Komputer
	c. Tegangan Tinggi & Pengukuran	g. Elektronika Komunikasi
	d. Sistem Kendali Industri	h. lainnya
4.	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen*) <i>I Kong Samudra, ST MT</i>	Ketua Jurusan Ir. F. Yudi Limpraptono, MT NIP. P. 1039500274
5.	Judul yang diajukan mahasiswa:	SISTEM PENGENAL TANDA TANGAN DENGAN JARINGAN SARAF TIPUAN METODE PROPAGASI BALIK
6.	Perubahan judul yang disetujui Dosen sesuai materi bidang ilmu
	Catatan: <i>1. judul dan metode JST</i>	
7.	Persetujuan Judul skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu	Disetujui Dosen 21-008 2009

Perhatian:

1. Formulir pengajuan ini harap dikembalikan kepada jurusan paling lambat satu minggu setelah disetujui kelompok dosen keahlian dengan dilampirkan proposal skripsi beserta persyaratan skripsi sesuai form S-1
2. Keterangan: *) Coret yang tidak perlu
 **) dilingkari a, b, c,atau g sesuai bidang keahlian

Lampiran : 1 (satu) berkas
Pembimbing Skripsi

Kepada : Yth. I Komang Somawirata, ST. MT
Dosen Institut Teknologi Nasional
MALANG

Yang bertanda tangan dibawah ini :

Nama : ZULAIKHA. P
Nim : 05.12.696
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik (~~Elektronika/Energi Listrik~~/Komputer&Informatika)

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing Utama / Pendamping *) , untuk penyusunan Skripsi dengan judul (proposal terlampir) :

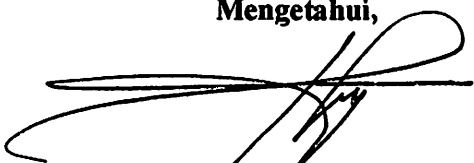
**SISTEM PENGENAL TANDA TANGAN DENGAN JARINGAN
SARAF TIRUAN METODE PROPAGASI BALIK**

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.

Demikian permohonan kami dan atas kesediaan Bapak kami ucapkan terima kasih.

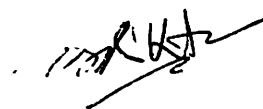
Malang, Agustus 2009

Hormat kami,

Mengetahui,


Ir. F. Yudi Limpraptono, MT

NIP.P. 1030100361



Zulaikha. P

NIM. 05.12.696

*) Coret yang tidak perlu

Form S-3 a

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : ZULAIKHA. P
Nim : 05.12.696
Semester : IX (Sembilan)
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik (~~Elektronika~~ / ~~Energi Listrik~~ / Komputer & Informatika)

Dengan ini menyatakan bersedia / tidak bersedia *) Membimbing Skripsi dari mahasiswa tersebut, dengan judul :

**SISTEM PENGENAL TANDA TANGAN DENGAN JARINGAN SARAF TIRUAN
METODE PROPAGASI BALIK**

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, Agustus 2009

Kami yang membuat pernyataan,



I Komang Somawirata, ST. MT

NIP.Y. 1030100361

Catatan :

Setelah disetujui agar formulir ini diserahkan mahasiswa/i yang bersangkutan kepada Jurusan untuk diproses lebih lanjut.

*) Coret yang tidak perlu

Form S-3 b

**BERITA ACARA SEMINAR PROPOSAL SKRIPSI
 PROGRAM STUDI TEKNIK ELEKTRO S1**

SENTRASI	KOMPUTER DAN INFORMATIKA
-----------------	--------------------------

Nama Mahasiswa	ZULAIKHA PRABAWATI	NIM	05.12.090
Keterangan	Tanggal	Waktu	Tempat / Ruang
Pelaksanaan	24 NOVEMBER 2009		

Spesifikasi Judul (berilah tanda silang *)

a.	Sistem Tenaga Elektrik	e.	Embedded System	i.	Sistem Informasi
b.	Konversi Energi	f.	Antar Muka	j.	Jaringan Komputer
c.	Sistem Kendali	g.	Elektronika Telekomunikasi	k.	Web
d.	Tegangan Tinggi	h.	Elektronika Instrumentasi	l.	Algoritma Cerdas

Judul Proposal yang diseminarkan Mahasiswa	SISTEM PENGEJAL TANDA TANGAN DENGAN JARINGAN SARAF TIRUAN METODE PROPAGASI BALIK
---	--

Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian	
---	--

Catatan :

Catatan :

Keterangan :
 *) diilingkari a, b, c,



INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo 10 KM
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : ZULAIKHA PRABAWATI
NIM : 05.12.696
Masa Bimbingan : 24 Mei 2010 s/d 24 November 2010
Judul Skripsi : SISTEM PENGENAL TANDA TANGAN DENGAN
JARINGAN SARAF TIRUAN METODE PROPAGASI
BALIK

No	Tanggal	Uraian	Paraf Pembimbing
1	26 Juli 2010	1. BAB I : Batasan masalah 2. BAB II : Teori <i>cropping</i>	
2	5 Agustus 2010	1. BAB III : Cara binerisasi ke desimal 2. Tambahkan daftar pustaka	
3	6 Agustus 2010	1. BAB III : Tipe data pada tabel database	
4			
5			

Malang, 2/ Agustus 2010

Dosen Pembimbing

(I KOMANG SOMAWIRATA ST, MT)
NIP. Y.1030100361

FORM S-4b



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

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

BNI (PERSERO) MALANG
BANK NIAGA MALANG

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

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : **ZULAIKHA PRABAWATI**
NIM : **05.12.696**
Jurusan : **T. Elektro S-1**
Konsentrasi : **T. Komputer dan Informatika**
Judul Skripsi : **SISTEM PENGENAL TANDA TANGAN DENGAN
JARINGAN SARAF TIRUAN METODE PROPAGASI
BALIK**

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

Hari : **Senin**
Tanggal : **23 Agustus 2010**
Dengan Nilai : **81,25 (A) 84**

**Panitia Ujian Skripsi
Ketua Majelis Penguji**


Ir. Yusuf Ismail Nakhoda, MT

NIP. Y. 1018800189

Dosen Penguji I


Joseph Dedy Irawan, ST.MT

NIP. 19740416 200501 1 002

Dosen Penguji II


Irmalia Suryani F, ST.MT

NIP. P. 1030100365



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


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

NI (PERSERO) MALANG
BANK NIAGA MALANG

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

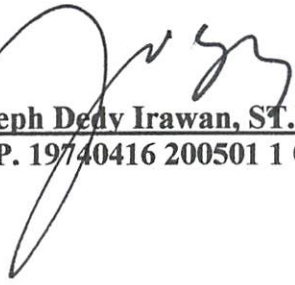
FORMULIR PERBAIKAN SKRIPSI

Nama : **ZULAIKHA PRABAWATI**
Nim : **05.12.696**
Jurusan : **T. Elektro S-1**
Konsentrasi : **T. Komputer dan Informatika**
Masa Bimbingan : **24 Mei s/d 24 November 2010**
Judul Skripsi : **SISTEM PENGENAL TANDA TANGAN DENGAN
JARINGAN SARAF TIRUAN METODE PROPAGASI
BALIK**

Tanggal	Penguji	Uraian	Paraf
23 Agustus 2010	Penguji I	1. Tambahkan progres dari waktu tunggu sehingga user bisa memperkirakan waktu selesainya proses perhitungan	
	Penguji II	1. BAB IV : direvisi 2. BAB V : menyesuaikan	

Disetujui,

Dosen Penguji I


Joseph Dedy Irawan, ST.MT
NIP. 19740416 200501 1 002

Dosen Penguji II


Irmalia Suryani F, ST. MT
NIP. P. 1030100365

Mengetahui,

Dosen Pembimbing


I Komang Somawirata, ST. MT
NIP. Y. 1030100361

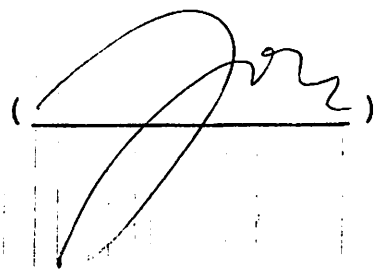
Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : ZULFIKHA P.
NIM : 0512696
Perbaikan melalui :

1) TAMBAH KAN PROGRES DARI
WAKTU TUNGGU SEHINGGA USER
BISA MEMPERILAKKAN WAKTU SELESAINYA
PROSES PERHITUNGAN.

Malang, 23-8-2020





INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

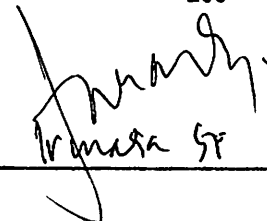
Dalam pelaksanaan Ujian Skripsi Janjang Strata I Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Zulailha Probowati
NIM : 0512696
Perbaikan meliputi :

BAB IV Direvisi
/ Menyempatkan .

Malang,

200


(Irma SR)