

**PENYELESAIAN MODEL REGRESI
MENGUNAKAN PEMBELAJARAN NEUROFUZZY**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika**



**Nama : Rangga Aulia Hakim
NIM : 02 523 257**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2006**

LEMBAR PENGESAHAN PEMBIMBING

**PENYELESAIAN MODEL REGRESI
MENGUNAKAN PEMBELAJARAN NEUROFUZZY**

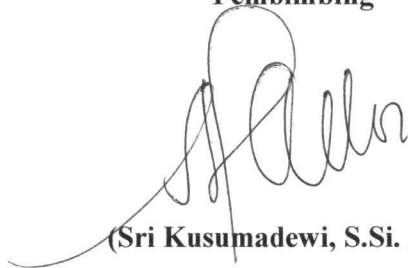
TUGAS AKHIR

Disusun oleh :

**Nama : Ranga Aulia Hakim
NIM : 02 523 257**

Yogyakarta, 3 November 2006

Pembimbing

A handwritten signature in black ink, appearing to read 'Sri Kusumadewi', written over a horizontal line.

(Sri Kusumadewi, S.Si. , MT)

LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR

Yang bertanda tangan di bawah ini

Nama : Rangga Aulia Hakim

NIM : 02 523 257

Tugas Akhir dengan judul :

PENYELESAIAN MODEL REGRESI MENGUNAKAN PEMBELAJARAN NEUROFUZZY

Dengan ini saya menyatakan dengan sesungguhnya bahwa dalam Tugas Akhir ini tidak terdapat keseluruhan tulisan atau karya yang saya ambil dengan menyalin, meniru dalam bentuk rangkaian kalimat atau simbol atau algoritma atau program yang menunjukkan gagasan atau pendapat atau pemikiran orang lain, yang saya aku seolah-olah sebagai tulisan atau karya saya sendiri.

Apabila saya melakukan hal tersebut di atas, baik sengaja atau tidak, dengan ini saya menyatakan menarik Tugas Akhir yang saya ajukan sebagai hasil karya saya sendiri ini. Bila di kemudian hari terbukti bahwa saya melakukan tindakan di atas, gelar dan ijazah yang telah diberikan oleh Universitas Islam Indonesia batal saya terima.

Yogyakarta, 3 November 2006

Yang Membuat Pernyataan



Rangga Aulia Hakim

Saksi-saksi

Sri Kusumadewi, S.Si., MT

Saksi I

Nur Wijyaning R, S.Kom

Saksi II





LEMBAR PENGESAHAN PENGUJI

PENYELESAIAN MODEL REGRESI MENGUNAKAN PEMBELAJARAN NEUROFUZZY

TUGAS AKHIR

Oleh :

Nama : Ranga Aulia Hakim

NIM : 02 523 257

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, 3 November 2006

Tim Penguji

Sri Kusumadewi, S.Si. , MT
Ketua

Nur Wijyaning R, S.Kom
Anggota I

Yudi Prayudi. S.Si. , M.Kom
Anggota II

Mengetahui,

Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia



Yudi Prayudi. S.Si. , M.Kom

HALAMAN PERSEMBAHAN

*Rasa syukur ku haturkan Kehadirat Allah SWT yang maha SATU
atas segala limpahan karunia-Nya, keridhoan-Nya, dan cinta kasih-Nya
yang selalu tercurahkan pada makhluknya.*

*Ayahanda & Ibunda tercinta, yang telah melahirkan ku ke dunia.
Terima kasih atas segala kucuran air mata, doa, didikan, kasih sayang
serta tiap tetes keringat yang selalu tercurahkan untuk ku.
Semoga pahala Allah SWT yang berlipat mereka dapatkan,
karena ketidakmampuan ku untuk membalas
semua kebbaikannya.....*

*Adek-adekku yang ku cinta,
terima kasih atas segala doa dan dukungannya.*

UCAPAN TERIMA KASIH

Sobat2x CV.Expertindo : Galuh (ngeljain apa gal?), Gigih (dah jadi laporannya?), Arif (pinjam buku donk), Rudi (pak ST), Uji (kapan ngelembur lagi ji?), Agung (Ayo ke Toga Mas), Khumaini (alo dub..?), Fahrul (MU menang gak ?). Makasih atas dukungan dan toleransinya, ayo kita ngerjain proyek...

Teman2x Lab di lantai 3 : makasih untuk akses gratis internetnya. Rif, ngenet sampe malam gak hari ini ?, Muzid (thank's cd matlabnya), Edo (ayo ke lab mas Romi), mas Romi (thanks for tips), Firdi (thank's bukunya).

Teman2x KKN Reguler 1 Unit KP-122 : Mahmudc, Mulyanto, Rini, QQ, Dinda, Dian, Karien. Makasih atas motivasinya, kapan lagi ke Jarakan ? Temen2x VOIP '02 UII, semoga Kebersamaan VOIP selalu langgeng selalu amien....

Cah2x kos Ragil Mulya : si Kembar (alo.. ne' thank's komputernya waktu pendadaran and dasi yo..), Firman, Ito', Apink, Radid, Ali, Tunggal Bayu "Rosonerri" (sorry ngerepotin), Kebo (ex), Billy the Kid (thank's celana panjang hitamnya ya..), Budi (ex), dan semua yang tak bisa disebutkan. Makasih atas bantuannya, dan awas listrik *jeglek* lagi (lagi.., lagi... dan lagi ?) he2x....⊗.



MOTTO

"Hasbiyallah wa ni'mal Wakil"

(Cukuplah Allah Menjadi penolongku, dan Allah adalah sebaik-baik pelindung)

"Wahai hamba-hamba-Ku, Aku telah mengharamkan diri Ku dari beerbuat kedzaliman. Aku pun mengharamkan hal tersebut kepadamu. Karena itu, jangan lah kamu saling mendzalimin.

Wahai hamba-hamba-Ku kamu semua dalam keadaan lapar, kecuali yang telah Aku beri makanan; maka mintalah makanan kepadaku, niscaya Aku berikan makanan kepadamu.

Wahai hamba-hamba-Ku, kamu semua dalam keadaan telanjang; kecuali yang telah Aku beri pakaian; maka mintalah pakaian kepada Ku niscaya Aku berikan pakaian kepadamu.

Wahai hamba-hamba-Ku, kamu semua melakukan dosa siang dan malam, sedang Aku mengampuni dosa semuanya; maka mintalah ampunan kepada Ku,

niscaya Aku ampuni dosa kamu.

Wahai hamba-hamba-Ku, kamu tidak akan bisa memberikan Ku mudharat, hingga bisa mencelakai Ku, serta tidak bisa pula memberikan manfaat kepada Ku,

hingga Aku mengambil manfaat darimu.

Wahai hamba-hamba-Ku, jikalau orang pertama dari kamu atau terakhir, manusia, jin; mereka memiliki hati yang sangat jahat kemudian ia meminta Ku, Aku akan memberikan setiap orang apa yang dimintanya. Hal tersebut tidak mengurangi kekuasaan Ku sedikitpun,

kecuali seperti berkurangnya jarum yang dimasukkan kedalam lautan.

Wahai hamba-hamba-Ku, ini semua adalah amalan-amalanmu. Aku menjaganya untukmu, siapa yang mendapatkan (amalnya berupa) kebaikan, hendaklah ia bersyukur kepada Allah.

Barang siapa yang mendapati hal lain selain itu, hendaklah tidak mencela siapapun kecuali dirinya sendiri "

(HR. Tirmidzi, Shahih Muslim, dari Abi Dzar al-Ghifari)

KATA PENGANTAR



Assalamu'alaikum. Wr. Wb

Dengan mengucapkan Alhamdulillah, puji dan syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir, dengan judul "*Penyelesaian Model Regresi Menggunakan Pembelajaran Neurofuzzy*". Shalawat serta salam semoga senantiasa tercurah atas Nabi Muhammad SAW, para kerabat, serta pengikutnya hingga hari kiamat nanti.

Penyusunan tugas akhir merupakan sebagian upaya untuk memenuhi syarat kelulusan studi serta syarat untuk memperoleh gelar sarjana dari Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia Yogyakarta.

Dalam pembuatan tugas akhir ini, penulis mendapat banyak bimbingan dan pengarahan serta bantuan yang sangat bermanfaat dari berbagai pihak. Oleh karena itu dalam kesempatan ini, dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Ayah, Ibu dan Keluarga tercinta, atas kasih sayang dan doanya.
2. Bapak Fathul Wahid, ST., M.Sc. selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia, dan seluruh jajaran dekanat Fakultas Teknologi Industri.

3. Bapak Yudi Prayudi, SSi.,Mkom., selaku Ketua Jurusan Teknik Informatika, Universitas Islam Indonesia.
4. Ibu Sri Kusumadewi, SSi., MT., selaku dosen pembimbing tunggal, yang telah memberikan arahan, saran, dan dorongan semangat kepada penulis sehingga penulis dapat secepatnya menyelesaikan tugas akhir ini.
5. Teman-teman CV.Expertindo yang telah membantu memberikan ide, saran, dan kritik dalam pembuatan tugas akhir ini.
6. Teman-teman VOIP 2002 pada khususnya dan informatika UII pada umumnya.
7. Semua pihak yang telah memberikan bantuan dan dorongan yang tidak bisa penulis sebutkan satu persatu.

Semoga dengan segala bantuan yang telah diberikan kepada penulis, akan mendapat pahala yang setimpal dari ALLAH SWT. Amin

Penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna. Untuk itu segala saran dan kritik yang sifatnya membangun sangat penulis harapkan guna peneympurnaan dimasa mendatang.

Akhir kata, semoga laporan tugas akhir ini dapat bermanfaat bagi kita semua. Amiin.

Yogyakarta, 3 November 2006

Rangga Aulia Hakim

ABSTRAKSI

Model regresi interval dapat diselesaikan dengan pembelajaran *neurofuzzy*. Misalkan diberikan pasangan *input-output* (x_k, d_k) , $k = 1, 2, \dots, p$ dengan $x_k = (x_{k1}, x_{k2}, \dots, x_{kn})$. Suatu model regresi *fuzzy* pada pola ke- k direpresentasikan sebagai:

$$Y(x_k) = A_0 + A_1x_{k1} + \dots + A_nx_{kn}$$

dengan A_i adalah bilangan *fuzzy*. Oleh karena itu, nilai estimasi output $Y(x_k)$ juga merupakan bilangan *fuzzy*. Analisis regresi *fuzzy* dapat disederhanakan menjadi analisis regresi interval dimana model regresi interval nantinya akan dibentuk.

Dalam pembelajaran model regresi menggunakan *neurofuzzy* ini, data input dilatih menggunakan dua jaringan *backpropagation* yang berbeda. Satu jaringan *backpropagation* digunakan untuk melatih data untuk mencari batas atas regresi dan jaringan *backpropagation* lainnya digunakan untuk mencari batas bawah regresi.

Kata kunci : *neurofuzzy, backpropagation, learning rate, momentum, iterasi*

TAKARIR

<i>Backward</i>	Perambatan mundur
<i>Backpropagation</i>	Model jaringan saraf tiruan
<i>Cluster</i>	Kelompok data
<i>Error tolerance</i>	Toleransi kesalahan
<i>Feedforward</i>	Perambatan maju
<i>File</i>	Data
<i>Flowchart</i>	Diagram alir
<i>Fuzzy</i>	Kabur, samar
<i>Hidden layer</i>	Lapisan tersembunyi
<i>Hidden unit</i>	Unit tersembunyi
<i>Input</i>	Masukan sistem
<i>Iterasi</i>	Perulangan
<i>Layer</i>	Lapisan
<i>Learning rate</i>	Laju pembelajaran
<i>Multi layer</i>	Banyak lapisan
<i>Neuron</i>	Sel saraf
<i>Neurofuzzy</i>	Gabungan jaringan saraf tiruan dengan logika fuzzy
<i>Output</i>	Keluaran sistem
<i>Single layer</i>	Satu lapisan
<i>Sigmoid biner/bipolar</i>	Salah satu fungsi aktivasi jaringan saraf
<i>Supervised learning</i>	Pembelajaran terbimbing
<i>Training</i>	Proses pembelajaran
<i>Threshold</i>	Nilai ambang
<i>Unsupervised learning</i>	Pembelajaran tidak terbimbing

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN PEMBIMBING	ii
LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR	iii
LEMBAR PENGESAHAN PENGUJI	iv
PERSEMBAHAN	v
MOTTO.....	vii
KATA PENGANTAR	viii
ABSTRAKSI.....	x
TAKARIR	xi
DAFTAR ISI.....	xii
DAFTAR GAMBAR	xvi
DAFTAR TABEL.....	xviii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	6
2.1 Jaringan Saraf Tiruan	6
2.1.1 Arsitektur Jaringan Saraf Tiruan	8

2.1.2 Fungsi Aktifasi.....	10
2.1.3 Proses Pembelajaran Jaringan Saraf Tiruan.....	13
2.2 Backpropagation	14
2.2.1 Arsitektur Model Backpropagation.....	14
2.2.2 Konsep Jaringan Backpropagation.....	16
2.3 Logika Fuzzy.....	19
2.3.1 Himpunan Fuzzy	20
2.4 Neuro-Fuzzy Regression Model	21
BAB III ANALISIS KEBUTUHAN PERANGKAT LUNAK	25
3.1 Metode Analisis.....	25
3.2 Hasil Analisis	25
3.2.1 Masukan Sistem	25
3.2.2 Keluaran Sistem	26
3.2.3 Kebutuhan Fungsi	27
3.2.4 Kebutuhan Perangkat Keras.....	27
3.2.5 Kebutuhan Perangkat Lunak.....	27
3.2.6 Kebutuhan Antarmuka	28
3.2.7 Kinerja Yang Diharapkan	28
BAB IV PERANCANGAN PERANGKAT LUNAK.....	29
4.1 Metode Perancangan	29
4.2 Hasil Perancangan	29
4.2.1 Perancangan Arsitektur	29
4.2.2 Fase Propagasi Maju	31
4.2.3 Fase Propagasi Mundur.....	32
4.3 Perancangan Diagram Alir Sistem	34
4.4 Perancangan Struktur Data.....	39
4.4.1 Perancangan Struktur Data Input	39

4.4.2 Perancangan Struktur Data Output.....	39
4.5 Perancangan Antarmuka.....	40
4.5.1 Rancang Antar Muka Judul.....	40
4.5.2 Rancang Antar Muka Menu	41
4.5.3 Rancang Antar Muka Keluaran.....	41
4.6 Perancangan File	42
4.6.1 Rancang File Untuk Input Data.....	42
4.6.2 Rancang File Untuk Output Data.....	43
BAB V IMPLEMENTASI PERANGKAT LUNAK.....	44
5.1 Pengantar Implementasi	44
5.2 Batasan Implementasi	44
5.2.1 Pemilihan Bahasa Pemrograman.....	44
5.3 Implementasi Sistem	45
5.3.1 Tampilan Form Judul	45
5.3.2 Tampilan Form Utama	46
5.3.3 Tampilan Form Output.....	49
BAB VI ANALISIS KINERJA PERANGKAT LUNAK	51
6.1 Pengujian Normal.....	52
6.1.1 Proses Input data	52
6.1.2 Proses Pelatihan Data.....	53
6.1.3 Proses Output Pembelajaran	54
6.2 Pengujian Tidak Normal	54
6.2.1 Proses Inisialisasi	55
6.2.2 Proses Pelatihan Data.....	56
6.2.3 Proses Output Pembelajaran	57
6.3 Contoh Kasus	57

BAB VII PENUTUP	77
7.1 Kesimpulan.....	77
7.2 Saran.....	77
DAFTAR PUSTAKA	79



DAFTAR GAMBAR

Gambar 2.1	Susunan saraf manusia.....	6
Gambar 2.2	Jaringan single layer	8
Gambar 2.3	Jaringan multi layer dengan satu hidden layer	9
Gambar 2.4	Fungsi aktivasi jaringan saraf tiruan	11
Gambar 2.5	Fungsi sigmoid biner	12
Gambar 2.6	Fungsi sigmoid bipolar	13
Gambar 2.7	Arsitektur jaringan backpropagation	15
Gambar 2.8	Contoh pemetaan input-output pada logika fuzzy	19
Gambar 4.1	Percanaan arsitektur backpropagation untuk neuro-fuzzy	30
Gambar 4.2	Flowchart pembelajaran neuro-fuzzy	36
Gambar 4.3	Flowchart inialisasi nilai alfa pada BP -	36
Gambar 4.4	Flowchart inialisasi nilai alfa pada BP +	37
Gambar 4.5	Flowchart hasil batas atas dan batas bawah dari pembelajaran neuro-fuzzy	38
Gambar 4.6	Rancang antarmuka judul	40
Gambar 4.7	Rancang antarmuka menu.....	41
Gambar 4.8	Rancang antarmuka keluaran.....	42
Gambar 4.9	Contoh file untuk input data	43
Gambar 5.1	Tampilan form judul.....	45
Gambar 5.2	Tampilan utama	46
Gambar 5.3	Tampilan proses pembelajaran	49
Gambar 5.4	Tampilan form output.....	50
Gambar 6.1	Tampilan form variabel input	52
Gambar 6.2	Tampilan load data matrik input.....	53
Gambar 6.3	Tampilan pembelajaran error jaringan.....	53
Gambar 6.4	Tampilan output pembelajaran neuro-fuzzy	54
Gambar 6.5	Tampilan kesalahan varabel input	55

Gambar 6.6	Tampilan kesalahan load data matrik	55
Gambar 6.7	Tampilan kesalahan matrik target.....	56
Gambar 6.8	Tampilan kesalahan pembelajaran jaringan backpropagation.....	56
Gambar 6.9	Tampilan kesalahan output pembelajaran	57
Gambar 6.10	Tampilan output dengan $lr=0.25$ dan $mom=0,9$	59
Gambar 6.11	Tampilan output dengan $lr=0.25$ dan $mom=0,5$	62
Gambar 6.12	Tampilan output dengan $lr=0.50$ dan $mom=0,9$	64
Gambar 6.13	Tampilan output dengan $lr=0,50$ dan $mom=0,5$	66
Gambar 6.14	Tampilan output dengan $lr=0.75$ dan $mom=0,9$	68
Gambar 6.15	Tampilan output dengan $lr=0.75$ dan $mom=0,5$	70
Gambar 6.16	Tampilan output dengan $lr=1$ dan $mom=0,9$	72
Gambar 6.17	Tampilan output dengan $lr=1$ dan $mom=0,5$	74

DAFTAR TABEL

Tabel 6.1. Pasangan input-output regresi	57
Tabel 6.2. Tabel hasil pembelajaran Neurofuzzy dengan learning rate = 0.25 dan momentum = 0,9	60
Tabel 6.3. Tabel hasil pembelajaran Neurofuzzy dengan learning rate = 0.25 dan momentum = 0,5	62
Tabel 6.4. Tabel hasil pembelajaran Neurofuzzy dengan learning rate = 0.50 dan momentum = 0,9	64
Tabel 6.5. Tabel hasil pembelajaran Neurofuzzy dengan learning rate = 0,50 dan momentum = 0,5	67
Tabel 6.6. Tabel hasil pembelajaran Neurofuzzy dengan learning rate = 0.75 dan momentum = 0,9	69
Tabel 6.7. Tabel hasil pembelajaran Neurofuzzy dengan learning rate = 0.75 dan momentum = 0,5	71
Tabel 6.8. Tabel hasil pembelajaran Neurofuzzy dengan learning rate = 1 dan momentum = 0,9	73
Tabel 6.9. Tabel hasil pembelajaran Neurofuzzy dengan learning rate = 1 dan momentum = 0,5	75

BAB I

PENDAHULUAN



1.1 Latar Belakang

Kondisi teknologi komputer saat sekarang yang sudah semakin maju dan berkembang dengan cepat, mengakibatkan berbagai dampak bagi manusia baik itu positif atau negatif. Sehingga bukan mustahil lagi pekerjaan yang biasanya dilakukan oleh tangan manusia dapat dikerjakan oleh sebuah mesin komputer dengan hasil yang sama atau lebih baik. Keunggulan lain dari mesin itu yaitu dapat mengerjakan perintah yang berulang-ulang serta perhitungan yang rumit dengan cepat dan memiliki ketepatan yang tinggi. Jika saja persoalan ini diberikan kepada manusia, tentu saja akan timbul rasa bosan, stres, dan lain sebagainya yang nantinya akan berdampak dengan hasil yang salah karena banyaknya kesalahan yang dilakukan.

Dengan kelebihan komputer yang seperti itu, banyak para ahli dan pakar yang memanfaatkan teknologi tersebut dalam perhitungan matematika. Misalnya saja dengan menggunakan metode jaringan saraf tiruan yang memiliki konsep yang hampir sama dengan jaringan saraf otak manusia. Pada jaringan saraf data dilatih, sehingga diharapkan dapat berfikir dan mengambil keputusan berdasarkan data-data input yang telah dipelajari polanya.

Pada penelitian ini akan diselesaikan permasalahan model persamaan regresi dengan menggunakan metode *neurofuzzy*, yaitu dengan model regresi interval. Misalkan diberikan pasangan *input - output* (x_k, d_k), $k = 1, 2, \dots, p$ dengan

$x_k = (x_{k1}, x_{k2}, \dots, x_{kn})$. Suatu model regresi *fuzzy* pada pola ke-k direpresentasikan sebagai:

$$Y(x_k) = A_0 + A_1x_{k1} + \dots + A_nx_{kn}$$

dengan A_i adalah bilangan *fuzzy*. Oleh karena itu, nilai estimasi *output* $Y(x_k)$ juga merupakan bilangan *fuzzy*. Analisis regresi *fuzzy* dapat disederhanakan menjadi analisis regresi interval dimana model regresi interval nantinya akan dibentuk.

Konsep dasar dari analisis regresi interval yang didasarkan pada jaringan *backpropagation*, diperkenalkan oleh Ishibuchi dan Tanaka (1992). Model tersebut menggunakan 2 jaringan *backpropagation*. Satu jaringan digunakan untuk batas atas interval, sedangkan satu jaringan lainnya digunakan untuk batas bawah interval data. Kedua jaringan tersebut dilatih secara terpisah. Setelah dilatih secara terpisah, mis: $g^+(x)$ adalah hasil dari pembelajaran batas atas dan $g^-(x)$ adalah hasil dari pembelajaran batas bawah. Karena dipelajari dari jaringan yang berbeda, memungkinkan terdapat $g^-(x) > g^+(x)$. Oleh karena itu dilakukan penyesuaian terhadap hasil dari pelatihan jaringan tersebut, sehingga $g^+(x) > g^-(x)$.

1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas pada laporan tugas akhir ini adalah, bagaimana mencari rentang nilai regresi, yang terdiri dari batas atas dan batas bawah dari sebuah model regresi menggunakan metode *neurofuzzy*.

1.3 Batasan Masalah

Batasan masalah dalam tugas akhir ini, adalah:

1. Program pelatihan dibuat menggunakan Matlab 7
2. Jaringan yang dibangun terdiri dari tiga lapisan, yaitu: satu lapisan *input*, satu lapisan tersembunyi, dan satu lapisan *output*
3. Lapisan *output* hanya terdiri dari 1 unit saja.
4. *Input* program berupa data yang terdiri dari *input* model regresi dan *output* yang akan dicari rentang intervalnya
5. *Output* program berupa *file* data dan grafik yang menggambarkan batas atas dan bawah serta *output* model regresi,

1.4 Tujuan Penelitian

Penelitian tugas akhir ini memiliki beberapa tujuan, yaitu :

1. Melakukan studi dan pengembangan terhadap jaringan saraf tiruan dalam menyelesaikan permasalahan regresi
2. Menguji kemampuan *neurofuzzy* dalam menyelesaikan permasalahan model regresi
3. Membuat program yang dapat mencari rentang interval regresi dari sebuah model regresi

1.5 Manfaat Penelitian

Manfaat dari penelitian tugas akhir ini antara lain :

1. Menambah perbendaharaan perangkat lunak, terutama di bidang sistem cerdas
2. Program yang dibuat dapat membantu kerja seseorang, dalam menyelesaikan rentang interval regresi

1.6 Sistematika Penulisan

Dalam sistematika penelitian tugas akhir ini diberikan uraian bab demi bab untuk mempermudah pemahaman. Penulisan laporan tugas akhir ini, disusun dalam sistematika yang terbagi menjadi tujuh bab dengan rincian sebagai berikut :

BAB I PENDAHULUAN

Berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Berisi tentang teori-teori yang berhubungan dengan penelitian, antara lain : teori tentang jaringan saraf tiruan, teori tentang jaringan *backpropagation*, teori tentang pembelajaran *fuzzy* dan teori mengenai pembelajaran *neurofuzzy*.

BAB III ANALISIS KEBUTUHAN PERANGKAT LUNAK

Berisi tentang analisis perangkat lunak yang dibangun menggunakan metode terstruktur menggunakan *flowchart* (bagan alir).

BAB IV PERANCANGAN PERANGKAT LUNAK

Berisi tentang aliran proses atau alur dari sistem yang akan dibuat serta *flowchart* dari program serta pembahasannya.

BAB V IMPLEMENTASI PERANGKAT LUNAK

Berisi pembahasan implementasi secara umum dari perancangan perangkat lunak, alasan pemilihan bahasa pemrograman, tahapan pembuatan perangkat lunak, implementasi antar muka serta prosedur dari algoritmanya.

BAB VI ANALISIS KINERJA PERANGKAT LUNAK

Berisi tentang hasil pengujian perangkat lunak yang dibandingkan kebenaran dan kesesuaiannya dengan kebutuhan perangkat lunak.

BAB VII PENUTUP

Berisi kesimpulan dari sistem yang dirancang serta saran pengembangan dari keseluruhan tahapan pembuatan Tugas Akhir ini maupun pembuatan perangkat lunaknya.

BAB II

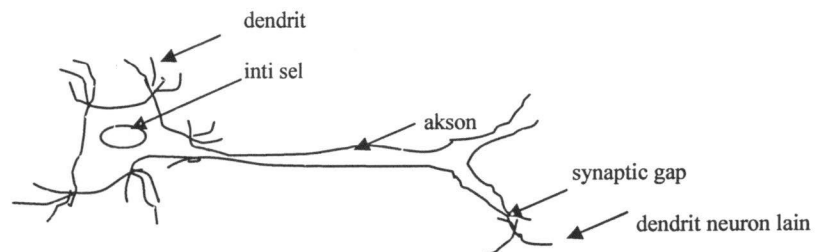
LANDASAN TEORI



2.1 Jaringan Saraf Tiruan

Jaringan saraf tiruan merupakan salah satu model sistem kecerdasan buatan yang merupakan representasi buatan dari sistem otak manusia dan layaknya berfungsi seperti mekanisme kerja dari otak dan saraf biologis pada diri manusia yang selalu mencoba untuk menstimulasikan proses pembelajaran pada otak manusia tersebut. [KUS03]

Dimana dalam jaringan saraf manusia terdiri atas sel-sel yang disebut *neuron* yang dikelompokkan menjadi tiga, yaitu: *dendrit*, *soma* dan *akson*. Dimana ketiga komponen tersebut memiliki fungsi dan tugasnya masing-masing. *Dendrit* memiliki fungsi sebagai bagian dari *neuron* yang menerima sinyal rangsangan dari *neuron* lain atau dari sumber rangsangan. yang berupa *impuls* listrik. Sinyal rangsangan tersebut akan diolah terlebih dahulu di dalam badan sel (*soma*), yaitu dengan menjumlahkannya lalu akan ditransmisikan melalui *synaptic gap* melalui proses kimia. Gambar 2.1 berikut adalah representasi dari *neuron* (sel saraf) manusia.



Gambar 2.1 Susunan saraf manusia

Jaringan saraf tiruan yang merupakan representasi buatan dari sel saraf biologis manusia, juga memiliki bagian-bagian yang berfungsi layaknya bagian-bagian *neuron* dalam sel saraf biologis. Pada jaringan saraf tiruan istilah *neuron* sering disebut sebagai unit, *node*, atau sel. Tiap-tiap *neuron* dalam jaringan saraf tiruan terhubung antara satu dengan lainnya melalui *layer-layer* (lapisan) dengan bobot tertentu. Pada jaringan saraf manusia, bobot tersebut dapat dianalogikan dengan aksi dari proses kimia yang terjadi di dalam *synaptic gap*.

Bobot merupakan suatu nilai yang menghubungkan sebuah sel dengan sel lainnya dalam jaringan. Nilai bobot ini menentukan kuat lemahnya hubungan antar sel tersebut serta menentukan tipe interaksi antar dua sel. Bobot yang bernilai positif dalam jaringan saraf tiruan disebut dengan *excitation*, sedangkan bobot yang bernilai negatif disebut *inhibition*. Nilai suatu bobot dikatakan baik jika memberikan keluaran yang sesuai, dalam arti mendekati keluaran (*output*) yang diharapkan dari suatu masukan (*input*) yang diberikan.

Sinyal rangsangan (*input*) yang datang melalui *dendrit*, akan dijumlahkan dan dikirimkan ke *dendrit neuron* lain yang saling bersentuhan melalui *akson*. Sinyal ini akan diterima *neuron* lain jika memenuhi batasan tertentu, yang disebut dengan nilai ambang (*threshold*) dan *neuron* tersebut dikatakan telah teraktivasi. Akan tetapi pada suatu saat hanya ada satu sinyal yang dapat dikeluarkan walaupun sinyal tersebut ditransmisikan pada beberapa *neuron* lain.

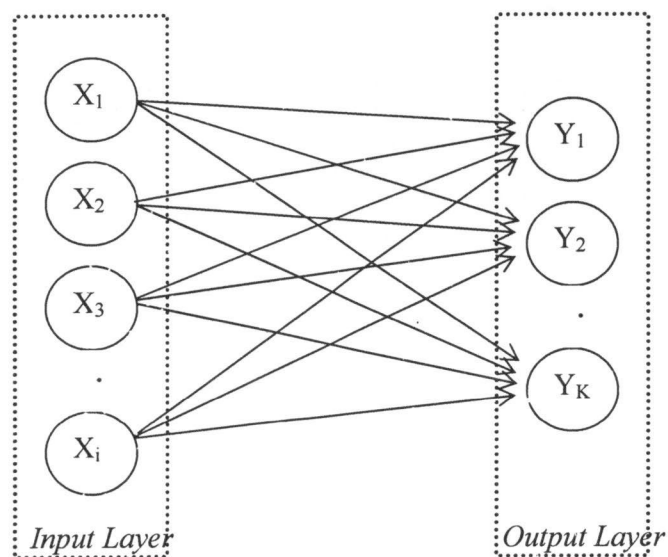
2.1.1 Arsitektur Jaringan Saraf Tiruan

Telah diketahui bahwa *neuron-neuron* dalam jaringan saraf tiruan terkumpul atau dikelompokkan didalam *layer-layer* (lapisan). Kelompok *neuron* yang berada pada lapisan yang sama biasanya memiliki keterhubungan dan karakteristik yang sama. Faktor terpenting dalam menentukan kelakuan dari *neuron* adalah fungsi aktivasi dan pola bobotnya. [KUS03]

Berdasarkan arsitekturnya, jaringan saraf tiruan dibedakan menjadi dua macam, yaitu *single layer* dan *multilayer*.

a) Jaringan *single layer*.

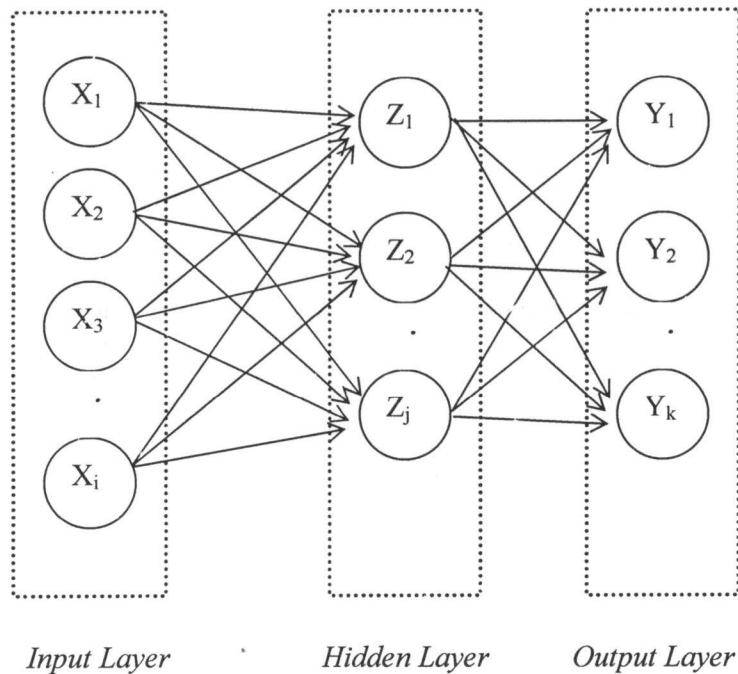
Pada model jaringan ini hanya memiliki satu lapisan *input*, dan satu lapisan *output*. Pemrosesannya langsung dengan mengolah data *input* lalu hasilnya langsung dikirimkan ke *output* tanpa melalui lapisan tersembunyi. Gambar 2.2 merupakan contoh dari jaringan saraf tiruan dengan *single layer*.



Gambar 2.2 Jaringan single layer

b) Jaringan *multilayer*.

Model ini memiliki satu atau lebih lapisan yang terletak antara lapisan *input* dengan lapisan *output*, yang disebut dengan lapisan tersembunyi (*hidden layer*). Gambar 2.3 merupakan contoh jaringan model *multilayer*.



Gambar 2.3 Jaringan multilayer dengan satu hidden layer

Lapisan input (*Input Layer*) berhubungan langsung dengan data masukan kedalam jaringan, sehingga jumlah sel pada lapisan ini besarnya sesuai dengan jumlah data masukannya.

Lapisan keluaran (*Output Layer*) digunakan untuk memberikan hasil akhir, sehingga jumlah sel pada lapisan ini biasanya sesuai dengan jumlah kombinasi keluaran ideal (*target output*) yang diinginkan.

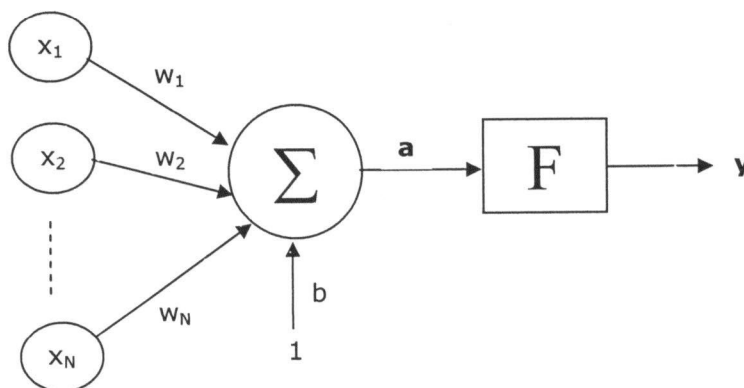
Lapisan tersembunyi (*hidden Layer*) tidak langsung berinteraksi dengan dunia luar, melainkan berhubungan dengan keluaran dari lapisan masukan dan memberikan masukan kepada lapisan keluaran. Jumlah lapisan tersembunyi bisa lebih dari satu lapisan. Banyaknya lapisan tersembunyi ini tergantung dari kompleksitas permasalahan yang akan dipecahkan. Biasanya jumlah lapisan ini berkisar antara jumlah lapisan

Ada berbagai macam hubungan antar lapisan antara lain:

- a. sel dalam satu lapisan saling berhubungan
- b. sel pada lapisan yang satu terhubung dengan sel pada lapisan berikutnya
- c. sel dari lapisan keluaran mempunyai hubungan dengan lapisan masukannya

2.1.2 Fungsi Aktifasi

Fungsi aktivasi merupakan fungsi yang dibangkitkan untuk mengaktifkan *neuron* bila telah melewati suatu nilai ambang tertentu (*threshold*), dengan cara menjumlahkan nilai bobot-bobot yang masuk. Gambar 2.4 merupakan struktur fungsi aktivasi dari jaringan saraf tiruan.



Gambar 2.4 Fungsi aktivasi jaringan saraf tiruan

Pada gambar 2.4 tersebut sebuah *neuron* akan mengolah N *input* (x_1, x_2, \dots, x_n) yang masing-masing memiliki bobot w_1, w_2, \dots, w_n dan bobot bias b , dengan rumus :

$$a = \sum_{i=1}^N x_i w_i \quad (2.1)$$

Kemudian fungsi aktivasi F akan mengaktivasi a menjadi *output* jaringan y .

Ada beberapa fungsi aktivasi yang digunakan dalam membangun jaringan saraf tiruan, yaitu :

- a. Fungsi *linier* atau identitas
- b. Fungsi *step biner*
- c. Fungsi *bipolar*
- d. Fungsi *saturating linier*
- e. Fungsi *symmetric saturating linier*
- f. Fungsi *sigmoid biner*
- g. Fungsi *sigmoid bipolar*

Fungsi aktivasi yang sering digunakan adalah fungsi *sigmoid*, baik itu *sigmoid linier* ataupun *bipolar*. Berikut adalah penjelasan fungsi *sigmoid*.

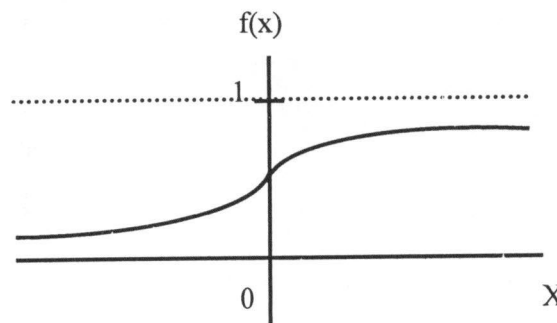
1. Sigmoid Biner

Fungsi yang berbentuk kurva S ini biasanya digunakan untuk melatih jaringan saraf tiruan dengan metode *backpropagation*. Dimana fungsi ini memiliki *range* nilai antara 1 sampai 0 dengan persamaan, sbb:

$$y = f(x) = \frac{1}{1 + e^{-\alpha x}} \quad (2.2)$$

$$\text{dengan, } f'(x) = \sigma f(x)[1 - f(x)] \quad (2.3)$$

Gambar 2.5 adalah fungsi dari *sigmoid biner*.



Gambar 2.5 Fungsi sigmoid biner

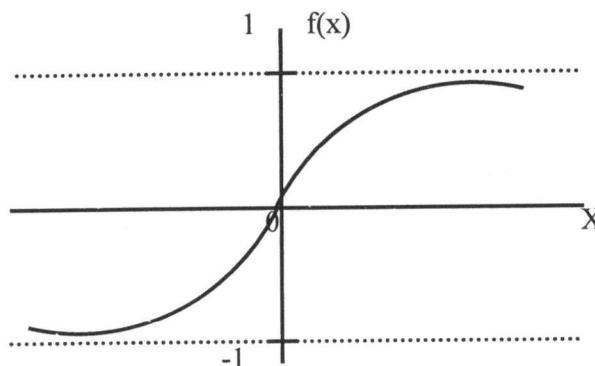
2. Sigmoid Bipolar

Hampir sama dengan fungsi *sigmoid biner*, perbedaannya pada *range* nilai fungsi ini antara -1 sampai 1. dengan persamaan, sbb:

$$\begin{aligned} g(x) &= 2f(x) - 1 = \frac{2}{1 + e^{-\alpha x}} - 1 \\ &= \frac{1 - e^{-\alpha x}}{1 + e^{-\alpha x}} \end{aligned} \quad (2.4)$$

$$\text{dengan, } g'(x) = \frac{\sigma}{2} [1 + g(x)][1 - g(x)] \quad (2.5)$$

Gambar 2.6 adalah fungsi dari *sigmoid bipolar*.



Gambar 2.6 Fungsi sigmoid bipolar

2.1.3 Proses Pembelajaran Jaringan Saraf Tiruan

Keistimewaan dari jaringan saraf tiruan adalah kemampuannya dalam melakukan pembelajaran dari *input* yang diberikan. Agar menghasilkan keluaran yang diinginkan, maka jaringan harus diberi pembelajaran terhadap bobot-bobot yang telah ditentukan pada tiap *layer* (lapisan). Ada dua jenis pembelajaran dalam jaringan saraf tiruan, yaitu : [ERW04]

1. *Supervised learning* (pembelajaran terawasi)

Pada model ini, seolah-olah jaringan diberikan seorang “guru” yang membimbing proses pembelajarannya. Pembelajaran dilakukan terhadap pola pasangan *input-output* yang telah ditentukan oleh pemakai (ahli), dan jaringan diharapkan dapat mempelajari dari pola tersebut. Sehingga jika diberikan *input* lain, maka akan dihasilkan *output* yang sesuai dengan harapan. Beberapa model jaringan yang menggunakan metode pembelajaran terawasi, antara lain: *Delta Rule*, *Backpropagation* dan *Counterpropagation*.

2. *Unsupervised learning* (pembelajaran tak terawasi)

Model ini kebalikan dari model terawasi di atas, disini jaringan hanya diberikan data *input* tanpa memberikan pasangan *outputnya*. Jaringan hanya memodifikasi bobot-bobot data *input* sedemikian rupa, sehingga jika diberikan *input* yang hampir sama akan dihasilkan *output* yang sama pula. Hasil dari model ini adalah berupa *cluster* yaitu pengelompokan dari data *input*.

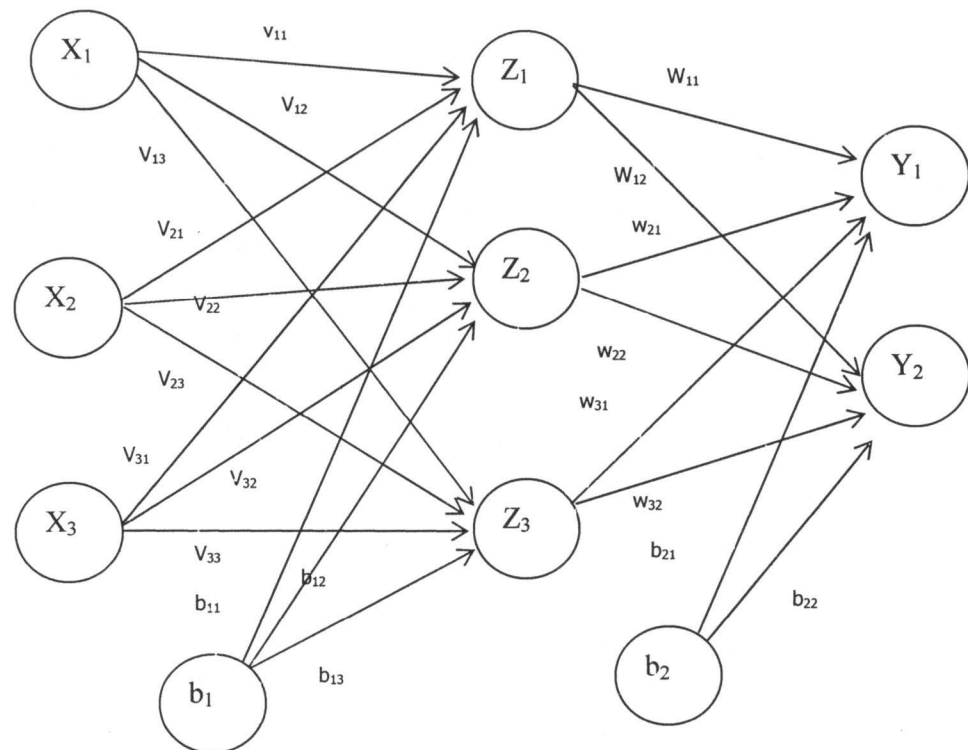
2.2 Backpropagation

Backpropagation merupakan salah satu model jaringan saraf tiruan yang menggunakan metode pembelajaran terawasi (*supervised learning*). Model jaringan ini sangat banyak digunakan, karena memiliki unjuk kerja yang baik dari tingkat ketelitiannya serta kemudahan dalam melakukan pembelajaran.

Backpropagation merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan *neuron-neuron* yang ada pada lapisan tersembunyinya. 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. [KUS04]

2.2.1 Arsitektur Model Backpropagation

Model jaringan *backpropagation* adalah *multilayer* dengan minimal memiliki satu lapisan tersembunyi. Arsitektur dari model *backpropagation* merupakan kombinasi dari sekumpulan sel atau elemen yang tersusun dalam bentuk lapisan serta membentuk suatu jaringan. Gambar 2.7 merupakan contoh dari arsitektur jaringan *backpropagation* dengan satu lapisan tersembunyi.



Gambar 2.7 Arsitektur jaringan backpropagation

Gambar diatas adalah model *backpropagation* dengan satu lapisan *input* dengan tiga unit (x_1, x_2, x_3), satu lapisan tersembunyi dengan tiga unit (z_1, z_2, z_3), satu lapisan *output* dengan dua unit (y_1, y_2) serta dua buah *indeks bias* masing-masing satu pada lapisan tersembunyi (b_1) dan satu lagi pada lapisan output (b_2). V adalah bobot dari unit *input* ke unit tersembunyi, w adalah bobot dari unit

tersembunyi ke unit *output* sedangkan b_1 adalah bobot *bias* ke unit tersembunyi dan b_2 adalah bobot *bias* ke unit *output*.

2.2.2 Konsep Jaringan Backpropagation

Pada intinya, proses pelatihan dengan metode *backpropagation* terdiri dari tiga langkah, yaitu:

1. Data-data dimasukkan kedalam lapisan *input*
2. Perhitungan dan propagasi balik dari *error* yang bersangkutan
3. Pembaharuan bobot dan *bias*.

Saat umpan maju (*feedforward*), setiap unit *input* (x_i) menerima sinyal *input* yang, kemudian sinyal *input* tersebut dialirkan kedalam jaringan menuju lapisan tersembunyi (z_j). Didalam lapisan tersembunyi, sinyal tersebut dihitung nilai aktivasinya lalu mengirimkannya ke lapisan *output*. Kemudian sinyal tersebut oleh unit *output* (y_k) dihitung lagi aktivasinya untuk menghasilkan respon terhadap *input* yang diberikan dalam jaringan.

Saat proses pelatihan setiap unit *output* membandingkan aktivasinya (y_k) dengan nilai target untuk menentukan besarnya *error*. Berdasarkan *error* tersebut, dihitung faktor δ_k yang nantinya akan digunakan untuk mendistribusikan *error* dari *output* ke lapisan sebelumnya.

Algoritma dari backpropagation : [KUS03]

1. Inisialisasi bobot-bobot awal (bilangan acak yang cukup kecil)
2. Kerjakan langkah berikut, selama kondisi berhenti masih FALSE :

Feedforward :

- a. Tiap-tiap unit *input* (x_1, x_2, \dots, x_n) menerima sinyal x_i dan meneruskannya ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).
- b. Tiap-tiap unit tersembunyi (z_1, z_2, \dots, z_p) menjumlahkan sinyal-sinyal *input*:

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.6)$$

gunakan fungsi *aktivasi* untuk menghitung sinyal *outputnya* :

$$z_j = f(z_in_j) \quad (2.7)$$

lalu kirimkan sinyal tersebut ke semua unit pada lapisan atasnya (lapisan *output*)

- c. Tiap-tiap unit *output* (y_1, y_2, \dots, y_k) menjumlahkan sinyal-sinyal *input* terbobot :

$$Y_in_k = w_{0k} + \sum_{i=1}^p z_i w_{jk} \quad (2.8)$$

gunakan fungsi *aktifasi* untuk menghitung sinyal *outputnya* :

$$y_k = f(y_in_k) \quad (2.9)$$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit *output*).

Backpropagation :

- d. Tiap-tiap unit *output* (y_1, y_2, \dots, y_k) menerima target pola yang berhubungan dengan pola input pembelajaran, hitung informasi *errornya*:

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

kemudian hitung 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* (yang nantinya akan digunakan untuk memperbaiki nilai w_{0k}) :

$$\Delta w_{0k} = \alpha \delta_k \quad (2.12)$$

kiriman δ_k ini ke unit-unit yang ada di lapisan dibawahnya

- e. Tiap-tiap unit tersembunyi (z_1, z_2, \dots, z_p) menjumlahkan *delta inputnya* (dari unit-unit yang berada diatasnya) :

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

kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error* :

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

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai v_{ij}) :

$$\Delta v_{jk} = \alpha \delta_j x_i \quad (2.15)$$

hitung juga koreksi *bias* (yang nantinya akan digunakan untuk memperbaiki nilai v_{0j}) :

$$\Delta v_{0k} = \alpha \delta_j \quad (2.16)$$

- f. Tiap-tiap unit *output* (y_1, y_2, \dots, y_m) memperbaiki *bias* dan bobotnya :

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

Tiap-tiap unit tersembunyi (z_1, z_2, \dots, z_p) memperbaiki *bias* dan bobotnya :

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

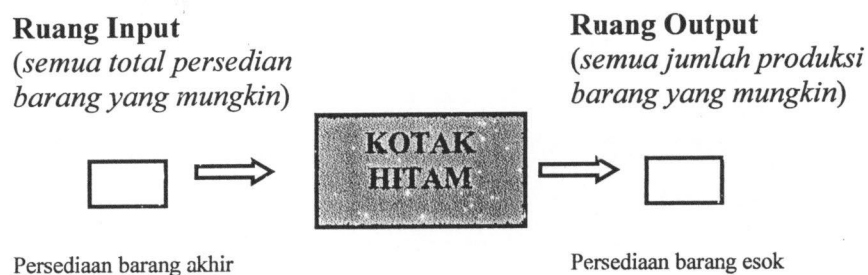
3. Tes kondisi berhenti

Kondisi dikatakan berhenti pada *backpropagation*, jika:

- Iterasi* telah mencapai batas *iterasi* maksimum yang ditentukan oleh *user*.
- Error* yang terjadi telah mencapai batas *error* yang ditentukan oleh *user* sebelumnya.

2.3 Logika Fuzzy

Logika *fuzzy* merupakan suatu cara yang tepat untuk memetakan suatu ruang *input* kedalam suatu ruang *output* [KUS03]. Gambar 2.8 merupakan salah satu contoh gambar dari pemetaan suatu ruang *input* ke *output*.



Gambar 2.8 Contoh pemetaan input-output pada logika fuzzy

Beberapa alasan mengapa menggunakan logika *fuzzy*, antara lain : [KUS02]

- Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah di mengerti.
- Logika *fuzzy* sangat fleksibel.
- Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.

4. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
5. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
6. Logika *fuzzy* didasarkan pada bahasa alami.
7. Logika *fuzzy* mampu memodelkan fungsi-fungsi nonlinier yang sangat kompleks.

2.3.1 Himpunan Fuzzy

Himpunan *fuzzy* tidak sama dengan himpunan *crisp* (tegas), dimana nilai keanggotaannya diperkenalkan oleh Zadeh (1965). Sebuah himpunan *crisp* A didefinisikan oleh item-item yang ada pada himpunan A itu sendiri. Dimana anggota himpunan A itu adalah pasti, misalkan $a \in A$ maka nilai keanggotaan a adalah 1, akan tetapi jika $a \notin A$ maka nilai keanggotaannya menjadi 0. Berbeda dengan nilai keanggotaan dari himpunan *fuzzy* yang didasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sedemikian hingga fungsi tersebut akan mencakup bilangan real pada interval $[0,1]$. Nilai keanggotaannya menunjukkan bahwa suatu item dalam semesta pembicaraan tidak hanya berada dalam rentang 0 dan 1, namun juga nilai yang terletak diantaranya. [KUS02].

Himpunan *fuzzy* memiliki dua variabel, yaitu :

1. Variabel *linguistik* (contoh : MUDA, PAROBAYA, TUA)
2. Variabel *numeris* (contoh : 10, 25, 45, 60)

2.4 Neuro-Fuzzy Regression Model

Penggabungan dari *neuro-fuzzy* atau *fuzzy-neuro* memberikan keuntungan dan keunggulan yang didapat dari sistem jaringan saraf dan logika *fuzzy*. Dimana sistem jaringan saraf mempunyai kelebihan dalam konektivitas strukturnya (kemampuan untuk mengurangi kesalahan) dan kemampuan mempelajari pola masukan, sedangkan logika *fuzzy* memberikan keuntungan *frame work* dengan penalaran *fuzzy IF-THEN* kepada jaringan saraf tiruan.

Keunggulan dari sistem *neurofuzzy* ini, yang coba digunakan untuk menyelesaikan masalah model regresi dengan mencari rentang interval dari persamaan :

$$Y(x_k) = A_0 + A_1x_{k1} + \dots + A_nx_{kn} \quad (2.19)$$

Konsep dasar dari analisis regresi interval yang didasarkan pada jaringan *backpropagation*, diperkenalkan oleh Ishibuchi dan Tanaka (1992). Model tersebut menggunakan dua jaringan *backpropagation*. Satu jaringan digunakan untuk menghitung batas atas interval, sedangkan jaringan lainnya digunakan untuk menghitung batas bawah interval. Dimana kedua jaringan *backpropagation* tersebut dilatih secara terpisah. [LIN96]

Kedua jaringan *backpropagation* tersebut memiliki n buah *neuron* pada lapisan *input* dan satu buah *neuron* pada lapisan *output*. Misalkan $g^+(x)$ adalah hasil dari jaringan *backpropagation* untuk interval batas atas (BPN⁺) dan $g^-(x)$ adalah hasil dari jaringan *backpropagation* untuk interval batas bawah (BPN⁻). Proses pembelajaran dilakukan kepada dua jaringan (BPN⁺ dan BPN⁻) untuk mendapatkan *output* jaringan, sehingga :

$$g^-(x_k) \leq d_k \leq g^+(x_k), k=1,2,\dots,p \quad (2.20)$$

Proses pembelajaran pada BPN⁺ memiliki fungsi biaya untuk menghitung *error* jaringan adalah :

$$E = \sum_{k=1}^p E_k = \frac{1}{2} \sum_{k=1}^p \alpha_k [d_k - g^+(x_k)]^2 \quad (2.21)$$

dengan α_k , memiliki nilai sebagai berikut :

$$\alpha_k = \begin{cases} 1; & d_k > g^+(x_k) \\ \alpha; & d_k \leq g^+(x_k) \end{cases} \quad (2.22)$$

dengan α adalah bilangan positif yang kecil dengan rentang (0 sampai 1).

Atau dapat pula dicari dengan persamaan berikut :

$$\alpha(t) = \frac{1}{\left[1 + \left(\frac{t}{2000}\right)\right]^3} \quad (2.23)$$

dengan t adalah iterasi ke.

Aturan pembelajaran *backpropagation* dengan 3 lapisan (1 lapisan *input*, 1 lapisan tersembunyi, dan 1 lapisan *output*) dan fungsi *aktivasi sigmoid*, digunakan untuk mendapatkan bobot-bobot w_j (bobot antara lapisan tersembunyi dengan lapisan *output*) dan w_{ji} (bobot antara lapisan *input* dengan lapisan tersembunyi) pada jaringan BPN⁺, perbaikan bobot dilakukan sebagai berikut.

$$\Delta w_j = \eta \left(-\frac{\partial E_k}{\partial w_j} \right) = \eta \delta_k y_{kj} \quad (2.24)$$

$$\Delta w_{ji} = \eta \left(-\frac{\partial E_k}{\partial w_{ji}} \right) = \eta \delta_{kj} x_{ki} \quad (2.25)$$



dengan

$$\delta_k = \alpha_k (d_k - y_k) y_k (1 - y_k) \quad (2.26)$$

$$\delta_{kj} = y_{kj} (1 - y_{kj}) \delta_k w_j \quad (2.27)$$

$y_k = g^+(x_k)$ adalah *output* jaringan, dan y_{kj} adalah *output* neuron ke- j pada lapisan tersembunyi untuk *input* x_k .

Dengan cara yang sama, pembelajaran juga dilakukan pada jaringan BPN⁻ untuk mendapatkan *output* jaringan $g^-(x_k)$. Fungsi biaya yang digunakan untuk pembelajarannya adalah :

$$E = \sum_{k=1}^p E_k = \frac{1}{2} \sum_{k=1}^p \alpha_k [d_k - g^-(x_k)]^2 \quad (2.28)$$

dengan α_k diberikan sebagai berikut:

$$\alpha_k = \begin{cases} \alpha; & d_k \geq g^-(x_k) \\ 1; & d_k < g^-(x_k) \end{cases} \quad (2.29)$$

dengan α adalah bilangan positif yang cukup kecil pada interval (0, 1).

Dengan menggunakan kedua algoritma pembelajaran tersebut, kita dapat menentukan 2 fungsi, $g^+(x)$ dan $g^-(x)$ dimana $g^-(x) \leq d_k \leq g^+(x)$, $k = 1, 2, \dots, p$.

Dari sini, dapat diperoleh interval:

$$G(x) = [g^-(x), g^+(x)] \quad (2.30)$$

Karena $g^+(x)$ dan $g^-(x)$ diperoleh dari pembelajaran yang terpisah, maka sangat dimungkinkan $g^+(x) < g^-(x)$. Sehingga, jika hal tersebut terjadi, maka dapat dilakukan modifikasi pada interval tersebut sebagai berikut:

$$h^-(x) = \begin{cases} g^-(x); & g^-(x_k) \leq g^+(x) \\ \frac{1}{2}(g^-(x) + g^+(x)); & g^-(x_k) > g^+(x) \end{cases} \quad (2.31)$$

$$h^+(x) = \begin{cases} g^+(x); & g^-(x_k) \leq g^+(x) \\ \frac{1}{2}(g^-(x) + g^+(x)); & g^-(x_k) > g^+(x) \end{cases} \quad (2.32)$$

Dengan demikian, interval yang terjadi adalah:

$$G(x) = [h^-(x), h^+(x)] \quad (2.33)$$

Sehingga batas interval dari persamaan regresi tersebut telah diketahui dengan $h^-(x)$ merupakan batas bawah interval regresi dan $h^+(x)$ merupakan batas atas interval regresi.

BAB III

ANALISIS KEBUTUHAN PERANGKAT LUNAK

3.1 Metode Analisis

Metode yang digunakan dalam membangun program Penyelesaian Model Regresi Menggunakan Pembelajaran *Neurofuzzy* ini adalah dengan menggunakan metode struktural dengan *flowchart* (diagram alir). Dengan metode ini diharapkan hasil analisis menjadi lebih terstruktur, sehingga dapat dijelaskan dengan baik serta memungkinkan masukan, proses-proses dan keluaran program dapat terdefinisi dengan baik.

3.2 Hasil Analisis

Berdasarkan analisis yang telah dilakukan maka dapat diketahui apa saja yang menjadi masukan sistem, keluaran sistem fungsi atau metode yang digunakan oleh sistem, kebutuhan perangkat keras, kebutuhan perangkat lunak serta antarmuka sistem yang akan dibuat, sehingga sistem yang dibuat nantinya sesuai dengan apa yang diharapkan.

3.2.1 Masukkan Sistem

Masukan dari program pencarian interval regresi menggunakan *neurofuzzy* adalah berupa variabel-variabel yang digunakan dalam membangun sistem *neurofuzzy*. Adapun data masukan yang dibutuhkan program, adalah :

1. Data-data *input* dari persamaan model regresi, yaitu berupa nilai x_i ($i=1,2,3,..,n$).
2. Data-data pasangan *output*-nya dari data-data input model regresi, yaitu berupa nilai d_k ($k=1,2,3,..,p$), dimana jumlah data *input* sama dengan jumlah data *output* ($n = p$).
3. *Error* maksimum, yang digunakan sebagai batas error yang diperbolehkan oleh program.
4. Jumlah *iterasi*, yaitu banyaknya perulangan yang dikerjakan oleh program dalam melakukan pembelajaran.
5. Jumlah unit *hidden*, yaitu banyaknya unit yang terdapat dalam lapisan *hidden* (lapisan tersembunyi).
6. Nilai *learning rate*, yaitu nilai yang diset sebagai laju pembelajaran dalam pembelajaran jaringan.
7. Nilai *momentum*, yaitu nilai yang diset untuk mempengaruhi hasil dari bobot-bobot pada pembelajaran jaringan.

3.2.2 Keluaran Sistem

Adapun keluaran dari program ini adalah berupa batas interval atas dan batas interval bawah dari suatu model persamaan regresi, setelah melalui proses pembelajaran dalam jaringan. Keluaran dari program akan ditampilkan dalam bentuk grafik, sehingga dapat dilihat dengan jelas rentang intervalnya. Dan juga hasil *output* pembelajaran akan disimpan kedalam *fileoutput.txt*.

3.2.3 Kebutuhan Fungsi

Kebutuhan fungsi yang dibutuhkan program adalah berupa fungsi-fungsi pembelajaran yang digunakan dalam jaringan *backpropagation* yang digunakan untuk melatih jaringan mencari batas atas dan batas bawah nilai interval regresi. Fungsi tersebut berupa penghitungan bobot tiap-tiap unit, pencarian nilai aktivasi, penghitungan total *error* serta pembaruan bobot-bobot tiap unit.

3.2.4 Kebutuhan Perangkat Keras

Program ini dapat berjalan dengan baik, apabila memenuhi standar minimal dari perangkat keras (*hardware*) yang dimiliki. Spesifikasi minimal dari perangkat keras yang dapat digunakan untuk menjalankan program ini, yaitu :

1. Satu unit komputer dengan spesifikasi prosessor Pentium IV.
2. Memori (RAM) 128 Mb.
3. Harddisk 2Gb.
4. Monitor VGA atau SVGA.
5. Mouse.
6. Keyboard.

3.2.5 Kebutuhan Perangkat Lunak

Perangkat lunak (*software*) yang dibutuhkan untuk pengembangan dan implementasi program Penyelesaian Model Regresi Menggunakan Pembelajaran *Neurofuzzy* ini adalah :

1. Sistem operasi Windows 2000, atau yang lebih tinggi.
2. Bahasa Pemrograman Matlab 7, untuk pengembangan dan pembuatan perangkat lunak

3.2.6 Kebutuhan Antarmuka

Antarmuka yang digunakan dalam program Penyelesaian Model Regresi Menggunakan Pembelajaran *Neurofuzzy* ini menggunakan *Guide*. Yang merupakan salah satu fasilitas pada Matlab untuk membangun antarmuka program. Sehingga program ini dapat dipakai oleh semua *user*, karena tampilan yang mudah dan tidak membuat user sulit untuk menjalankannya termasuk *user* yang masih pemula.

3.2.7 Kinerja Yang Diharapkan

Kinerja yang diharapkan dari hasil analisis yang diatas ialah perangkat lunak yang dapat mencari nilai batas interval dari sebuah model regresi dengan menggunakan metode pembelajaran *neurofuzzy*. Dimana *user* dapat memasukkan data *input* program, berupa *input-output* dari persamaan regresi.

BAB IV

PERANCANGAN PERANGKAT LUNAK

4.1 Metode Perancangan

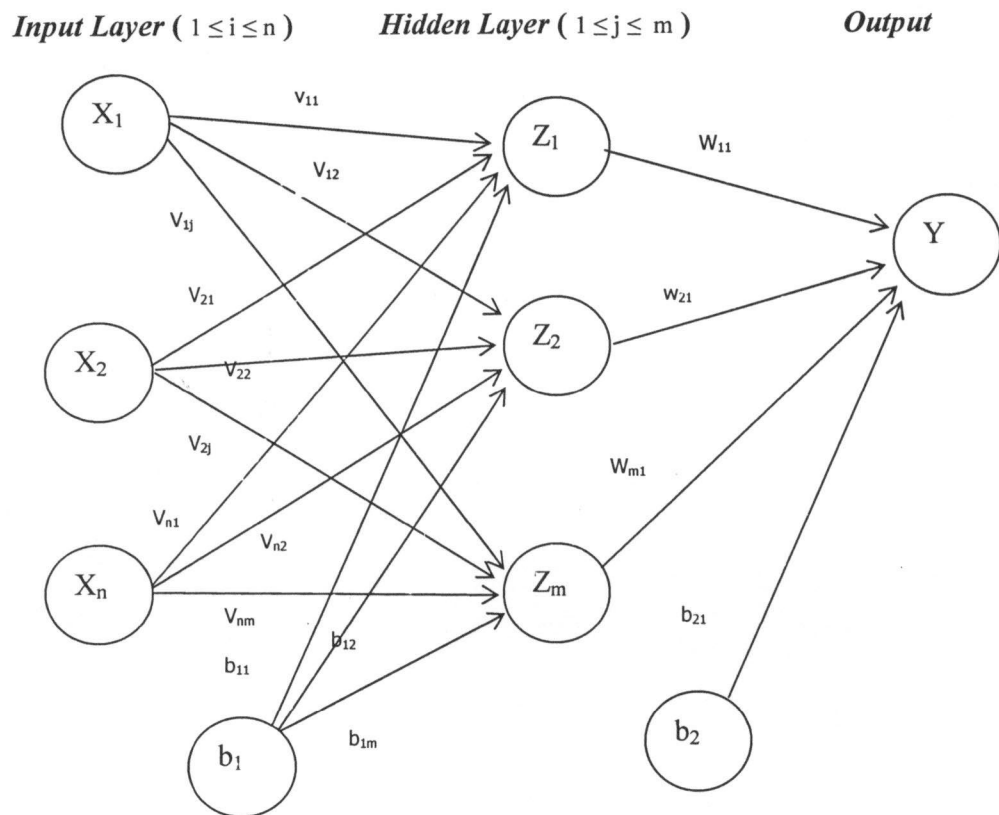
Metode perancangan yang digunakan dalam merancang Sistem Penyelesaian Model Regresi Menggunakan Pembelajaran *Neurofuzzy* adalah menggunakan metode *Top Down Design* dengan *Flowchart* (bagan alir). Dimana tahapan untuk tiap-tiap proses digambarkan dengan jelas dan mudah dimengerti.

4.2 Hasil Perancangan

Berdasarkan analisis yang telah dilakukan maka dapat diketahui apa saja yang menjadi masukan sistem, keluaran sistem, fungsi atau metode yang digunakan oleh sistem, kebutuhan perangkat keras, kebutuhan perangkat lunak serta antarmuka sistem yang akan dibuat, sehingga sistem yang dibuat nantinya sesuai dengan apa yang diharapkan. Dengan hasil analisis maka di dapat suatu gambaran tentang Sistem Penyelesaian Model Regresi Menggunakan Pembelajaran *Neurofuzzy*.

4.2.1 Perancangan Arsitektur

Merupakan perancangan yang menitik beratkan kepada penggambaran arsitektur jaringan saraf tiruan *backpropagation* yang digunakan dalam membangun sistem *Neurofuzzy* yang sesuai dengan analisis sebelumnya. Perancangan arsitektur jaringan *backpropagation* dapat dilihat pada gambar 4.1.



Gambar 4.1 Perancangan arsitektur backpropagation untuk neuro-fuzzy

Keterangan :

X_1, X_2, \dots, X_n : Unit masukan (*input*), dimana $1 \leq i \leq n$

n : Banyak data masukan (X)

Z_1, Z_2, \dots, Z_m : Unit *hidden layer*, dimana $1 \leq j \leq m$

m : Banyaknya unit *hidden*

V_{xy} : Bobot koneksi sel ke- x pada lapisan *input layer* dengan sel ke- y pada lapisan *hidden layer*

W_{xy} : Bobot koneksi sel ke- x pada lapisan *hidden layer* dengan sel ke- y pada lapisan *layer output*

- b_{1x} : Bobot koneksi bias pertama ke sel-x pada lapisan *hidden layer*
- b_{2x} : Bobot koneksi bias kedua dengan unit output pada lapisan *layer output*
- Y : Unit *output* jaringan *backpropagation*



4.2.2 Fase Propagasi Maju

Merupakan fase pada jaringan *backpropagation* dimana pada tiap-tiap unit menghitung jumlah bobot yang masuk kemudian menyebarkan hasilnya ke lapisan berikutnya. Fase ini dimulai dari :

1. Unit *input* menerima sinyal *input* x_i dan menyebarkan sinyal bobot tersebut ke unit pada lapisan tersembunyi
2. Unit pada lapisan tersembunyi menjumlahkan sinyal bobot yang dikirimkan unit lapisan *input*, lalu dihitung nilai aktivasinya. Hasil dari nilai aktivasi lalu disebarkan ke lapisan berikutnya (lapisan *output*)
3. Unit pada lapisan *output* menjumlahkan sinyal bobot dari nilai teraktivasi yang dikirimkan lapisan tersembunyi. Lalu nilai tersebut dihitung aktivasinya.

Kemudian dilakukan penghitungan *error* pembelajaran, yaitu :

- a. Nilai *error* dihitung dengan menghitung selisih antara hasil akhir (y_k) dengan *output* ideal (t_k).

$$error = (t_k - y_k) \quad (4.1)$$

- b. Minimalkan *error* yang terjadi dengan menghitung jumlah dari *error* untuk semua sel keluaran dikalikan dengan faktor $\frac{1}{2}$.

$$E = \frac{1}{2} \sum_{k=1}^p \alpha_k [t_k - y_k]^2 \quad (4.2)$$

Keterangan :

P : *maximum iterasi*

α_k : nilai *alfa* yang digunakan dalam pembelajaran *neurofuzzy*

Jika *error* pelatihan lebih kecil dari toleransi *error* yang diharapkan maka, iterasi berhenti dan bobot-bobot akhir telah didapat.

4.2.3 Fase Propagasi Mundur

Merupakan fase dalam *backpropagation* yang menghitung perubahan bobot pada tiap lapisan untuk diperbaiki sehingga tercapai hasil pembelajaran yang diharapkan.

1. Lapisan *output*

Perhitungan *error* balik dari lapisan tersembunyi adalah :

$$\delta_k = \alpha_k (t_k - y_k) y_k (1 - y_k) \quad (4.3)$$

Perbaiki bobot yang terletak antar lapisan *output* dengan lapisan tersembunyi adalah :

Jika iterasi=0 (iterasi untuk pertama kalinya) maka perhitungannya :

$$\Delta w_j = \eta \delta_k y_{kj} \quad (4.4)$$

Jika iterasi $\diamond 0$ (bukan iterasi untuk pertama kalinya) maka perhitungannya :

$$\Delta w_j = mom * \Delta w_{j-1} + (1 - mom) * \eta \delta_k y_{kj} \quad (4.5)$$

$$w_j(\text{baru}) = w_j(\text{lama}) + \Delta w_j \quad (4.6)$$

Keterangan :

η : *learning rate* (laju pembelajaran)

mom : nilai momentum

Δw_{j-1} : delta bobot *output-hidden* sebelumnya

δ_k : hasil penghitungan *error* pada lapisan *output*

y_{kj} : hasil dari nilai aktivasi pada unit *input - hidden*

2. Lapisan tersembunyi

Melakukan penjumlahan *error* balik dari unit lapisan *output*, kemudian dikalikan dengan bobotnya masing-masing pada tiap unit lapisan tersembunyi (w_{jk}).

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (4.7)$$

Sehingga dapat dihitung *error* balik pada lapisan tersembunyi adalah :

$$\delta_{kj} = y_{kj} (1 - y_{kj}) \delta_{in_j} \quad (4.8)$$

Keterangan :

y_{kj} : *output neuron* ke- j pada lapisan tersembunyi untuk *input* x_k .

Perhitungan bobot yang terletak antara lapisan tersembunyi dengan lapisan *input* adalah :

Jika iterasi=0 (iterasi untuk pertama kalinya) maka perhitungannya :

$$\Delta v_{ji} = \eta \delta_{kj} x_{ki} \quad (4.8)$$

Jika itersi $\diamond 0$ (bukan iterasi untuk pertama kalinya) maka perhitungannya :

$$\Delta v_{ji} = mom * \Delta v_{ji-1} + (1 - mom) * \eta \delta_{kj} x_{ki} \quad (4.9)$$

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (4.10)$$

Keterangan :

η : *learning rate* (laju pembelajaran)

mom : nilai momentum

Δv_{ji-1} : delta *input-hidden* sebelumnya

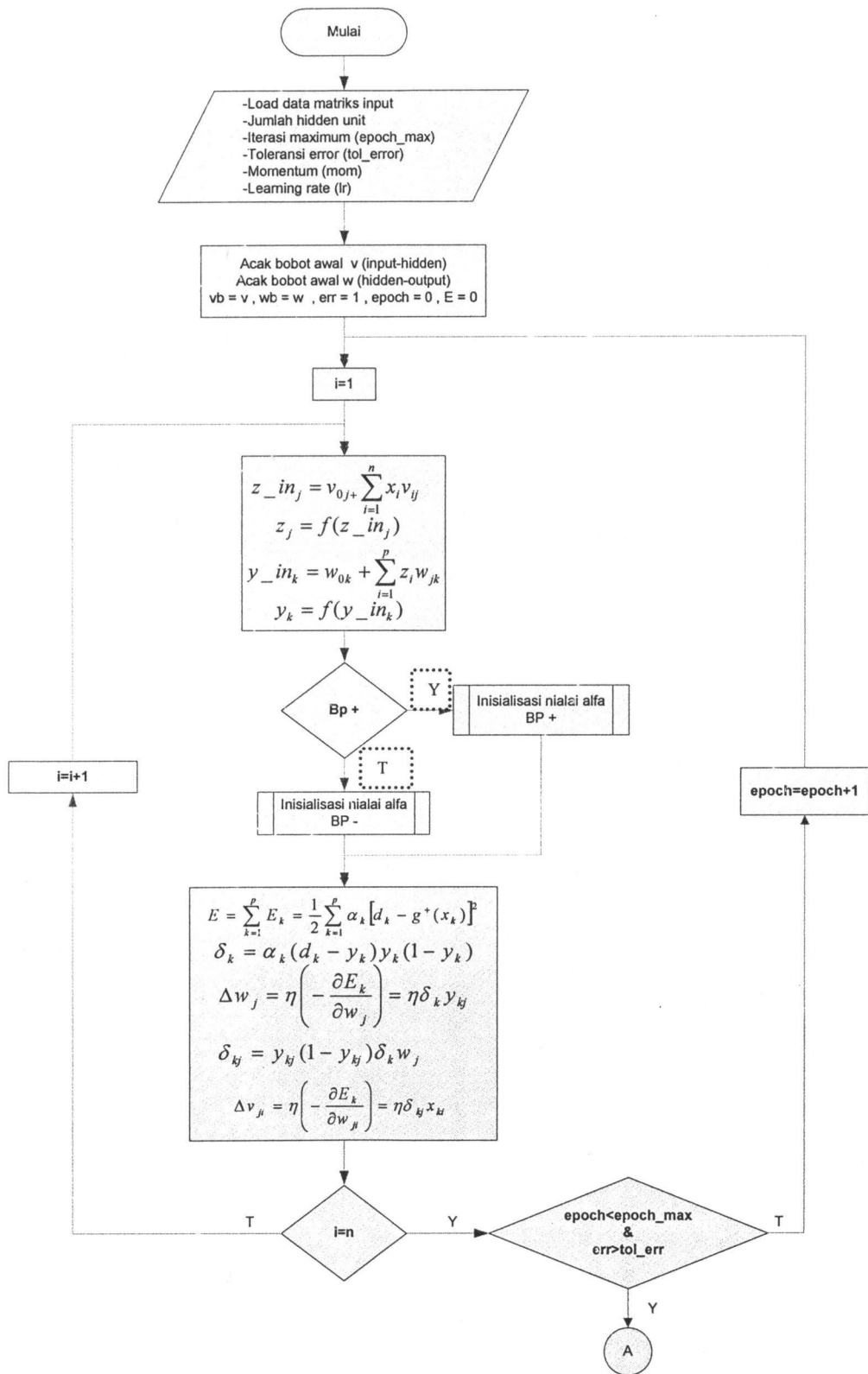
δ_{kj} : hasil penghitungan *error* pada lapisan tersembunyi

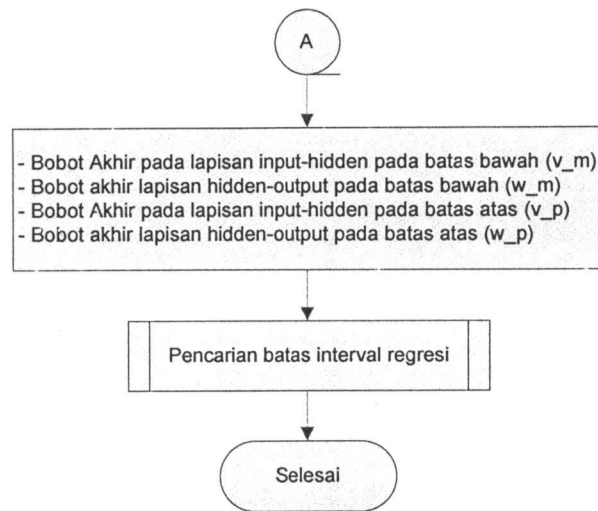
x_{ki} : nilai unit *input* (x) pada data ke-i

4.3 Perancangan Diagram Alir Sistem

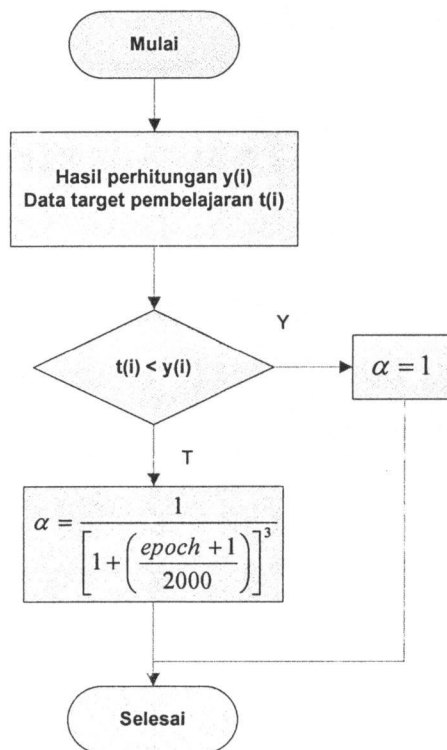
Bagan alir sistem (*flowchart*) merupakan salah satu metode analisis yang sering digunakan untuk menggambarkan keseluruhan langkah kerja dari sistem yang akan dibuat, sehingga tiap-tiap urutan dari alur kerja sistem dapat dipahami dengan mudah dan jelas.

Gambar 4.2 adalah *flowchart* untuk Sistem Penyelesaian Model Regresi Menggunakan Pembelajaran *Neurofuzzy* :



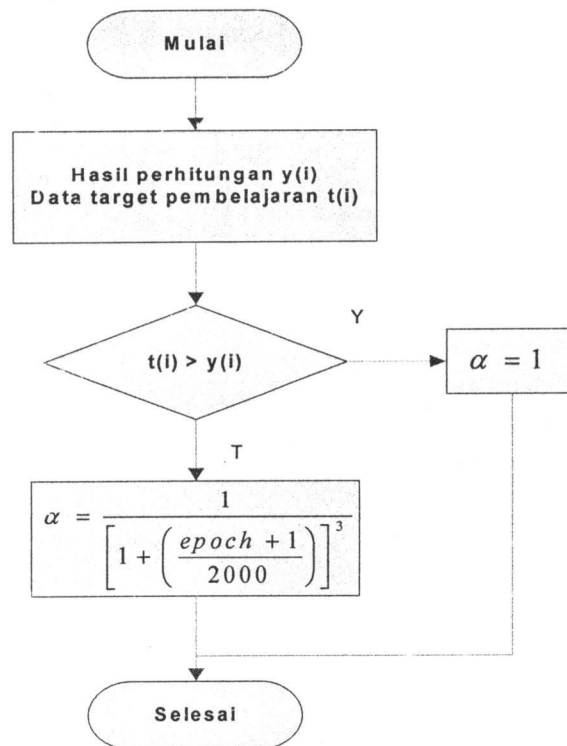


Gambar 4.2 Flowchart pembelajaran neuro-fuzzy



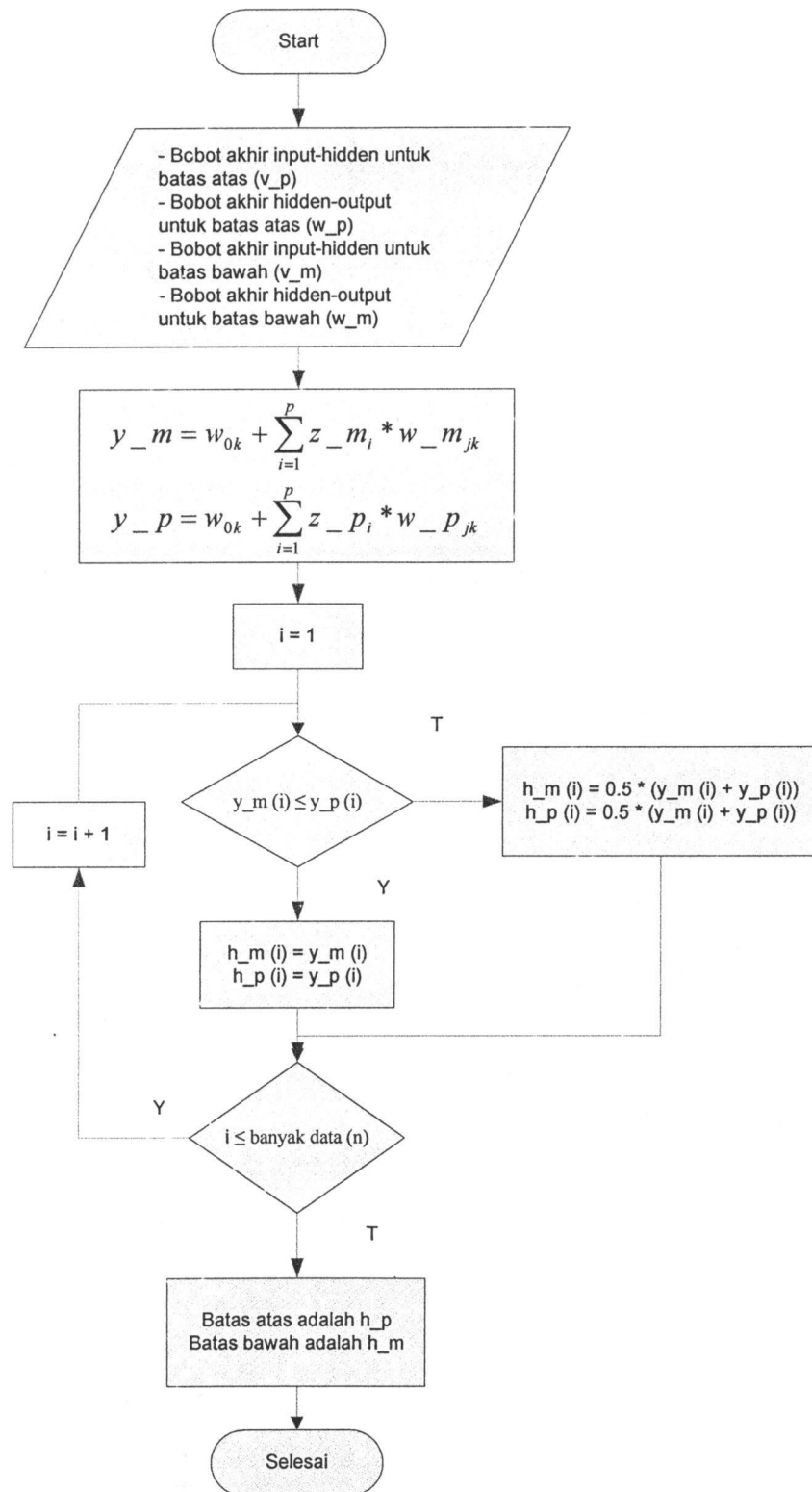
Gambar 4.3 Flowchart inialisasi nilai alfa pada BP -

Gambar 4.3 adalah *flowchart* yang menggambarkan alur proses yang dikerjakan untuk melakukan inisialisasi nilai alfa pada pembelajaran jaringan *backpropagation* batas bawah.



Gambar 4.4 Flowchart inisialisasi nilai alfa pada BP +

Gambar 4.4 adalah *flowchart* yang menggambarkan alur proses yang dikerjakan untuk melakukan inisialisasi nilai alfa pada pembelajaran jaringan *backpropagation* batas atas.



Gambar 4.5 Flowchart hasil batas atas dan batas bawah dari pembelajaran neuro-fuzzy

Gambar 4.5 adalah *flowchart* yang menggambarkan alur proses pada penentuan batas atas dan batas bawah yang didapatkan pada pembelajaran *backpropagation* sebelumnya.

4.4 Perancangan Struktur Data

Berikut adalah perancangan struktur data yang digunakan dalam membangun Sistem Penyelesaian Model Regresi Menggunakan Pembelajaran *Neurofuzzy*.

4.4.1 Perancangan Struktur Data Input

Perancangan struktur data input adalah sebagai berikut:

$X = \text{array } [1..n] \text{ of double}$

$T = \text{array } [1..n] \text{ of double}$

$V_{\text{awal}} = \text{array } [1..a, 1..b] \text{ of double}$

$W_{\text{awal}} = \text{array } [1..c] \text{ of double}$

Keterangan :

- n adalah banyak data input
- a adalah jumlah unit *input* + satu unit *bias*
- b adalah jumlah unit *hidden*
- c adalah jumlah unit *hidden* + satu unit *bias*

4.4.2 Perancangan Struktur Data Output

Perancangan struktur data output adalah sebagai berikut :

`h_m = array [1..n] of double`

`h_p = array [1..n] of double`

Keterangan :

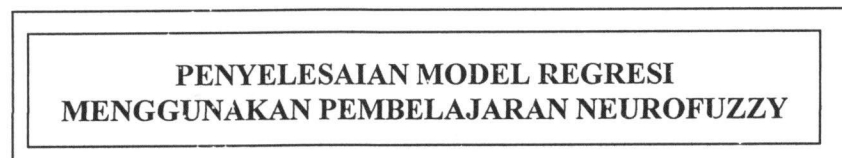
- *n* adalah jumlah/banyak data input

4.5 Perancangan Antarmuka

Rancangan antarmuka dari program aplikasi pencarian batas interval regresi menggunakan *neurofuzzy* dibangun menggunakan perancangan model grafis (*visual*). Sehingga memudahkan pengguna dalam menjalankan aplikasi dan mengurangi waktu pembelajaran serta pengoperasian program aplikasi tersebut. Perancangan tersebut dibagi menjadi tiga bagian, yaitu : bagian antarmuka judul, bagian antarmuka menu, bagian antarmuka keluaran.

4.5.1 Rancang Antarmuka Judul

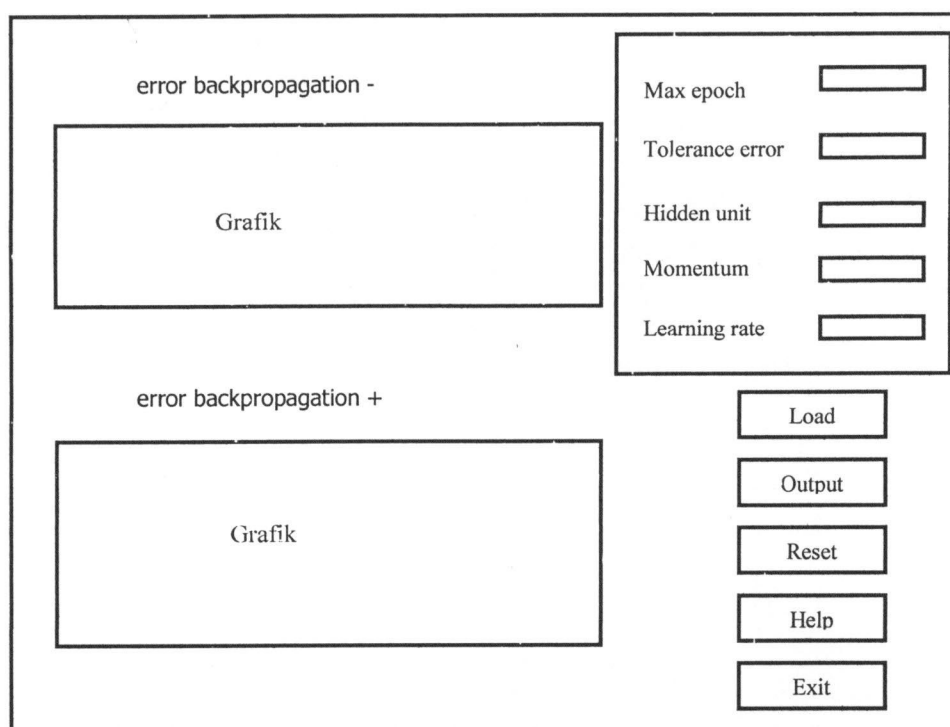
Antarmuka hanya merupakan judul dari Sistem Penyelesaian Model Regresi Menggunakan Pembelajaran *Neurofuzzy*. Gambar 4.6 adalah rancangan antarmuka judul.



Gambar 4.6 Rancang antarmuka judul

4.5.2 Rancang Antarmuka Menu

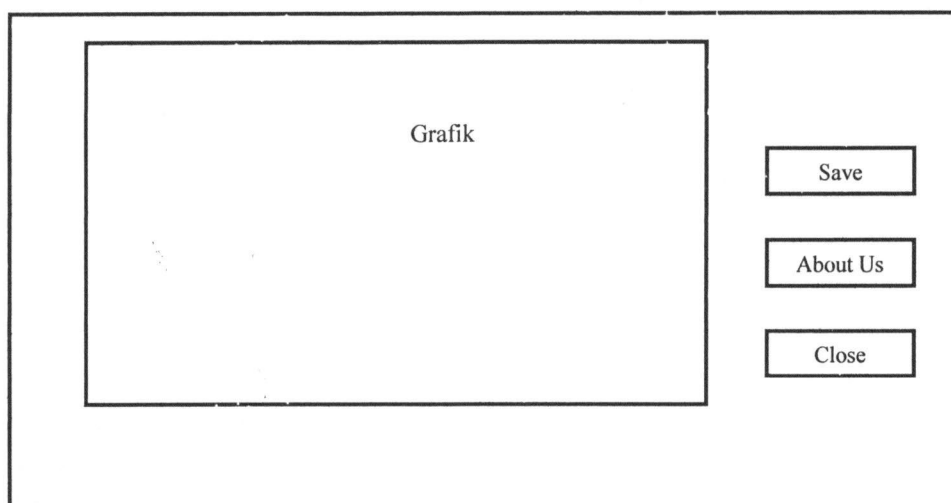
Antarmuka ini dibuat sebagai sarana *user* dalam memasukkan *input* data yang diperlukan dalam menjalankan program aplikasi. Pada bagian antarmuka ini disediakan tampilan berupa grafik, yang menggambarkan pembelajaran *error* dalam jaringan yang dibangun. Gambar 4.7 adalah rancangan antarmuka menu.



Gambar 4.7 Rancang antarmuka menu

4.5.3 Rancang Antarmuka Keluaran

Antarmuka ini berhubungan dengan output dari sistem yang dijalankan. Antarmuka berisi grafik dari hasil keluaran dan fasilitas untuk mencetak hasil proses pembelajaran. Gambar 4.8 adalah rancangan antarmuka hasil keluaran (*output*).



Gambar 4.8 Rancang antarmuka keluaran

4.6 Perancangan File

Perancangan *file* yang digunakan atau dibutuhkan dalam menjalankan program aplikasi Penyelesaian Model Regresi Menggunakan Pembelajaran *Neurofuzzy*.

4.6.1 Rancang File Untuk Input Data

Pada bagian *input data file* yang digunakan sistem adalah *file* data yang berformat *.txt*. Rancangan *file* tersebut berbentuk list menurun dengan data-data angka yang membentuk matriks. Dimana tiap kolom dalam data *input* dipisahkan oleh tanda *spasi*. Kolom terakhir pada matriks dijadikan sebagai target dari pembelajaran jaringan, dan kolom sebelum target adalah data *input* jaringan. Gambar 4.9 adalah contoh dari format data *input* dalam file *.txt*.

0.567	0.678	0.239
-0.224	-0.895	0.341
-0.567	0.897	0.787
0.432	0.452	0.176

↓ Data input (x) ↓ Data target (t)

Gambar 4.9 Contoh file untuk input data

4.6.2 Rancang File Untuk Output Data

Pada bagian *output* sistem, *file* yang digunakan untuk menyimpan laporan pembelajaran *neurofuzzy* adalah berupa *file* dengan format *.txt*. Isi dari *file output.txt* adalah matriks hasil pembelajaran *neurofuzzy*, yaitu matriks hasil batas atas dan batas bawah dari persamaan regresi.

BAB V

IMPLEMENTASI PERANGKAT LUNAK

5.1 Pengantar Implementasi

Tujuan dari tahap implementasi ini untuk memastikan perangkat lunak yang dibuat dapat bekerja secara efektif dan efisien sesuai dengan yang diinginkan, untuk itu pada tahap implementasi ini akan menjelaskan antara lain; pemilihan bahasa pemrograman yang digunakan, lingkungan pengembangan, dan uji coba terhadap perangkat lunak.

5.2 Batasan Implementasi

Perangkat lunak yang dibuat memiliki batasan-batasan dalam implementasinya yang dapat lebih mengarahkan bentuk program aplikasi sesuai dengan rancangan program. Diasumsikan bahwa perangkat lunak yang dibuat merupakan suatu program aplikasi (*software*) yang dapat digunakan untuk penelitian pembelajaran *Neurofuzzy* dalam menyelesaikan masalah regresi dengan batasan-batasan seperti yang telah dijelaskan dalam batasan masalah.

5.2.1 Pemilihan Bahasa Pemrograman

Sistem Penyelesaian Model Regresi Menggunakan Pembelajaran *Neurofuzzy* dibangun menggunakan bahasa pemrograman *Matlab*. Alasan penggunaan bahasa pemrograman *Matlab* adalah karena kemampuan *Matlab* sebagai bahasa pemrograman sekaligus alat visualisasi, yang menawarkan banyak

kemampuan untuk menyelesaikan berbagai kasus yang berhubungan langsung dengan disiplin keilmuan matematika, seperti bidang rekayasa teknik, fisika, statistika, modeling dan komputasi. *Matlab* dibangun dari bahasa induknya yaitu *C*, oleh karena itu *Matlab* juga mewarisi kelebihan dari bahasa *C* salah satunya yaitu mampu berjalan pada semua *platform*. [AWA06]

Kemudahan lain yang didapat dari *Matlab* adalah memberikan sistem interaktif yang menggunakan konsep *array*/matrik sebagai standart variabel elemennya tanpa membutuhkan pendeklarasian *array* seperti pada bahasa lainnya. [AWA06]

5.3 Implementasi Sistem

Implementasi ini berisi tentang pembuatan tampilan antarmuka dari perangkat lunak. Sesuai dengan perancangan antarmuka yang telah dijelaskan pada bab empat.

5.3.1 Tampilan Form Judul

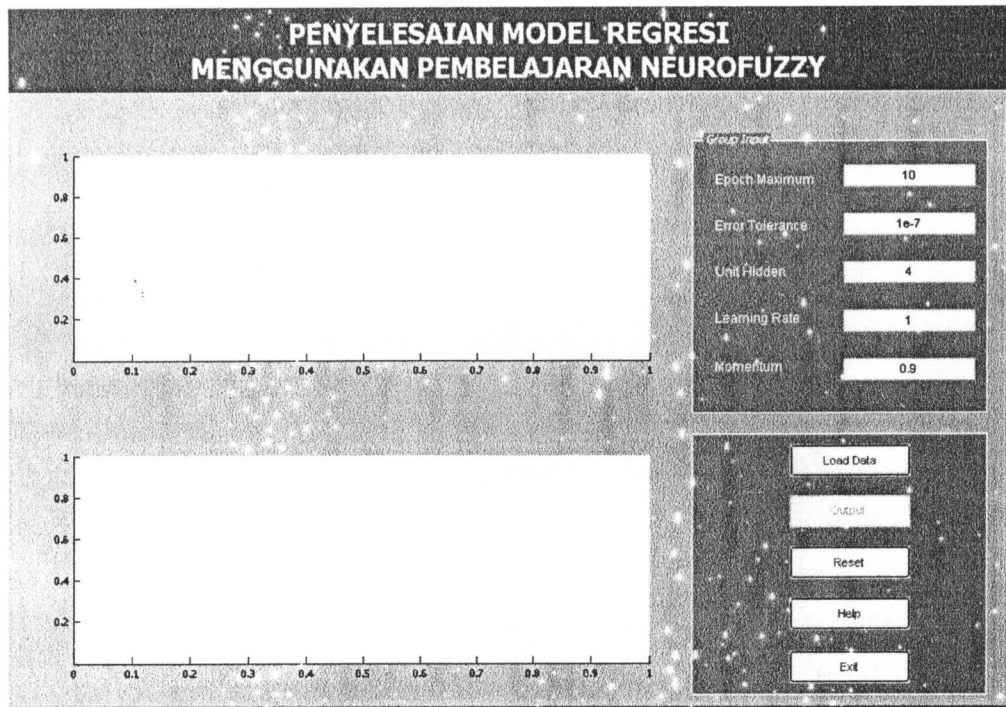
Merupakan form yang diletakkan paling atas pada form utama. Yang menggambarkan judul dari sistem yang dibuat. Gambar 5.1 adalah tampilan form judul,



Gambar 5.1 Tampilan form judul

5.3.2 Tampilan Form Utama

Tampilan utama dari sistem yang dibuat, terdiri dari beberapa menu input masukan dan tombol-tombol eksekusi. Gambar 5.2 adalah tampilan dari menu utama,



Gambar 5.2 Tampilan utama

Beberapa input yang diperlukan adalah sebagai berikut :

a. *Epoch Maximum*

Banyaknya *iterasi* yang digunakan dalam pembelajaran jaringan *backpropagation*.

b. *Error Tolerance*

Toleransi *error* yang diperbolehkan, dengan kata lain *error* maksimum dari pembelajaran.



c. *Unit Hidden*

Jumlah unit *hidden* yang terdapat dalam lapisan *hidden*.

d. *Learning Rate*

Laju pembelajaran jaringan *backpropagation* memiliki nilai antara 0 sampai 1.

e. *Momentum*

Nilai *momentum* sewaktu penghitungan perubahan bobot jaringan *backpropagation* nilai *defaultnya* 0.9 (terbaik).

Tombol-tombol yang terdapat pada menu utama dijelaskan sebagai berikut ;

1. Tombol *Load data*

Digunakan untuk mengambil data matriks input, sekaligus melakukan pembelajaran terhadap data input yang diambil.

2. Tombol *Output*

Digunakan untuk menampilkan form *output*, yaitu form yang menampilkan hasil dari pembelajaran.

3. Tombol *Reset*

Digunakan untuk mereset semua data variabel, dan juga grafik menjadi keadaan semula.

4. Tombol *Help*

Menampilkan teori dari pembelajaran *Neuro-fuzzy* terhadap model regresi.

5. Tombol *Exit*

Digunakan untuk keluar dari program utama.

Berikut adalah prosedur atau baris perintah yang terdapat didalam tombol

Load Data :

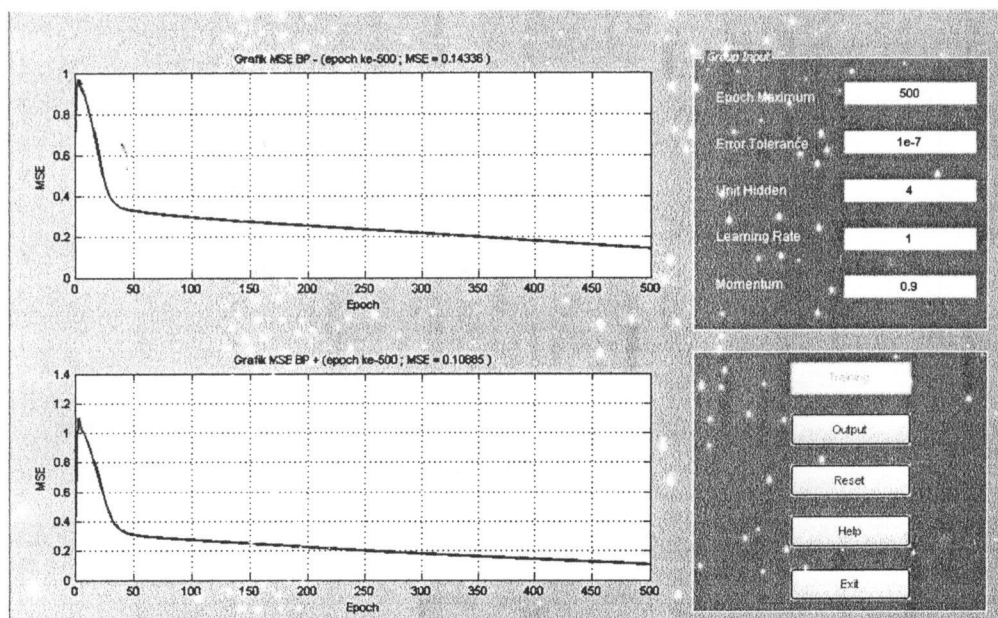
```
% ambil matriks data input
load data input;
normalisasi data input;
epoch_max=str2num(get(formku.ed_iter,'String'));
tol_error=str2double(get(formku.ed_err,'String'));
lr=str2double(get(formku.ed_lr,'String'));
nd=str2num(get(formku.ed_nd,'String'));
mom=str2double(get(formku.ed_mom,'String'));
% pembelajaran BPN-
[vm,wm]=neuralfuzzy(X_train,T_train,epoch_max,tol_error,
lr,nd,mom,0,data);
% pembelajaran BPN+
[vp,wp]=neuralfuzzy(X_train,T_train,epoch_max,tol_error,
lr,nd,mom,1,data);
simpan kedalam file 'var_input.mat';
```

Berikut adalah prosedur atau baris perintah yang terdapat didalam tombol

Output:

```
% ambil data variabel input
load data 'var_input.mat';
% inisialisasi
var=data-data variabel pada file 'var_input.mat';
% Fungsi untuk mencari output dari hasil pembelajaran
[y_m,y_p]=backpropagation(var.vm,var.wm,var.vp,var.wp,var.
nd,var.X_train,var.data);
for i=1:banyak data
    if y_m(i)<=y_p(i)
        h_m(i)=y_m(i);
        h_p(i)=y_p(i);
    else
        h_m(i)=0.5*(y_m(i)+y_p(i));
        h_p(i)=0.5*(y_m(i)+y_p(i));
    end;
end;
de-normalisasi hasil pembelajaran dan matriks input;
simpan kedalam file 'var_output.mat';
```


Gambar 5.3 adalah tampilan dari menu utama ketika dilakukannya *training* data *input*,

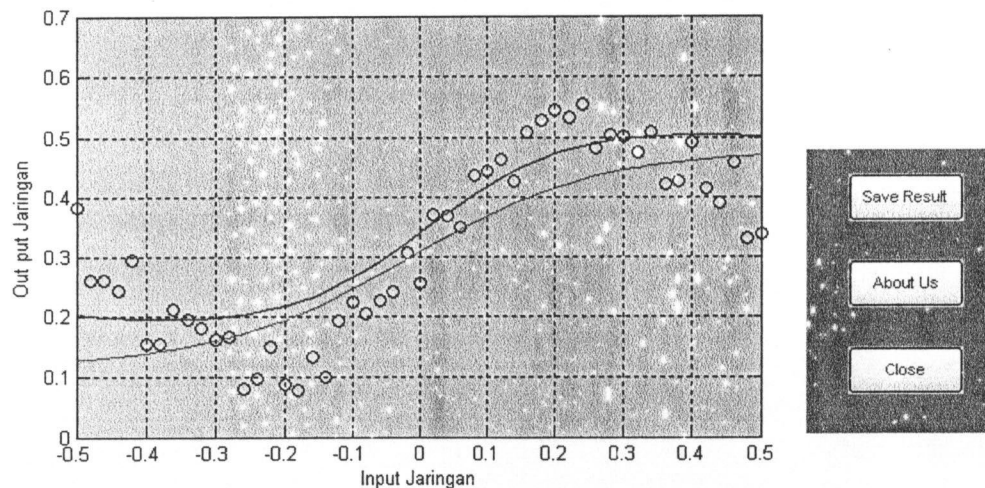


Gambar 5.3 Tampilan proses pembelajaran

5.3.3 Tampilan Form Output

Tampilan antarmuka untuk menampilkan hasil dari proses pembelajaran sistem pada form utama, dapat dilihat pada gambar 5.4 berikut,

HASIL PEMBELAJARAN NEUROFUZZY



Gambar 5.4 Tampilan form output

Tampilan *output* terdiri tiga jenis tombol, dijelaskan sebagai berikut :

1. Tombol *Save Result*

Digunakan untuk melakukan penyimpanan hasil dari sistem kedalam format *file .txt*.

2. Tombol *About Us*

Menampilkan informasi tentang pembuat perangkat lunak.

3. Tombol *Close*

Menutup tampilan *output* sistem.

Berikut adalah prosedur atau baris perintah yang terdapat dalam prosedur *CreateFcn form Output* ;

```

% ambil data variabel output
Load data 'var_output.mat';
% inisialisasi
Data=data-data variabel pada file 'var_output.mat';
%gambar grafik
Plot (data.X,data.T,data.X,data.h_m,data.X,data.h_p);

```

Berikut adalah prosedur atau baris perintah yang terdapat dalam tombol

Save Result :

```

% tentukan lokasi penyimpanan
set lokasi penyimpanan dan nama file;
% inisialisasi
data1=load data-data variable pada file 'var_input.mat';
data2=load data-data variable pada file 'var_output.mat';
% membuka file untuk ditulisi
f=fopen(namafile,'w');
% tulis kedalam file isi dari matriks awal
fprintf(f, matriks awal);
% tulis kedalam file isi matriks hasil batas bawah
fprintf(f, matriks hasil pembelajaran batas bawah);
% tulis kedalam file isi matriks hasil batas atas
fprintf(f, matriks hasil pembelajaran batas atas);
% tutup file
fclose(f);

```

BAB VI

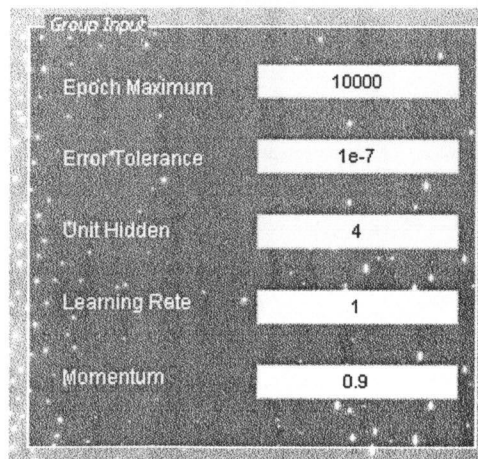
ANALISIS KINERJA PERANGKAT LUNAK

6.1 Pengujian Normal

Pengujian normal dilakukan dengan memberikan masukan yang sesuai dengan spesifikasi awal dan ketentuan-ketentuan yang diperbolehkan sistem. Pengujian dilakukan pada proses-proses berikut ini.

6.1.1 Proses Input Data

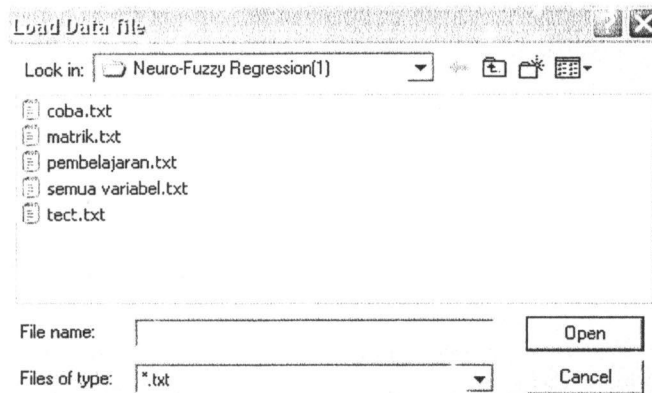
Pengujian pada proses ini dilakukan dengan memasukkan konfigurasi dari jaringan saraf tiruan pada form inisialisasi variabel input dengan benar atau normal, lalu mengambil data matriks input (*file.txt*) yang akan dilakukan *training*. Proses pemasukkan konfigurasi variabel input dapat dilihat pada gambar 6.1



Group Input	
Epoch Maximum	10000
Error Tolerance	1e-7
Unit Hidden	4
Learning Rate	1
Momentum	0.9

Gambar 6.1 Tampilan form variabel input

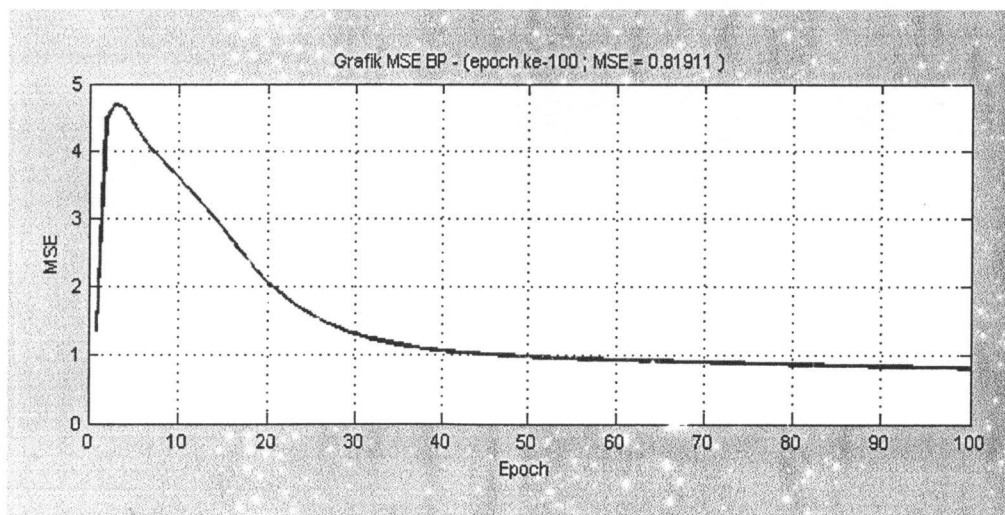
Proses pengambilan data matriks input dapat dilakukan dengan menekan tombol “Load Data”, gambar 6.2 adalah tampilan untuk *input* data matriks.



Gambar 6.2 Tampilan load data matriks input

6.1.2 Proses Pelatihan Data

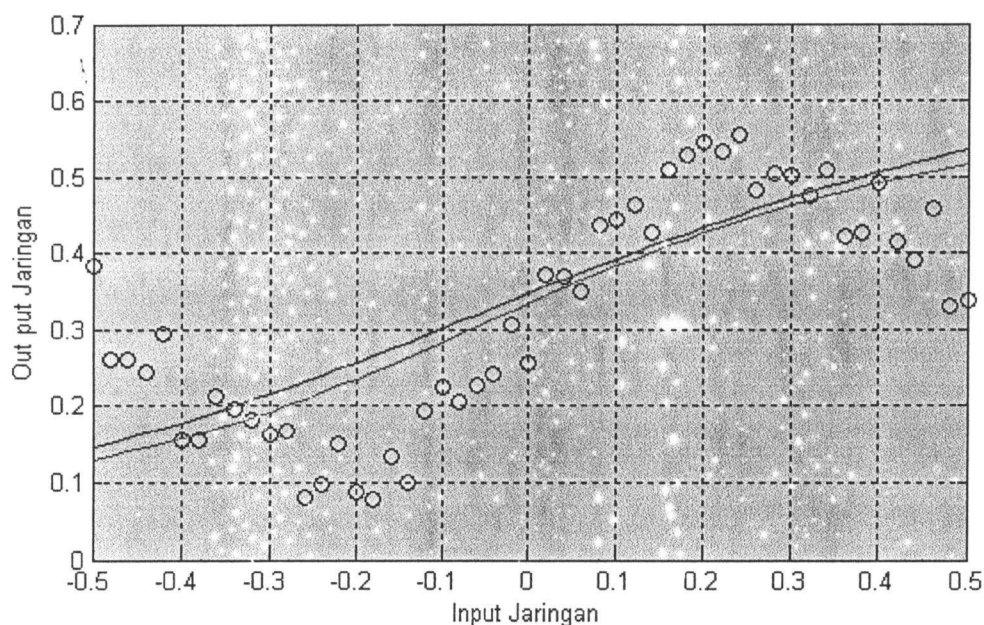
Setelah melalui proses *input* data, dilanjutkan pada proses pelatihan data. Data matriks akan dilatih berdasarkan variabel-variabel yang telah diinputkan sebelumnya. Gambar 6.3 adalah gambar dari pembelajaran *error* pada pelatihan jaringan *backpropagation*.



Gambar 6.3 Tampilan pembelajaran error jaringan

6.1.3 Proses Output Pembelajaran

Setelah sistem selesai melakukan pembelajaran jaringan *backpropagation*, maka *output* dari pembelajaran dapat dilihat dengan menekan tombol “Output”. Gambar 6.4 adalah tampilan dari *output* pembelajaran *neurofuzzy*.



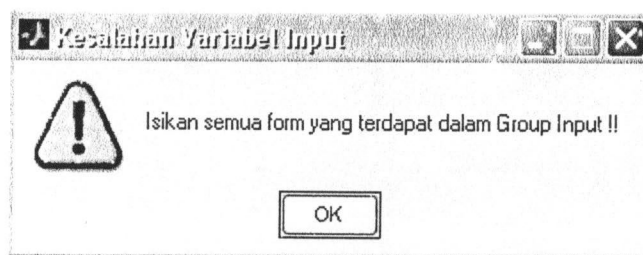
Gambar 6.4 Tampilan output pembelajaran *neuro-fuzzy*

6.2 Pengujian Tidak Normal

Pengujian tidak normal dilakukan dengan cara memberikan masukan yang tidak sesuai dengan spesifikasi awal atau ketentuan-ketentuan yang tidak diijinkan sistem. Dengan kata lain pengujian cara ini dilakukan dengan memberikan isian yang salah, sehingga dapat diketahui apakah sistem dapat menangani kesalahan yang ada dan memberitahukannya kepada pengguna sistem.

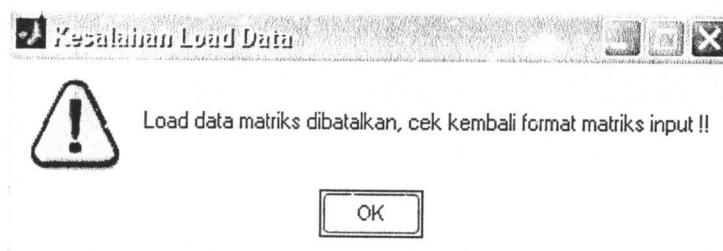
6.2.1 Proses Inisialisasi

Pengujian pada proses ini dilakukan dengan tidak mengisikan salah satu atau semua variabel-variabel input yang akan digunakan untuk proses pembelajaran jaringan *backpropagation*. Gambar 6.5 adalah tampilan kesalahan yang akan ditampilkan sistem jika terjadi kesalahan tersebut.



Gambar 6.5 Tampilan kesalahan variabel input

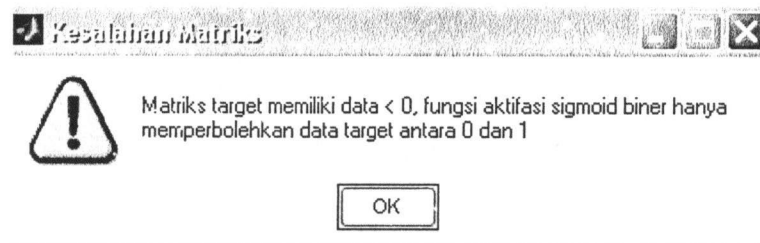
Jika matriks data *input* yang akan di lakukan proses pelatihan, tidak sesuai dengan format yang seharusnya maka akan ditampilkan pesan kesalahan, seperti pada gambar 6.6



Gambar 6.6 Tampilan kesalahan load data matriks

Jika matriks data *input* memiliki target dengan nilai < 0 , maka akan muncul pesan kesalahan. Karena dalam pembelajaran *backpropagation* digunakan fungsi

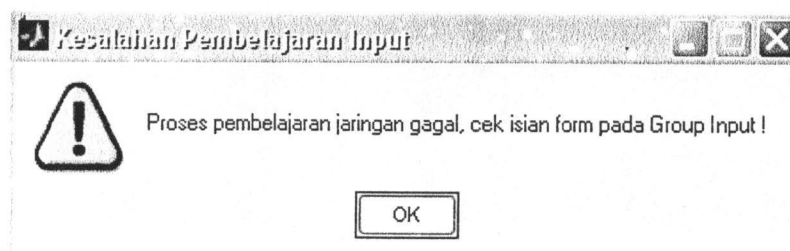
aktifasi *sigmoid biner*, dimana *output* yang dihasilkan berada dalam rentang nilai antara 0 dan 1. Gambar 6.7 adalah pesan kesalahan yang ditampilkan bila data target tidak sesuai.



Gambar 6.7 Tampilan kesalahan matriks target

6.2.2 Proses Pelatihan Data

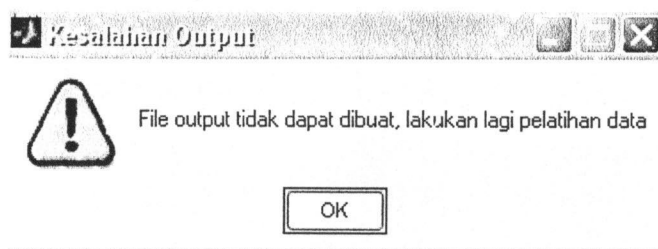
Pengujian pada proses ini adalah untuk mendeteksi kesalahan dalam proses pelatihan jaringan *backpropagation*. Kesalahan ini bisa diakibatkan karena format pengisian pada form input variabel yang tidak semestinya. Misalkan variabel *unit hidden* seharusnya diisi oleh data bilangan, akan tetapi diisi data huruf / abjad. Gambar 6.8 adalah tampilan pesan kesalahan pada pembelajaran jaringan *backpropagation*.



Gambar 6.8 Tampilan kesalahan pembelajaran jaringan backpropagation

6.2.3 Proses Output Pembelajaran

Pengujian pada proses ini adalah untuk mendeteksi kesalahan sewaktu sistem membuat *file output* pembelajaran. Kesalahan tersebut dapat diakibatkan karena hilangnya *file* yang dibutuhkan sistem untuk membuat *output* pembelajaran *neurofuzzy*. Gambar 6.9 adalah tampilan dari kesalahan *output* pembelajaran sistem.



Gambar 6.9 Tampilan kesalahan output pembelajaran

6.3 Contoh Kasus

Pada sub bab ini akan ditampilkan contoh kasus model regresi , yang akan diselesaikan oleh sistem. Misalkan data matriks regresi yang akan dipelajari dengan *neurofuzzy*, adalah :

Tabel 6.1. Pasangan *input-output* regresi

Data ke-	Variabel input (x)	Target output (d(x))
1	-0,50	0,3846
2	-0,48	0,2611
3	-0,46	0,2609
4	-0,44	0,2437
5	-0,42	0,2954
6	-0,40	0,1563
7	-0,38	0,1558
8	-0,36	0,2135
9	-0,34	0,1964
10	-0,32	0,1825
11	-0,30	0,1635

Data ke-	Variabel input (x)	Target output (d(x))
12	-0,28	0,1668
13	-0,26	0,0806
14	-0,24	0,0977
15	-0,22	0,1498
16	-0,20	0,0875
17	-0,18	0,0773
18	-0,16	0,1328
19	-0,14	0,0987
20	-0,12	0,1948
21	-0,10	0,2238
22	-0,08	0,2049
23	-0,06	0,2274
24	-0,04	0,2408
25	-0,02	0,3060
26	0,00	0,2555
27	0,02	0,3705
28	0,04	0,3692
29	0,06	0,3490
30	0,08	0,4365
31	0,10	0,4442
32	0,12	0,4640
33	0,14	0,4279
34	0,16	0,5096
35	0,18	0,5284
36	0,20	0,5450
37	0,22	0,5338
38	0,24	0,5553
39	0,26	0,4834
40	0,28	0,5046
41	0,30	0,5020
42	0,32	0,4739
43	0,34	0,5093
44	0,36	0,4228
45	0,38	0,4279
46	0,40	0,4927
47	0,42	0,4160
48	0,44	0,3916
49	0,46	0,4588
50	0,48	0,3300
51	0,50	0,3373

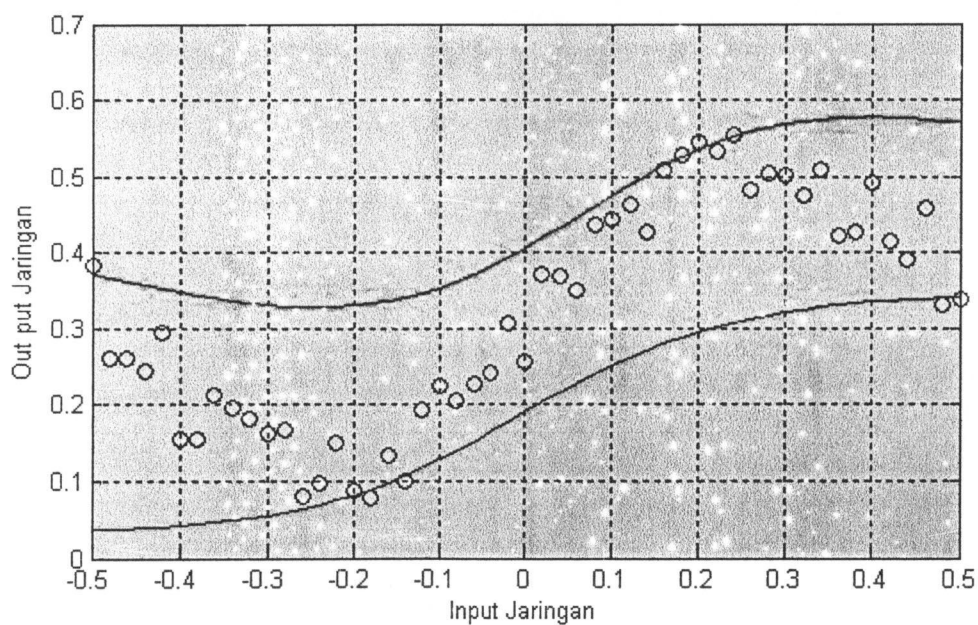
Selanjutnya akan dilakukan pelatihan jaringan *backpropagation* terhadap matriks input tersebut. Dimana bobot-bobot awal jaringan *backpropagation* diset menjadi bilangan *random* dengan *range* 0.5 sampai -0.5.

Parameter-parameter variabel input diberikan nilai sebagai berikut :

1. *Maximum iterasi* = 10000
2. *Error Tolerance* = $1e-7$ (1×10^{-7})
3. *Unit Hidden* = 4

a. Pembelajaran dengan *learning rate* = 0.25 dan *momentum* = 0,9

Gambar 6.10 adalah grafik *output* yang ditampilkan pada pembelajaran data *neurofuzzy*.



Gambar 6.10 Tampilan output dengan $lr = 0.25$ dan $mom = 0,9$

Keterangan dari gambar 6.10 diatas adalah :

- a. Gambar di grafik yang berbentuk lingkaran berwarna hitam, merupakan target dari data matriks yang dipelajari polanya.
- b. Gambar pada grafik yang berwarna merah, adalah hasil *output* pembelajaran *backpropagation* untuk batas atas regresi
- c. Gambar pada grafik yang berwarna biru, adalah hasil *output* pembelajaran *backpropagation* untuk batas bawah regresi.

Berikut adalah tabel dari hasil pembelajaran *neurofuzzy* dengan *learning rate* = 0.25 dan *momentum* = 0,9.

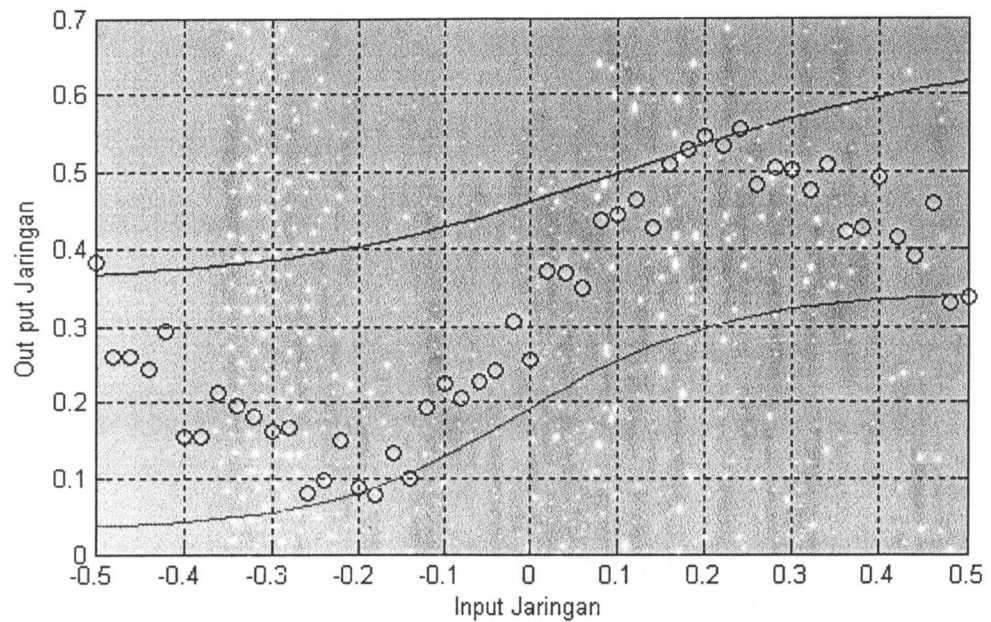
Tabel 6.2. Tabel hasil pembelajaran *Neurofuzzy* dengan *learning rate* 0.25 dan *momentum* 0,9

Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
1	0.03651	0,3846	0.37195
2	0.03737	0,2611	0.36655
3	0.03836	0,2609	0.36134
4	0.03953	0,2437	0.35636
5	0.04089	0,2954	0.35163
6	0.04248	0,1563	0.34720
7	0.04433	0,1558	0.34311
8	0.04649	0,2135	0.33939
9	0.04900	0,1964	0.33610
10	0.05191	0,1825	0.33330
11	0.05528	0,1635	0.33103
12	0.05917	0,1668	0.32936
13	0.06366	0,0806	0.32836
14	0.06880	0,0977	0.32811
15	0.07467	0,1498	0.32866
16	0.08133	0,0875	0.33011
17	0.08883	0,0773	0.33252
18	0.09720	0,1328	0.33598
19	0.10646	0,0987	0.34055
20	0.11658	0,1948	0.34629
21	0.12752	0,2238	0.35323

Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
22	0.13919	0,2049	0.36141
23	0.15146	0,2274	0.37080
24	0.16419	0,2408	0.38136
25	0.17720	0,3060	0.39302
26	0.19031	0,2555	0.40562
27	0.20332	0,3705	0.41901
28	0.21607	0,3692	0.43297
29	0.22840	0,3490	0.44726
30	0.24017	0,4365	0.46162
31	0.25128	0,4442	0.47579
32	0.26166	0,4640	0.48951
33	0.27126	0,4279	0.50257
34	0.28005	0,5096	0.51478
35	0.28806	0,5284	0.52598
36	0.29528	0,5450	0.53608
37	0.30176	0,5338	0.54501
38	0.30753	0,5553	0.55276
39	0.31265	0,4834	0.55933
40	0.31716	0,5046	0.56477
41	0.32112	0,5020	0.56913
42	0.32458	0,4739	0.57247
43	0.32759	0,5093	0.57489
44	0.33019	0,4228	0.57645
45	0.33244	0,4279	0.57725
46	0.33436	0,4927	0.57736
47	0.33601	0,4160	0.57687
48	0.33740	0,3916	0.57584
49	0.33858	0,4588	0.57434
50	0.33957	0,3300	0.57244
51	0.34039	0,3373	0.57020

b. Pembelajaran dengan $learning\ rate = 0.25$ dan $momentum = 0,5$

Gambar 6.11 adalah grafik *output* yang ditampilkan pada pembelajaran data *neurofuzzy*.



Gambar 6.11 Tampilan output dengan $lr = 0.25$ dan $mom = 0,5$

Berikut adalah tabel dari hasil pembelajaran *neurofuzzy* dengan *learning rate* = 0.25 dan *momentum* = 0,5.

Tabel 6.3. Tabel hasil pembelajaran *Neurofuzzy* dengan *learning rate* 0.25 dan *momentum* 0,5

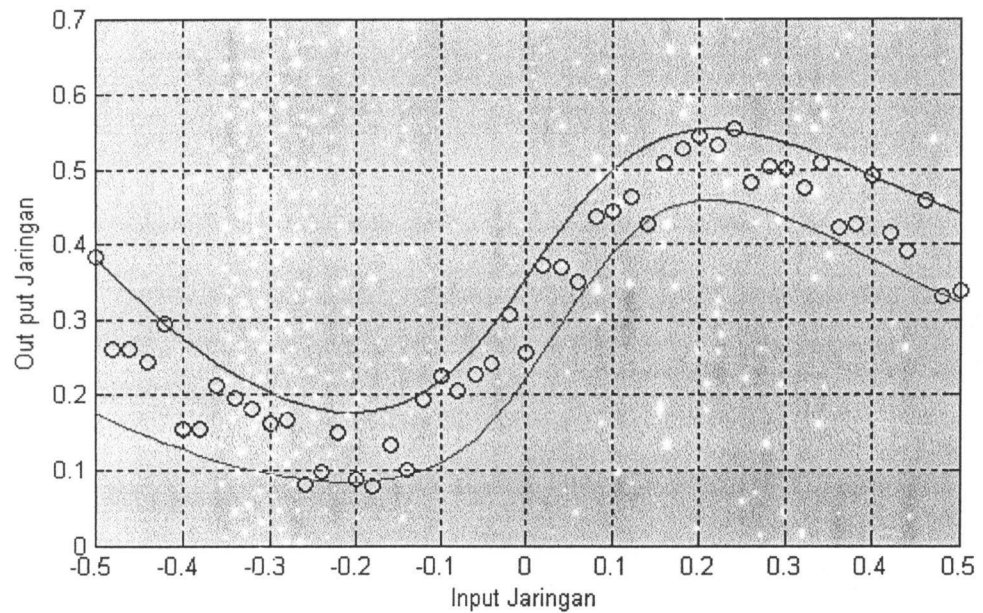
Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
1	0.03622	0,3846	0.36757
2	0.03705	0,2611	0.36869
3	0.03802	0,2609	0.36994
4	0.03915	0,2437	0.37134
5	0.04048	0,2954	0.37289
6	0.04203	0,1563	0.37461
7	0.04384	0,1558	0.37651
8	0.04596	0,2135	0.37859
9	0.04843	0,1964	0.38088
10	0.05130	0,1825	0.38339
11	0.05464	0,1635	0.38612
12	0.05851	0,1668	0.38910
13	0.06299	0,0806	0.39232

Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
14	0.06815	0,0977	0.39581
15	0.07406	0,1498	0.39958
16	0.08079	0,0875	0.40362
17	0.08839	0,0773	0.40796
18	0.09691	0,1328	0.41259
19	0.10636	0,0987	0.41752
20	0.11672	0,1948	0.42275
21	0.12794	0,2238	0.42827
22	0.13992	0,2049	0.43409
23	0.15253	0,2274	0.44019
24	0.16560	0,2408	0.44656
25	0.17896	0,3060	0.45318
26	0.19238	0,2555	0.46003
27	0.20569	0,3705	0.46708
28	0.21868	0,3692	0.47432
29	0.23119	0,3490	0.48170
30	0.24310	0,4365	0.48919
31	0.25428	0,4442	0.49676
32	0.26468	0,4640	0.50436
33	0.27424	0,4279	0.51197
34	0.28297	0,5096	0.51954
35	0.29085	0,5284	0.52703
36	0.29793	0,5450	0.53440
37	0.30424	0,5338	0.54163
38	0.30983	0,5553	0.54868
39	0.31475	0,4834	0.55553
40	0.31905	0,5046	0.56215
41	0.32280	0,5020	0.56851
42	0.32605	0,4739	0.57461
43	0.32885	0,5093	0.58043
44	0.33124	0,4228	0.58596
45	0.33328	0,4279	0.59120
46	0.33501	0,4927	0.59614
47	0.33645	0,4160	0.60079
48	0.33766	0,3916	0.60515
49	0.33865	0,4588	0.60923
50	0.33945	0,3300	0.61302
51	0.34009	0,3373	0.61655



c. Pembelajaran dengan *learning rate* = 0.50 dan *momentum* = 0,9

Gambar 6.12 adalah grafik *output* yang ditampilkan pada pembelajaran data *neurofuzzy*.



Gambar 6.12 Tampilan output dengan $lr = 0.50$ dan $mom = 0,9$

Berikut adalah tabel dari hasil pembelajaran *neurofuzzy* dengan *learning rate* = 0.50 dan *momentum* = 0,9.

Tabel 6.4. Tabel hasil pembelajaran *Neurofuzzy* dengan *learning rate* 0.50 dan *momentum* 0,9

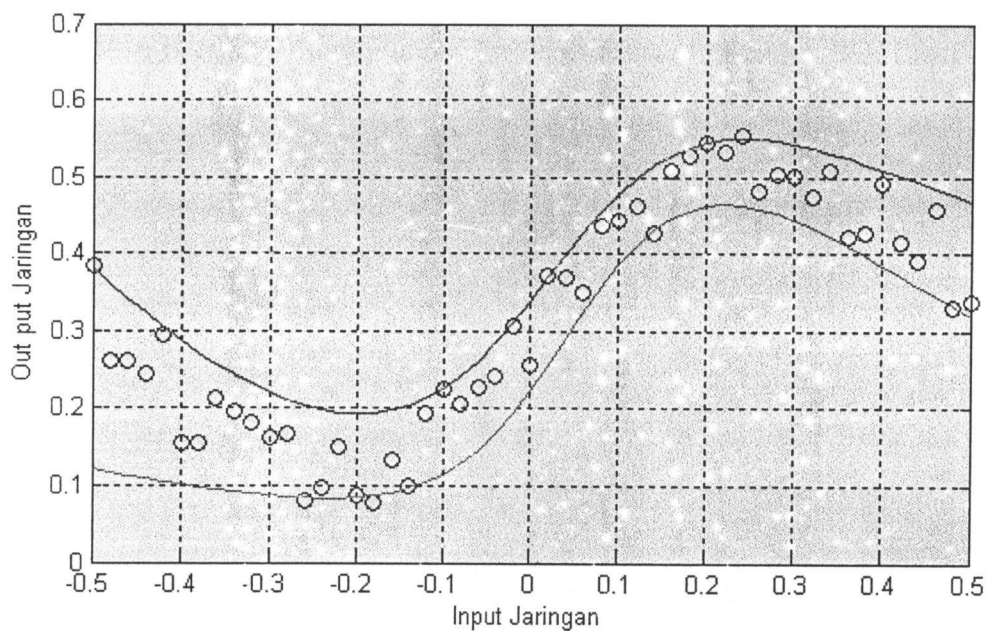
Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
1	0.17632	0,3846	0.38235
2	0.16558	0,2611	0.35936
3	0.15531	0,2609	0.33710
4	0.14555	0,2437	0.31573
5	0.13633	0,2954	0.29543

Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
6	0.12770	0,1563	0.27634
7	0.11969	0,1558	0.25860
8	0.11235	0,2135	0.24233
9	0.10569	0,1964	0.22763
10	0.09978	0,1825	0.21460
11	0.09464	0,1635	0.20333
12	0.09034	0,1668	0.19391
13	0.08692	0,0806	0.18645
14	0.08449	0,0977	0.18106
15	0.08313	0,1498	0.17789
16	0.08300	0,0875	0.17713
17	0.08429	0,0773	0.17901
18	0.08725	0,1328	0.18381
19	0.09222	0,0987	0.19186
20	0.09962	0,1948	0.20350
21	0.10996	0,2238	0.21906
22	0.12381	0,2049	0.23876
23	0.14175	0,2274	0.26259
24	0.16418	0,2408	0.29026
25	0.19119	0,3060	0.32106
26	0.22236	0,2555	0.35388
27	0.25660	0,3705	0.38732
28	0.29227	0,3692	0.41988
29	0.32744	0,3490	0.45019
30	0.36022	0,4365	0.47719
31	0.38909	0,4442	0.50019
32	0.41310	0,4640	0.51889
33	0.43186	0,4279	0.53329
34	0.44539	0,5096	0.54363
35	0.45405	0,5284	0.55027
36	0.45836	0,5450	0.55363
37	0.45891	0,5338	0.55415
38	0.45631	0,5553	0.55226
39	0.45113	0,4834	0.54836
40	0.44388	0,5046	0.54282
41	0.43503	0,5020	0.53597
42	0.42499	0,4739	0.52808
43	0.41409	0,5093	0.51940
44	0.40263	0,4228	0.51016
45	0.39084	0,4279	0.50053
46	0.37892	0,4927	0.49066
47	0.36703	0,4160	0.48069
48	0.35529	0,3916	0.47071

Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
49	0.34380	0,4588	0.46081
50	0.33263	0,3300	0.45107
51	0.32182	0,3373	0.44153

d. Pembelajaran dengan *learning rate* = 0.50 dan momentum = 0,5

Gambar 6.13 adalah grafik *output* yang ditampilkan pada pembelajaran data *neurofuzzy*.



Gambar 6.13 Tampilan output dengan $lr = 0.50$ dan $mom = 0,5$

Berikut adalah tabel dari hasil pembelajaran *neurofuzzy* dengan *learning rate* = 0.50 dan momentum = 0,5.

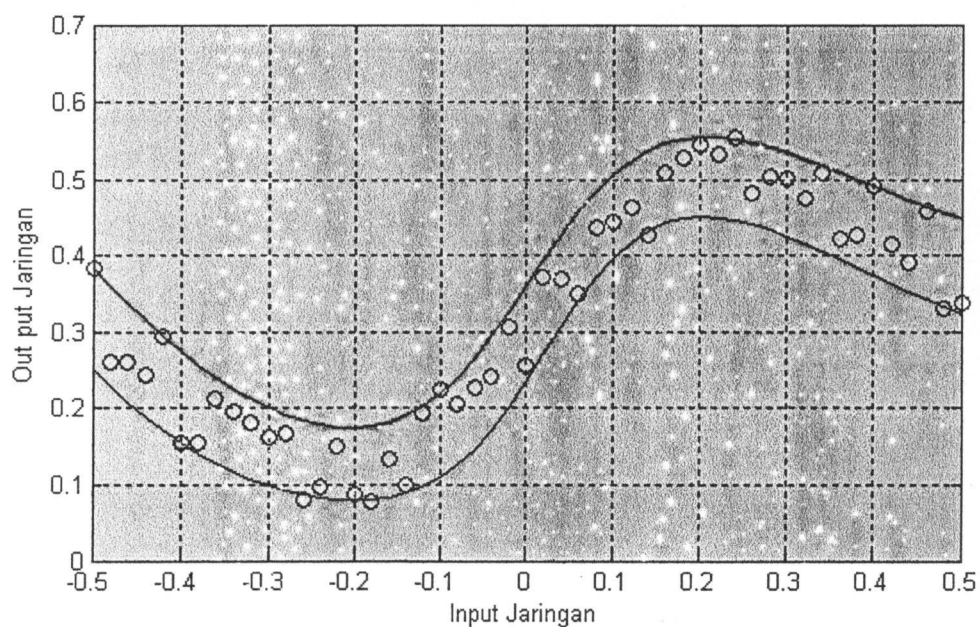
Tabel 6.5. Tabel hasil pembelajaran *Neurofuzzy* dengan *learning rate* 0.50 dan *momentum* 0,5

Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
1	0.12104	0,3846	0.38073
2	0.11692	0,2611	0.36073
3	0.11288	0,2609	0.34128
4	0.10894	0,2437	0.32251
5	0.10513	0,2954	0.30456
6	0.10149	0,1563	0.28756
7	0.09805	0,1558	0.27161
8	0.09484	0,2135	0.25684
9	0.09191	0,1964	0.24333
10	0.08932	0,1825	0.23121
11	0.08711	0,1635	0.22055
12	0.08538	0,1668	0.21146
13	0.08420	0,0806	0.20406
14	0.08367	0,0977	0.19845
15	0.08394	0,1498	0.19478
16	0.08517	0,0875	0.19323
17	0.08756	0,0773	0.19398
18	0.09139	0,1328	0.19728
19	0.09697	0,0987	0.20338
20	0.10472	0,1948	0.21257
21	0.11511	0,2238	0.22512
22	0.12866	0,2049	0.24123
23	0.14590	0,2274	0.26096
24	0.16722	0,2408	0.28419
25	0.19277	0,3060	0.31047
26	0.22227	0,2555	0.33908
27	0.25489	0,3705	0.36898
28	0.28927	0,3692	0.39900
29	0.32369	0,3490	0.42793
30	0.35641	0,4365	0.45470
31	0.38589	0,4442	0.47852
32	0.41106	0,4640	0.49889
33	0.43133	0,4279	0.51560
34	0.44657	0,5096	0.52868
35	0.45697	0,5284	0.53832
36	0.46291	0,5450	0.54482
37	0.46491	0,5338	0.54853
38	0.46350	0,5553	0.54981
39	0.45922	0,4834	0.54902

Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
40	0.45256	0,5046	0.54650
41	0.44398	0,5020	0.54253
42	0.43389	0,4739	0.53739
43	0.42264	0,5093	0.53132
44	0.41054	0,4228	0.52452
45	0.39786	0,4279	0.51716
46	0.38482	0,4927	0.50939
47	0.37162	0,4160	0.50135
48	0.35839	0,3916	0.49313
49	0.34528	0,4588	0.48482
50	0.33239	0,3300	0.47650
51	0.31979	0,3373	0.46822

e. Pembelajaran dengan $learning\ rate = 0.75$ dan $momentum = 0,9$

Gambar 6.14 adalah grafik *output* yang ditampilkan pada pembelajaran data *neurofuzzy*.



Gambar 6.14 Tampilan output dengan $lr = 0.75$ dan $mom = 0,9$

Berikut adalah tabel dari hasil pembelajaran *neurofuzzy* dengan *learning rate* = 0.75 dan *momentum* = 0,9.

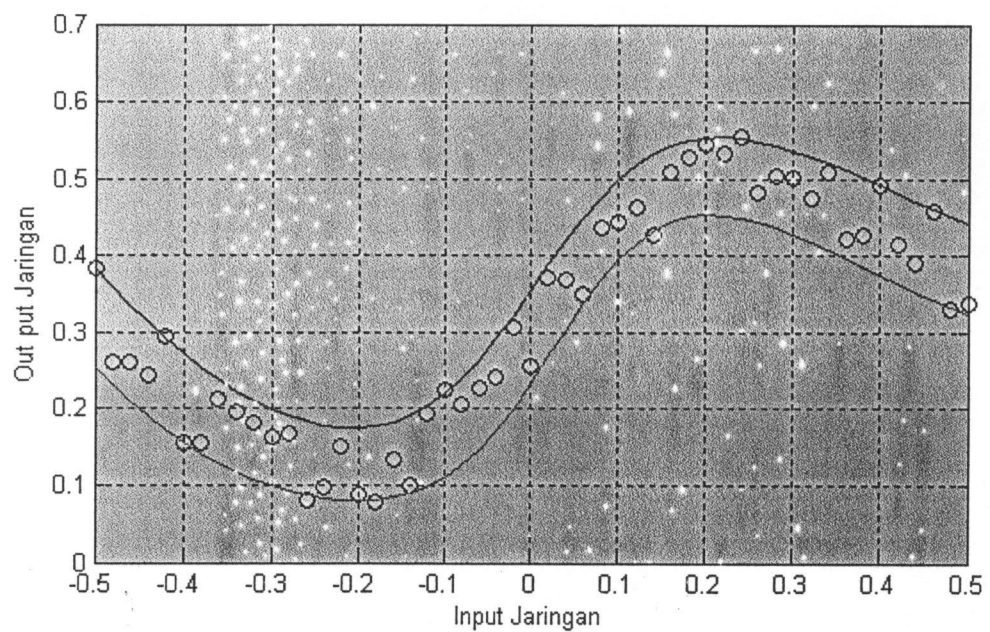
Tabel 6.6. Tabel hasil pembelajaran *Neurofuzzy* dengan *learning rate* 0.75 dan *momentum* 0,9

Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
1	0.25166	0,3846	0.38269
2	0.22975	0,2611	0.35967
3	0.20916	0,2609	0.33723
4	0.19000	0,2437	0.31559
5	0.17235	0,2954	0.29494
6	0.15627	0,1563	0.27546
7	0.14178	0,1558	0.25732
8	0.12887	0,2135	0.24065
9	0.11752	0,1964	0.22560
10	0.10770	0,1825	0.21226
11	0.09937	0,1635	0.20074
12	0.09252	0,1668	0.19114
13	0.08712	0,0806	0.18357
14	0.08319	0,0977	0.17814
15	0.08078	0,1498	0.17503
16	0.07998	0,0875	0.17441
17	0.08096	0,0773	0.17654
18	0.08399	0,1328	0.18172
19	0.08941	0,0987	0.19029
20	0.09769	0,1948	0.20263
21	0.10940	0,2238	0.21907
22	0.12516	0,2049	0.23982
23	0.14551	0,2274	0.26487
24	0.17072	0,2408	0.29382
25	0.20058	0,3060	0.32586
26	0.23420	0,2555	0.35974
27	0.27002	0,3705	0.39393
28	0.30604	0,3692	0.42685
29	0.34020	0,3490	0.45712
30	0.37078	0,4365	0.48370
31	0.39662	0,4442	0.50599
32	0.41720	0,4640	0.52379
33	0.43250	0,4279	0.53719
34	0.44284	0,5096	0.54650
35	0.44875	0,5284	0.55214

Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
36	0.45085	0,5450	0.55459
37	0.44977	0,5338	0.55431
38	0.44609	0,5553	0.55176
39	0.44034	0,4834	0.54735
40	0.43300	0,5046	0.54148
41	0.42445	0,5020	0.53445
42	0.41504	0,4739	0.52658
43	0.40503	0,5093	0.51810
44	0.39466	0,4228	0.50923
45	0.38411	0,4279	0.50015
46	0.37351	0,4927	0.49100
47	0.36299	0,4160	0.48191
48	0.35262	0,3916	0.47296
49	0.34248	0,4588	0.46424
50	0.33261	0,3300	0.45580
51	0.32305	0,3373	0.44768

f. Pembelajaran dengan *learning rate* = 0.75 dan momentum = 0,5

Gambar 6.15 adalah grafik *output* yang ditampilkan pada pembelajaran data neurofuzzy.



Gambar 6.15 Tampilan output dengan $lr = 0.75$ dan $mom = 0,5$

Berikut adalah tabel dari hasil pembelajaran *neurofuzzy* dengan *learning rate* = 0.75 dan *momentum* = 0,5.

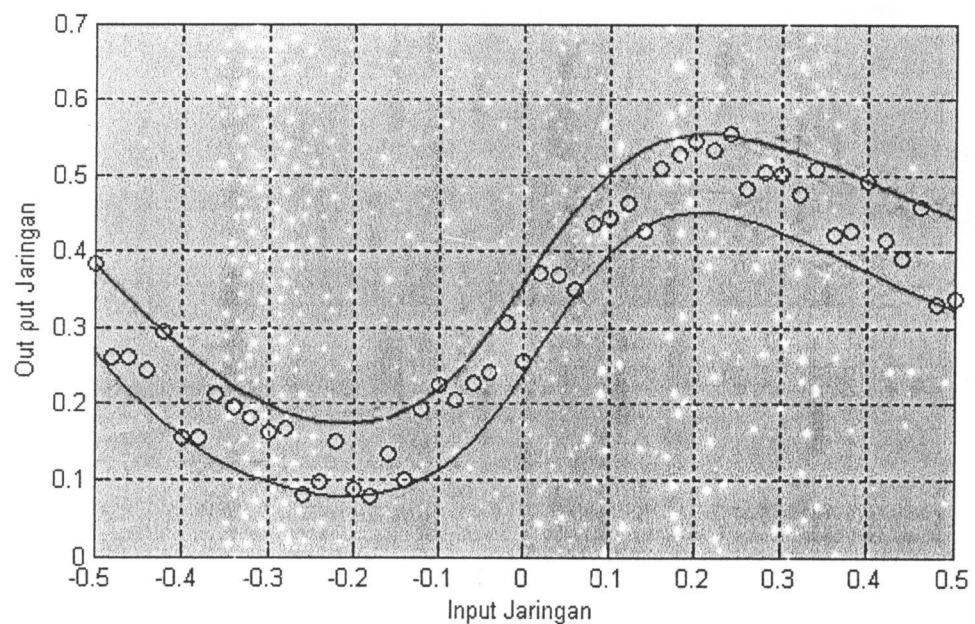
Tabel 6.7. Tabel hasil pembelajaran *Neurofuzzy* dengan *learning rate* 0.75 dan *momentum* 0,5

Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
1	0.25453	0,3846	0.38449
2	0.23180	0,2611	0.36011
3	0.21052	0,2609	0.33661
4	0.19082	0,2437	0.31418
5	0.17274	0,2954	0.29299
6	0.15634	0,1563	0.27320
7	0.14162	0,1558	0.25494
8	0.12856	0,2135	0.23833
9	0.11712	0,1964	0.22346
10	0.10727	0,1825	0.21040
11	0.09897	0,1635	0.19924
12	0.09216	0,1668	0.19005
13	0.08684	0,0806	0.18291
14	0.08300	0,0977	0.17792
15	0.08070	0,1498	0.17522
16	0.08001	0,0875	0.17499
17	0.08111	0,0773	0.17742
18	0.08422	0,1328	0.18280
19	0.08970	0,0987	0.19142
20	0.09798	0,1948	0.20361
21	0.10959	0,2238	0.21964
22	0.12512	0,2049	0.23969
23	0.14507	0,2274	0.26372
24	0.16972	0,2408	0.29139
25	0.19890	0,3060	0.32198
26	0.23184	0,2555	0.35442
27	0.26711	0,3705	0.38737
28	0.30285	0,3692	0.41941
29	0.33709	0,3490	0.44927
30	0.36809	0,4365	0.47592
31	0.39464	0,4442	0.49873
32	0.41609	0,4640	0.51738
33	0.43230	0,4279	0.53187
34	0.44349	0,5096	0.54240
35	0.45013	0,5284	0.54928

Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
36	0.45279	0,5450	0.55292
37	0.45209	0,5338	0.55372
38	0.44860	0,5553	0.55209
39	0.44289	0,4834	0.54843
40	0.43544	0,5046	0.54308
41	0.42667	0,5020	0.53636
42	0.41696	0,4739	0.52855
43	0.40662	0,5093	0.51991
44	0.39589	0,4228	0.51065
45	0.38499	0,4279	0.50096
46	0.37409	0,4927	0.49099
47	0.36330	0,4160	0.48089
48	0.35274	0,3916	0.47075
49	0.34247	0,4588	0.46068
50	0.33255	0,3300	0.45074
51	0.32301	0,3373	0.44099

g. Pembelajaran dengan *learning rate* = 1 dan momentum = 0,9

Gambar 6.16 adalah grafik *output* yang ditampilkan pada pembelajaran data neurofuzzy.



Gambar 6.16 Tampilan output dengan $lr = 1$ dan $mom = 0,9$

Berikut adalah tabel dari hasil pembelajaran *neuro-fuzzy* dengan *learning rate* =1 dan momentum = 0,9 .

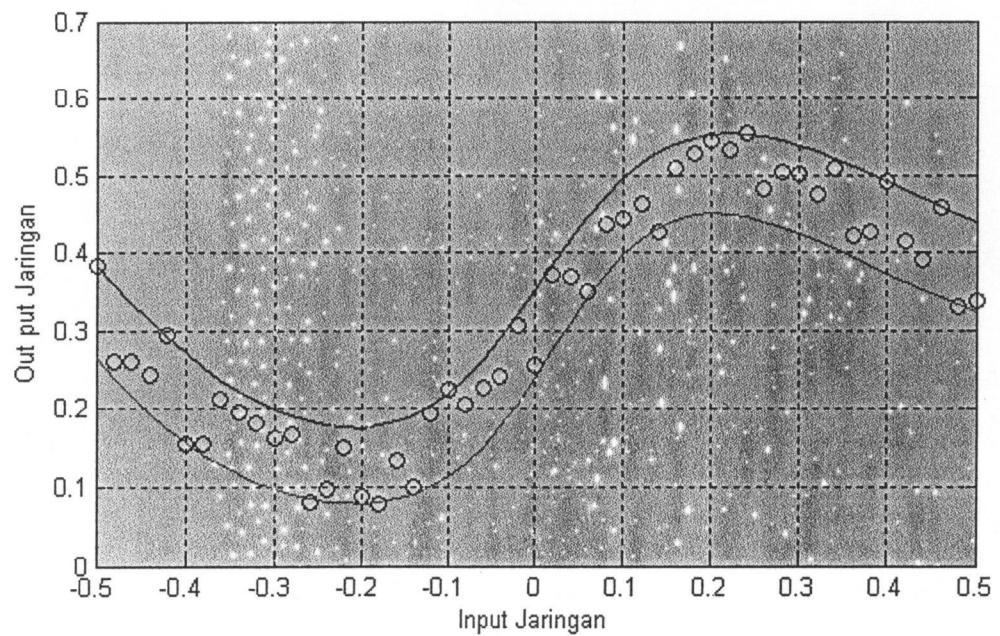
Tabel 6.8. Tabel hasil pembelajaran *Neurofuzzy* dengan *learning rate* 1 dan momentum 0,9

Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
1	0.25888	0,3846	0.38447
2	0.23560	0,2611	0.36017
3	0.21369	0,2609	0.33667
4	0.19327	0,2437	0.31419
5	0.17446	0,2954	0.29292
6	0.15735	0,1563	0.27302
7	0.14196	0,1558	0.25465
8	0.12830	0,2135	0.23793
9	0.11637	0,1964	0.22295
10	0.10612	0,1825	0.20981
11	0.09753	0,1635	0.19858
12	0.09055	0,1668	0.18933
13	0.08518	0,0806	0.18216
14	0.08142	0,0977	0.17717
15	0.07931	0,1498	0.17449
16	0.07897	0,0875	0.17429
17	0.08056	0,0773	0.17679
18	0.08438	0,1328	0.18226
19	0.09079	0,0987	0.19100
20	0.10028	0,1948	0.20334
21	0.11341	0,2238	0.21956
22	0.13075	0,2049	0.23982
23	0.15274	0,2274	0.26408
24	0.17944	0,2408	0.29198
25	0.21042	0,3060	0.32278
26	0.24454	0,2555	0.35537
27	0.28011	0,3705	0.38840
28	0.31516	0,3692	0.42045
29	0.34776	0,3490	0.45023
30	0.37645	0,4365	0.47677
31	0.40028	0,4442	0.49941
32	0.41890	0,4640	0.51790
33	0.43240	0,4279	0.53223
34	0.44116	0,5096	0.54261
35	0.44572	0,5284	0.54938

Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
36	0.44671	0,5450	0.55293
37	0.44474	0,5338	0.55368
38	0.44039	0,5553	0.55204
39	0.43418	0,4834	0.54837
40	0.42656	0,5046	0.54304
41	0.41795	0,5020	0.53635
42	0.40865	0,4739	0.52857
43	0.39894	0,5093	0.51996
44	0.38904	0,4228	0.51071
45	0.37913	0,4279	0.50102
46	0.36933	0,4927	0.49103
47	0.35976	0,4160	0.48087
48	0.35047	0,3916	0.47065
49	0.34153	0,4588	0.46047
50	0.33298	0,3300	0.45039
51	0.32482	0,3373	0.44047

h. Pembelajaran dengan *learning rate* = 1 dan *momentum* = 0,5

Gambar 6.17 adalah grafik *output* yang ditampilkan pada pembelajaran data *neurofuzzy*.



Gambar 6.17 Tampilan output dengan $lr = 1$ dan $mom = 0,5$

Berikut adalah tabel dari hasil pembelajaran *neurofuzzy* dengan *learning rate* = 1 dan *momentum* = 0,5.

Tabel 6.9. Tabel hasil pembelajaran *Neurofuzzy* dengan *learning rate* 1 dan *momentum* 0,5

Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
1	0.26554	0,3846	0.38474
2	0.24020	0,2611	0.36027
3	0.21658	0,2609	0.33667
4	0.19484	0,2437	0.31413
5	0.17504	0,2954	0.29286
6	0.15724	0,1563	0.27301
7	0.14141	0,1558	0.25474
8	0.12753	0,2135	0.23814
9	0.11553	0,1964	0.22332
10	0.10533	0,1825	0.21035
11	0.09687	0,1635	0.19930
12	0.09007	0,1668	0.19023
13	0.08490	0,0806	0.18321
14	0.08133	0,0977	0.17834
15	0.07941	0,1498	0.17575
16	0.07921	0,0875	0.17558
17	0.08091	0,0773	0.17806
18	0.08475	0,1328	0.18342
19	0.09108	0,0987	0.19196
20	0.10036	0,1948	0.20399
21	0.11311	0,2238	0.21976
22	0.12986	0,2049	0.23946
23	0.15103	0,2274	0.26304
24	0.17675	0,2408	0.29019
25	0.20668	0,3060	0.32025
26	0.23987	0,2555	0.35219
27	0.27482	0,3705	0.38474
28	0.30970	0,3692	0.41653
29	0.34266	0,3490	0.44631
30	0.37217	0,4365	0.47307
31	0.39720	0,4442	0.49612
32	0.41725	0,4640	0.51514
33	0.43225	0,4279	0.53007
34	0.44248	0,5096	0.54107
35	0.44839	0,5284	0.54843

Data ke-	Batas bawah interval $h^-(x)$	Target output $d(x)$	Batas atas interval $h^+(x)$
36	0.45053	0,5450	0.55252
37	0.44948	0,5338	0.55372
38	0.44582	0,5553	0.55244
39	0.44004	0,4834	0.54904
40	0.43263	0,5046	0.54386
41	0.42399	0,5020	0.53723
42	0.41447	0,4739	0.52941
43	0.40436	0,5093	0.52067
44	0.39391	0,4228	0.51121
45	0.38331	0,4279	0.50124
46	0.37273	0,4927	0.49090
47	0.36230	0,4160	0.48035
48	0.35209	0,3916	0.46970
49	0.34220	0,4588	0.45906
50	0.33266	0,3300	0.44850
51	0.32351	0,3373	0.43809



BAB VII

PENUTUP

7.1 Kesimpulan

Dari hasil penelitian, analisis, perancangan sistem, pembuatan program sampai tahap penyelesaian program, maka penulis dapat mengambil kesimpulan sebagai berikut :

- a. Dengan menggunakan pembelajaran *neurofuzzy* dapat menyelesaikan sebuah model regresi.
- b. Perubahan pada learning rate dan momentum dapat mengubah hasil pembelajaran.
- c. Perubahan pada momentum hanya berpengaruh sedikit pada hasil pembelajaran, tapi juga dapat mempercepat pembelajaran jaringan.
- d. Perangkat lunak ini bertujuan untuk mencari batas bawah dan batas atas dari suatu model regresi.
- e. File hasil pembelajaran *neurofuzzy* dapat disimpan kedalam format *file.txt*.

7.2 Saran

Berdasarkan pada pengujian yang telah dilakukan pada perangkat lunak yang dibuat, masih banyak kekurangan dan kelemahan sehingga perlu dikembangkan lagi agar kinerjanya lebih baik, oleh karena itu disarankan :

- a. Dalam sistem ini, input data matriks merupakan data yang sudah diolah. Sehingga diharapkan dalam pengembangan program selanjutnya input data tidak harus selalu data yang sudah diolah.
- b. Jumlah lapisan *hidden* hanya terdiri dari satu lapisan saja. Diharapkan untuk pengembangan program berikutnya jumlah lapisan *hidden* bisa fleksibel sesuai keinginan user.
- c. Jumlah unit pada lapisan *output* hanya terdiri dari satu unit saja, diharapkan untuk pengembangan selanjutnya jumlah unit *outputnya* dapat dibuat sesuai dengan *input* dari *user*.
- d. Fungsi aktivasi pada pembelajaran *backpropagation* di sistem ini hanya menggunakan satu fungsi pembelajaran saja, yaitu fungsi *sigmoid biner*. Diharapkan untuk pengembangan selanjutnya fungsi pembelajaran yang dibuat terdiri dari bermacam-macam fungsi pembelajaran sesuai dengan kebutuhan user.
- e. Tampilan untuk pengembangan berikutnya diharapkan dapat lebih bagus dan lebih jelas dalam menampilkan hasil pembelajaran *neuro-fuzzy*.

DAFTAR PUSTAKA

- [AWA06] Away, Gunaidi Abdia. *The Shortcut of Matlab Programming*. Bandung : Informatika. 2006
- [ERW04] Erwin, M.AH. *Diktat Kuliah Jaringan Saraf Tiruan*. Yogyakarta. 2004.
- [JAN97] Jang, JSR; Sun, CT; dan Mizutani, E. *Neuro-Fuzzy and Soft Computing*. London : Prentice-Hall. 1997.
- [KUS02] Kusumadewi, Sri. *Analisis & Desain Sistem Fuzzy Menggunakan Tool Box Matlab*. Yogyakarta : Graha Ilmu. 2002.
- [KUS03] Kusumadewi, Sri. *Artificial Intelligence*. Yogyakarta : Graha Ilmu. 2003.
- [KUS04] Kusumadewi, Sri. *Membangun Jaringan Saraf Tiruan Menggunakan Matlab & Excel Link*. Yogyakarta : Graha Ilmu. 2004.
- [LIN96] Lin, Ching-Teng; dan Lee, George. *Neural Fuzzy System*. London: Prentice-Hall. 1996.
- [WID05] Widodo, Thomas Sri. *Sistem Neuro Fuzzy untuk Pengolahan Informasi, Pemodelan, dan Kendali*. Yogyakarta : Graha Ilmu. 2005.