

**SIMULASI JARINGAN SYARAF TIRUAN
BERBASIS METODE *BACKPROPAGATION* SEBAGAI
PENGENDALI SUHU SHOWER**

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Teknik Elektro



Disusun oleh :

Nama : Asri Nurkomala

No.Mahasiswa : 02 524 085

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2007

LEMBAR PENGESAHAN PEMBIMBING

SIMULASI JARINGAN SYARAF TIRUAN BERBASIS METODE *BACKPROPAGATION* SEBAGAI PENGENDALI SUHU SHOWER

TUGAS AKHIR

Disusun oleh :

Nama : Asri Nurkomala

No. Mahasiswa : 02 524 085

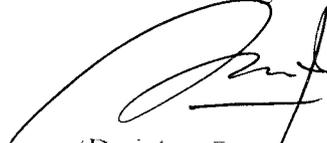
Yogyakarta, Mei 2007

Pembimbing I



(Ir. Hj. Budi Astuti, MT)

Pembimbing II



(Dwi Ana Ratna Wati, ST)

LEMBAR PENGESAHAN PENGUJI
SIMULASI JARINGAN SYARAF TIRUAN
BERBASIS METODE *BACKPROPAGATION* SEBAGAI
PENGENDALIAN SUHU SHOWER

TUGAS AKHIR

Disusun oleh :

Nama : Asri Nurkomala

No.Mahasiswa : 02 524 085

**Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat untuk
Memperoleh Gelar sarjana Jurusan Teknik Elektro Fakultas Teknologi Industri
Universitas Islam Indonesia**

Yogyakarta, 30 Mei 2007

Tim Penguji,

Ir. Hj. Budi Astuti, MT.

Ketua

Dwi Ana Ratna Wati, ST

Anggota I

Hendra Setiawan, ST., MT

Anggota II



Mengetahui,

Jurusan Teknik Elektro

Fakultas teknologi Industri

Universitas Islam Indonesia



(Dwono, ST., M.Sc)

MOTTO

“ Sungguh bersama kesukaran pasti ada kemudahan
Dan bersama kesukaran pasti ada kemudahan
Karena itu, bila selesai suatu tugas, mulailah tugas yang lain
Dengan sungguh – sungguh
Hanya kepada Tuhanmu hendaknya kau berharap “
(Asy Syarh (94) : 5 – 8)

Dosa terbesar adalah ketakutan
Rekreasi terbaik adalah bekerja
Musibah terbesar adalah keputusasaan
Keberanian terbesar adalah kesabaran
Guru terbaik adalah pengalaman
Misteri terbesar adalah kematian
Kehormatan terbesar adalah kesetiaan
Karunia terbesar adalah anak shaleh
Sumbangan terbesar adalah partisipasi
Modal terbesar adalah kemandirian

KATA PENGANTAR



Assalamu'alaikum Wr. Wb.

Puji syukur kehadiran Allah SWT, atas nikmat iman, rahmat, hidayah dan pikiran yang diberikan. Sehingga penulis dapat menyelesaikan tugas akhir dengan judul “**Simulasi jaringan syaraf tiruan berbasis metode *backpropagation* sebagai pengendali suhu shower**”. Tidak lupa shalawat serta salam selalu tercurah kepada Nabi Muhammad. SAW beserta keluarga dan para sahabatnya.

Adapun maksud dari penyusunan tugas akhir ini adalah untuk memenuhi kurikulum S-1 Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia. Disamping itu untuk menambah pengetahuan terhadap ilmu yang telah dipelajari di bangku perkuliahan untuk diterapkan ke masyarakat.

Dalam penyusunan ini, penulis banyak mendapat bantuan dari berbagai pihak, sehingga penulis ingin menyampaikan ucapan terima kasih kepada :

1. Kedua orang tuaku yang senantiasa memberikan dukungan moril, materi dan doa setiap saat.
2. Bpk Fathul Wahid, selaku Dekan Fakultas Teknologi Industri (FTI) Universitas Islam Indonesia (UII)
3. Bpk Tito Yuwono, ST. MT. selaku Kajur Teknik Elektro
4. Ibu Hj. Ir Budi Astuti, ST. MT. selaku dosen pembimbing I
5. Ibu Dwi Ana Ratna Wati, ST. selaku dosen pembimbing II atas waktu, kebaikan, kesabaran dan ilmunya.
6. Jhon Hendra, ST. yang telah banyak membantu.
7. Dosen dan karyawan Fakultas Teknologi Industri UII, Ka.Lab dan laboran jurusan Teknik Elektro atas waktu, tempat dan ilmu yang diberikan.
8. Seluruh mahasiswa jurusan Teknik Elektro UII, Asisten Lab Instalasi dan Mesin Listrik Dasar (imel) dan Asisten Lab Telekomunikasi.

9. Seluruh pihak yang tidak dapat di sebutkan satu-persatu, yang telah memberikan *support* dan doa.

Penulis menyadari bahwa Tugas Akhir ini masih terdapat kesalahan dan kekurangan. Oleh karena itu, kritik dan saran yang membangun akan senantiasa penulis terima dengan senang hati. Akhirnya, harapan penulis semoga Tugas Akhir ini dapat bermanfaat bagi kita semua. Amiin...

Wassalamu'alaikum Wr.Wb

Jogjakarta, Mei 2007

Asri Nurkomala

ABSTRAKSI

Untuk menjaga kestabilan dalam menangani pengendalian suhu tidak lagi memerlukan campur tangan untuk menaikkan dan menurunkan suhu. Tetapi peran itu telah menjadi bagian kerja sistem pengendali secara otomatis. Walaupun ada peran manusia lebih pada setting kondisi yang diperlukan. Kemudahan perancangan sistem ini didukung dengan penemuan – penemuan baru di bidang *Artificial Intelligence* yang banyak memberikan ide terobosan untuk merancang sistem pengendalian yang lebih efektif, stabil dan lebih mudah pengaturannya. Jaringan Syaraf Tiruan merupakan salah satu kendali suhu shower yang dapat disimulasikan menggunakan *neural network toolbox* pada *software* Matlab 6.5.

Dengan menggunakan metode *Backpropagation* dan fungsi *Algoritma Levenberg-Marquardt* diperoleh struktur jaringan yang terbaik, untuk input yang digeser terdiri dari 13 sel neuron lapisan *input*, 10 sel neuron lapisan tersembunyi dan 2 sel neuron lapisan *output*. Fungsi aktivasi pada setiap lapisan menggunakan fungsi *sigmoid bipolar*, dengan *learning rate* 0.4 dan menghasilkan *Mean Square Error* (MSE) 0.0728086.,persentase MSE pengujian adalah 0.44 dan untuk input yang tidak digeser diperoleh struktur jaringan terbaik terdiri dari 10 sel neuron lapisan *input*, 8 sel neuron lapisan tersembunyi dan 2 sel neuron lapisan *output*. Fungsi aktivasi pada setiap lapisan menggunakan fungsi *sigmoid bipolar*, dengan *learning rate* 0.6 dan menghasilkan *Mean Square Error* 0.0741973, persentase MSE pengujian adalah 0.12

Banyaknya jumlah data masukan berpengaruh terhadap banyaknya iterasi dan *mean square error* yang dihasilkan. Penelitian ini juga membuktikan bahwa dasar teori tentang pengaturan suhu shower belum bisa optimal dengan *Artificial Intelligence* menggunakan *Neural Network* (Jaringan Syaraf Tiruan).

DAFTAR ISI

Halaman Judul.....	i
Lembar Pengesahan Pembimbing.....	ii
Lembar Pengesahan Penguji.....	iii
Halaman Persembahan.....	iv
Kata Pengantar.....	v
Abstraksi.....	vi
Daftar Isi.....	vii
Daftar Gambar.....	x
Daftar Tabel.....	xiii

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Metodologi Penelitian.....	4
1.6 Sistematika Penulisan.....	4

BAB II LANDASAN TEORI

2.1 Sistem Jaringan Syaraf Manusia.....	6
2.1.1 Anatomi Sel Syaraf Biologis.....	7
2.1.2 Cara Kerja Sel Syaraf Biologis.....	8
2.2 Sistem Jaringan Syaraf Tiruan.....	9
2.2.1 Jaringan Syaraf Manusia sebagai Dasar Jaringan Syaraf Tiruan.....	11
2.2.2 Cara Kerja Komponen Jaringan Syaraf Tiruan	12
2.2.3 Konsep Belajar Jaringan Syaraf Tiruan.....	14
2.2.4 Arsitektur Jaringan Syaraf Tiruan.....	15
2.2.5 Fungsi Aktivasi.....	17
2.2.6 Metode <i>Backpropagation</i>	18
2.2.6.1 Arsitektur <i>Backpropagation</i>	19
2.2.6.2 Algoritma <i>Backpropagation</i>	20

2.2.7	<i>Algoritma Levenberg Marquardt</i>	22
2.3	Karakteristik <i>Plant</i>	23
2.3.1	Prinsip Shower.....	24
BAB III PERANCANGAN SISTEM		
3.1	Perancangan Jaringan Syaraf Tiruan.....	29
3.1.1	Prosedur Pelatihan.....	30
3.1.2	Prosedur Pengujian.....	33
3.1.3	Pemrograman Jaringan dengan <i>Prosedure</i> dan <i>Funcion</i>	34
3.2	Diagram Alir Jaringan Syaraf Tiruan Metode <i>Backpropagation</i>	36
3.3	Simulasi Sitem Kendali Suhu Shower.....	40
3.4	Perancangan Simulasi pada <i>Plant</i>	43
3.4.1	Gambaran Fisik <i>Plant</i>	44
3.4.2	Gambaran <i>Plant</i> pada simulasi.....	44
3.4.2.1	Katup.....	44
3.4.2.2	Flow Rate.....	45
3.4.2.3	Temperatur.....	46
BAB IV ANALISA DAN PEMBAHASAN		
4.1	Pelatihan Jaringan Syaraf Tiruan <i>Backpropagation</i>	48
4.1.1	Pelatihan Menggunakan 1 Lapisan Tersembunyi.....	49
4.1.1.1	Pelatihan input yang digeser.....	55
4.1.1.2	Pelatihan input yang tidak digeser.....	60
4.2	Pengujian Jaringan Syaraf Tiruan <i>Backpropagation</i>	68
BAB V KESIMPULAN DAN SARAN		
5.1	Kesimpulan.....	79
5.2	Saran.....	80
DAFTAR PUSTAKA		

DAFTAR GAMBAR

Gambar 2.1 Sistem Jaringan Syaraf Manusia.....	9
Gambar 2.2 Struktur Neuron Jaringan Syaraf Tiruan.....	12
Gambar 2.3 Jaringan Syaraf dengan Tiga Lapisan.....	13
Gambar 2.4 Jaringan Syaraf dengan Lapisan Tunggal.....	15
Gambar 2.5 Jaringan Syaraf dengan Lapisan Banyak.....	16
Gambar 2.6 Jaringan Syaraf dengan Lapisan Kompetitif.....	16
Gambar 2.7 Fungsi aktivasi <i>Purelin</i> (identitas).....	17
Gambar 2.8 Fungsi aktivasi <i>logsig</i> (<i>sigmoid</i> biner).....	18
Gambar 2.9 Fungsi aktivasi <i>tansig</i> (<i>sigmoid</i> bipolar).....	18
Gambar 2.10 Arsitektur jaringan sederhana.....	19
Gambar 2.11 Arsitektur algoritma <i>backpropagation</i>	20
Gambar 2.12 Model shower.....	24
Gambar 3.1 Blok diagram perancangan system.....	27
Gambar 3.2 Model <i>invers</i> pelatihan.....	28
Gambar 3.3 Arsitektur jaringan syaraf tiruan.....	29
Gambar 3.4 Proses pembangunan jaringan syaraf tiruan.....	29
Gambar 3.5 Diagram alir / <i>flowchart</i> Prosedur pelatihan.....	38
Gambar 3.6 Diagram alir / <i>flowchart</i> prosedur pengujian.....	40
Gambar 3.8 Rangkaian blok simulasi utama.....	44
Gambar 3.9 Blok katup pada simulasi.....	45
Gambar 4.1 Hasil pelatihan dengan 1 lapisan tersembunyi, 5 sel neuron, <i>learning rate</i> 0.7 dengan input digeser.....	50

Gambar 4.2 Hasil pelatihan dengan 1 lapisan tersembunyi, 7 sel neuron, <i>learning rate</i> 0.5 dengan input digeser.....	51
Gambar 4.3 Hasil pelatihan dengan 1 lapisan tersembunyi, 7 sel neuron, <i>learning rate</i> 0.1 sampai 0.1 dengan input digeser.....	52
Gambar 4.4 Hasil pelatihan dengan 1 lapisan tersembunyi, 5 sel neuron, <i>learning rate</i> 0.7 dengan input tidak digeser.....	53
Gambar 4.5 Hasil pelatihan dengan 1 lapisan tersembunyi, 7 sel neuron, <i>learning rate</i> 0.5 dengan input tidak digeser.....	54
Gambar 4.6 Hasil pelatihan dengan 1 lapisan tersembunyi, 7 sel neuron, <i>learning rate</i> 0.4 dengan input tidak digeser.....	55
Gambar 4.7 Hasil pelatihan dengan 2 lapisan tersembunyi, 13-10 sel neuron, <i>learning rate</i> 0.4 dengan input digeser.....	56
Gambar 4.8 Hasil pelatihan dengan 2 lapisan tersembunyi, 13-10 sel neuron, <i>learning rate</i> 0.4 dengan input digeser.....	57
Gambar 4.9 Hasil pelatihan dengan 2 lapisan tersembunyi, 13 - 10 sel neuron, <i>learning rate</i> 0.4 dengan input digeser.....	58
Gambar 4.10 Hasil pelatihan dengan 4 lapisan tersembunyi, 7 – 7 – 5 - 4 sel neuron, <i>learning rate</i> 0.6 dengan input digeser.....	60
Gambar 4.11 Hasil pelatihan dengan 2 lapisan tersembunyi, 10 - 8 sel neuron, <i>learning rate</i> 0.6 dengan input tidak digeser.....	61
Gambar 4.12 Hasil pelatihan dengan 2 lapisan tersembunyi, 10 – 8 sel neuron, <i>learning rate</i> 0.6 dengan input tidak digeser.....	62

Gambar 4.13 Hasil pelatihan dengan 2 lapisan tersembunyi, 10 – 8 sel neuron, <i>learning rate</i> 0.6 dengan input tidak digeser....	63
Gambar 4.14 Hasil pelatihan dengan 3 lapisan tersembunyi, 10 – 5 – 3 sel neuron, <i>learning rate</i> 0.4 dengan input tidak digeser.	64
Gambar 4.15 Hasil pengujian jaringn syaraf tiruan untuk input yang digeser....	69
Gambar 4.16 Hasil pengujian jaringn syaraf tiruan untuk input yang tidak digeser.....	69
Gambar 4.19 Hasil pengujian menggunakan <i>step input</i> yang digeser.....	70
Gambar 4.20 Hasil pengujian menggunakan <i>step input</i> yang tidak digeser.....	72
Gambar 4.21 Tampilan animasi suhu untuk input yang digeser.....	74
Gambar 4.22 Tampilan animasi suhu untuk input yang tidak digeser.....	74
Gambar 4.23 Tampilan pengujian menggunakan GUI.....	75

DAFTAR TABEL

Tabel 4.1 Hasil pelatihan dengan 5 neuron dengan berbagai macam fungsi aktivasi	50
Tabel 4.2 Hasil pelatihan dengan 7 sel neuron dengan berbagai fungsi aktivasi	51
Tabel 4.3 Hasil pelatihan dengan fungsi aktivasi <i>sigmoid</i> bipolar – <i>sigmoid</i> bipolar	52
Tabel 4.4 Hasil pelatihan dengan 5 sel neuron dengan berbagai fungsi aktivasi	53
Tabel 4.5 Hasil pelatihan dengan 7 sel neuron dengan berbagai fungsi aktivasi	53
Tabel 4.6 Hasil pelatihan dengan fungsi aktivasi <i>sigmoid</i> bipolar – <i>sigmoid</i> bipolar	54
Tabel 4.7 Hasil pelatihan dengan fungsi aktivasi <i>sigmoid</i> bipolar – <i>sigmoid</i> bipolar	55
Tabel 4.8 Hasil pelatihan dengan 13 – 10 – 2 sel neuron fungsi aktivasi <i>sigmoid</i> bipolar – <i>sigmoid</i> bipolar – <i>sigmoid</i> bipolar	57
Tabel 4.9 Hasil pelatihan dengan 13 – 10 – 2 sel neuron dengan berbagai fungsi aktivasi	58
Tabel 4.10 Hasil pelatihan dengan 3 – 4 lapisan tersembunyi dengan berbagai macam jumlah neuron	59
Tabel 4.11 Hasil pelatihan dengan fungsi aktivasi <i>Sigmoid</i> bipolar – <i>sigmoid</i> bipolar	60

Tabel 4.12 Hasil pelatihan dengan fungsi aktivasi <i>sigmoid</i> biner ... <i>sigmoid</i> biner – <i>sigmoid</i> biner	61
Tabel 4.13 Hasil pelatihan dengan berbagai macam fungsi aktivasi	62
Tabel 4.14 Hasil pelatihan dengan 3 – 4 lapisan tersembunyi dengan berbagai macam jumlah neuron	63
Tabel 4.15 Hasil pelatihan terbaik dari masing masing pengelompokan pada input yang digeser	65
Tabel 4.16 Hasil pelatihan terbaik dari masing masing pengelompokan pada input yang tidak digeser	65
Tabel 4.17 Perbandingan suhu air setpoint dengan suhu air tangki untuk input yang digeser	76
Tabel 4.18 Perbandingan suhu air setpoint dengan suhu air tangki untuk input yang tidak digeser	77

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Kebutuhan akan pengendalian suatu kondisi makin banyak diperlukan manusia. Hadirnya konsep teknologi otomasi telah memberikan kontribusi besar bagi pertumbuhan maupun perkembangan teknologi secara umum. Demikian pula untuk sistem pengendalian sebuah perangkat. Untuk banyak hal manusia tidak ingin direpotkan dengan peranannya khusus yang mendukung sisi kehidupannya. Jika memungkinkan perannya ingin seminimal dan sesederhana mungkin.

Seperti halnya dalam menangani pengendalian suhu. Untuk menjaga kestabilannya tidak lagi memerlukan campur tangan untuk menaikkan dan menurunkan suhu. Tetapi peran itu telah menjadi bagian kerja sistem pengendali secara otomatis. Walaupun ada peran manusia lebih pada setting kondisi yang diperlukan.

Kemudahan perancangan sistem ini didukung dengan penemuan – penemuan baru di bidang *Artificial Intelligence* yang banyak memberikan ide terobosan untuk merancang sistem pengendalian yang lebih efektif, stabil dan lebih mudah pengaturannya.

Artificial Intelligence (AI) adalah istilah yang umum digunakan untuk menggambarkan komputer atau program komputer yang mampu berfikir dan memecahkan masalah dengan pendekatan intuitif atau estimasi terbaik seperti yang



dilakukan manusia, bukannya pendekatan kuantitatif yang dimiliki oleh komputer konvensional. *Expert System* (Sistem Pakar), *Neural Network* (Jaringan Syaraf Tiruan) dan *Fuzzy Logic* adalah beberapa metode AI yang paling sering dipakai. Ketiganya relatif masih baru dalam implementasi dan pengembangannya, namun semuanya mempunyai implikasi yang luas dalam kehidupan kita.

Jaringan Syaraf Tiruan (JST) merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan syaraf diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. Secara garis besar, kelebihan utama JST adalah :

1. JST tidak perlu diprogram, karena hasil yang diinginkan dengan mudah dapat diajarkan. Hal ini mengurangi sebagian biaya dan waktu pemrograman.
2. JST dapat meningkatkan responnya dengan belajar, karena JST di desain agar dapat mengevaluasi output dan secara otomatis menyesuaikan kriteriannya berdasarkan selisih data *feedbauck* dan data refrensi.
3. Data input tidak harus presisi, karena JST bekerja berdasarkan penjumlahan beberapa input. Contohnya, JST untuk sistem pengenalan pola tetap dapat mengenali seorang meskipun orang tersebut mempunyai model rambut yang berbeda dengan image targetnya. Teknik digital akan sulit mewujudkan hal ini.
4. Informasinya tidak disimpan dalam lokasi memori yang spesifik seperti pada komputer digital, karena JST menyimpannya sebagai jaringan interkoneksi dan

berdasarkan bobotnya, sehingga tidak berfungsinya suatu *neuron* tidak akan menghambat operasi sistem secara keseluruhan

Kemudahan perancangan sistem ini didukung dengan penemuan - penemuan baru di bidang *Artificial Intelligence* yang banyak memberikan ide terobosan untuk merancang sistem pengendalian yang lebih efektif, stabil dan lebih mudah pengaturannya. Misalnya suhu shower, sebagai bentuk aplikasi yang akan dirancang ini. Perancangan sistem kendali ini diharapkan dapat memberikan kontribusi bagi pengguna dan menjadi teknologi alternatif dari kebutuhan sistem serupa dalam berbagai aplikasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, maka dapat diambil suatu rumusan masalah sebagai berikut : “Bagaimana mengendalikan suhu shower dengan algoritma Jaringan Syaraf Tiruan, berdasarkan metode *Back Propagation (BP)*.”

1.3 Batasan Masalah

Dengan adanya batasan masalah penulis dapat lebih menyederhanakan dan mengarahkan penelitian dan pembuatan sistem agar tidak menyimpang dari apa yang diteliti dan dikembangkan. Batasan-batasannya adalah sebagai berikut :

1. Penelitian difokuskan pada pendekatan pelatihan Jaringan Syaraf Tiruan menggunakan metode *Back Propagation*.
2. Suhu shower dimodelkan dengan persamaan matematis.
3. Pembuatan sistem ini menggunakan perangkat lunak Matlab 6.5.
4. Pelatihan dan pengujian JST menggunakan fungsi yang ada pada Matlab.

1.4 Tujuan Penelitian

Penelitian dan perancangan sistem memiliki beberapa tujuan, yaitu :

1. Merancang dan mensimulasikan sebuah sistem penggerak cerdas dengan algoritma jaringan syaraf tiruan metode *backpropagation*.
2. Mempelajari dan memanfaatkan *toolbox neural network* dan *simulink* pada Matlab sebagai media pelatihan dan simulasinya.
3. Membuat suatu pelatihan untuk menghilangkan atau memperkecil galat yang terjadi agar sistem JST dapat dikatakan sempurna.

1.5 Metodologi Penelitian

1. Pengumpulan Data

Data diperoleh dari studi pustaka berupa buku, artikel, makalah dan tutorial yang tersedia pada *website* di internet.

2. Studi Pustaka

Pengumpulan data ini digunakan untuk mendapatkan informasi-informasi yang berkaitan dengan proses penyusunan tugas akhir, sehingga dapat digunakan sebagai acuan dalam proses pembuatan simulasi.

3. Pemecahan Masalah

Setelah semua data terkumpul, maka dilakukan perancangan sistem, pembuatan simulasi sistem dan pengujian sistem.

1.6 Sistematika Penulisan

Sistematika penulisan tugas akhir ini terdiri dari 5 bab bagian isi laporan, dengan penjelasan bab sebagai berikut :

BAB I : PENDAHULUAN

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

BAB II : LANDASAN TEORI

Memuat dasar-dasar teori yang berhubungan dengan penelitian dan juga dasar teori yang berhubungan dengan jaringan syaraf dan pemodelan suhu shower yang akan digunakan.

BAB III : PERANCANGAN SISTEM

Menjelaskan tentang pemrograman jaringan syaraf, mensimulasikan rancangan dan pengujian sistem yang telah dibuat, pembagian fungsi kerja dalam diagram blok dan diagram alir serta berisi lebih terperinci tentang apa yang telah disampaikan pada proposal Tugas Akhir.

BAB IV : PENGUJIAN, ANALISIS DAN PEMBAHASAN

Membahas tentang hasil pengujian dan analisis dari sistem yang dibuat dibandingkan dengan dasar teori sistem atau uraian alasan ilmiah yang lain.

BAB V : PENUTUP

Berisi kesimpulan dan saran-saran dari proses perancangan, simulasi sistem, serta keterbatasan-keterbatasan yang ditemukan dan juga asumsi-asumsi yang dibuat selama melakukan penelitian.

BAB II

LANDASAN TEORI

2.1 Sistem Jaringan Syaraf Manusia

Tubuh Manusia dilengkapi dengan dua perangkat pengatur seluruh kegiatan tubuh. Kedua perangkat ini merupakan sistem koordinasi yang terdiri dari sistem syaraf dan sistem hormon. Perbedaan keduanya adalah sistem syaraf bekerja dengan cepat menanggapi adanya perubahan lingkungan yang merangsangnya baik yang berasal dari dalam tubuh (rasa lapar, kenyang, nyeri, kelelahan, dan lain-lain) maupun yang berasal dari luar tubuh (bau, pahit, manis, sentuhan cahaya, suhu, tekanan, gaya berat dan lain-lain) dan pengaturannya dilakukan oleh benang-benang syaraf. Sedangkan sistem hormon bekerja jauh lebih lambat tetapi lebih teratur dan berurutan dalam jangka waktu yang lama. Pengangkutan hormon dilakukan melalui pembuluh darah. Sistem jaringan syaraf dikenal dengan fungsi-fungsinya sebagai :

1. Sensor (reseptor yang mampu memonitor atau memantau dan mendeteksi besar serta kecepatan energi lingkungan).
2. Saluran komunikasi dari sensor ke pusat, yaitu serabut-serabut syaraf yang membawa impuls-impuls syaraf dengan pola *spasiotemporal* yang berfungsi untuk menerjemahkan impuls dari samping otak besar ke pusat.
3. Pusat komputasi dan pembuat keputusan berupa pola geometris kontak-kontak sinapsis antar neuron di pusat susunan syaraf.



4. Saluran komunikasi dari pusat ke efektor, berupa serabut-serabut syaraf yang membawa impuls dari pusat ke tepi.
5. Efektor, berupa sel-sel otot maupun kelenjar.

2.1.1 Anatomi Sel Syaraf Biologis

Otak manusia diperkirakan terdiri atas 10^{11} sel syaraf (neuron). Dalam otak inilah terdapat fungsi-fungsi yang sangat banyak dan rumit, di antaranya adalah ingatan, belajar, bahasa, penalaran, kecerdasan dan lain-lain. Untuk membentuk fungsi-fungsi tersebut, tiap sel syaraf saling berhubungan membentuk jaringan yang sangat rumit dan kompleks yang disebut jaringan syaraf. Jaringan syaraf menerima ribuan informasi kecil dari berbagai organ sensoris dan mengintegrasikannya untuk membentuk reaksi yang harus dilakukannya. Pada prinsipnya sistem jaringan syaraf mempunyai fungsi :

1. Menerima informasi (*acceptance*),
2. Menyimpan informasi (*storage*),
3. Mengirim informasi (*transfer*).

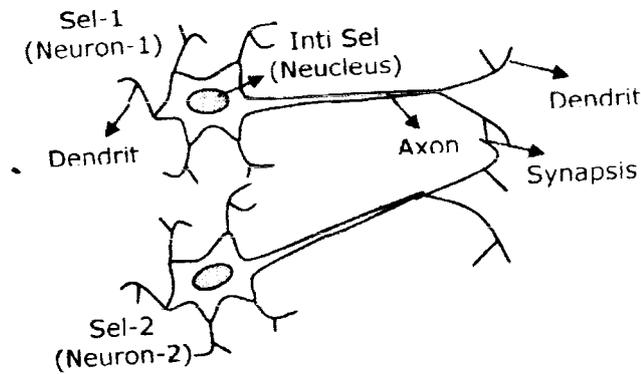
Sebagian kegiatan sistem jaringan syaraf didasari oleh pengalaman sensoris dari reseptor visual, reseptor auditoris dan reseptor raba di permukaan tubuh ataupun jenis reseptor lainnya, reseptor itu sendiri adalah alat yang mampu menerima rangsang. Pengalaman sensoris ini dapat menyebabkan suatu reaksi yang baru terjadi dan kenangannya dapat disimpan di dalam otak selama bermenit-menit, berminggu-minggu atau bertahun-tahun, dan kemudian dapat membantu reaksi tubuh di masa yang akan datang. Karakteristik organisasi sistem jaringan syaraf manusia pada

dasarnya terbagi menjadi reseptor, konduktor dan efektor. Susunan dan cara kerja sistem syaraf biologis adalah sebagai berikut :

1. Pemrosesan informasi dilakukan pada neuron.
2. Isyarat dilewatkan antar neuron melalui koneksi sinapsisnya.
3. Setiap koneksi sinapsis memiliki nilai bobot tertentu sebagai penyimpanan informasi.
4. Pada neuron terjadi proses aktivasi untuk menghasilkan isyarat keluaran.

2.1.2 Cara Kerja Sel Syaraf Biologis

Suatu neuron terdiri atas tiga bagian utama, yaitu soma (badan induk neuron), akson (jalur keluaran dari soma), dan dendrit (menerima impuls/rangsang dari ujung akson lainnya). Diperkirakan ada 10^4 sinapsis setiap neuron dalam otak manusia. Karena sinapsis merupakan tempat hubungan satu neuron dengan neuron berikutnya, maka pada tempat itu sangat menguntungkan untuk mengatur penghantar sinyal. Beberapa sinapsis menghantarkan sinyal dari satu neuron ke neuron lainnya dengan mudah, sedangkan sinapsis yang lain sulit menghantarkan sinyal. Selain itu beberapa neuron pasca sinapsis bereaksi dengan sejumlah besar impuls sedangkan lainnya hanya bereaksi terhadap beberapa impuls saja. Jadi sinapsis melakukan beberapa tindakan selektif, misalnya dengan menghalangi sinyal lemah tetapi meneruskan sinyal kuat, menyeleksi, menguatkan sinyal lemah tertentu, dan tidak jarang menyalurkan sinyal ke berbagai arah, bukan hanya ke satu arah saja.



Gambar 2.1 Sistem jaringan saraf manusia

Pada gambar 2.1 dapat diperhatikan bahwa terdapat ratusan sampai ribuan bongkol kecil yang disebut bongkol sinaptik (*sinaptik knob*) yang terdapat pada permukaan dendrit dan *neucleus*, kira-kira 80-90% di antaranya terdapat pada dendrit. Bongkol ini merupakan ujung terminal fibril syaraf yang berasal dari banyak neuron lain dan beberapa bongkol biasanya berasal dari satu neuron. Beberapa bongkol sinaptik ini bersifat eksitatif yang mensekresikan zat (seperti asetilkolin, norepinefrin, dopamin dan serotonin) yang merangsang neuron, sedangkan lainnya bersifat inhibitif yang mensekresikan suatu zat (seperti asam gamma, aminobutirat dan glisin) yang menghibisi neuron.

2.2 Sistem Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan digunakan karena jaringan syaraf diimplemantasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama pembelajaran.

JST dikenal juga sebagai model *free-estimator*, karena dibandingkan dengan cara perhitungan konvensional, JST tidak memerlukan atau menggunakan suatu model matematis dari permasalahan yang dihadapi. Kemudian dikenal juga sebagai *black box technology* (kotak hitam) atau *opaque* (tidak transparan), karena JST tidak dapat menerangkan bagaimana suatu hasil didapatkan. Hal inilah yang membuat JST mampu digunakan untuk menyelesaikan persoalan yang tidak terstruktur dan sulit didefinisikan, dan penerapannya yang telah meluas dipakai sebagai alat bantu memecahkan masalah pada berbagai bidang dan disiplin ilmu. Beberapa aplikasi jaringan syaraf tiruan :

1. Aplikasi : Memilih suatu input data ke dalam satu kategori tertentu yang ditetapkan
2. Asosiatif : Mengambarkan suatu obyek secara keseluruhan hanya dengan sebuah bagian dari obyek lain
3. Optimasi : Menemukan suatu jawaban atau solusi yang paling baik sehingga seringkali dengan meminimalkan suatu fungsi biaya (*optimizer*)
4. *Self organization* : Kemampuan untuk mengolah data-data input tanpa harus memiliki data sebagai target

Dengan kelebihan-kelebihan yang dimiliki, jaringan syaraf tiruan tetap mempunyai sejumlah keterbatasan, di antaranya jaringan syaraf kurang mampu dalam melakukan operasi-operasi numerik dengan presisi tinggi dan terkadang membutuhkan waktu berhari-hari untuk jumlah data yang besar karena sulitnya mengukur performansi sebenarnya dari jaringan syaraf tiruan.

2.2.1 Jaringan Syaraf Manusia sebagai Dasar Jaringan Syaraf Tiruan

Pada dasarnya ada beberapa hal yang mendasari kerja jaringan syaraf manusia ini, di antaranya mengenai penyimpanan informasi dan daya ingat, akson dan dendrit yang bercabang-cabang sedemikian banyaknya, dan proses pengolahan informasi yang terdapat pada jaringan syaraf manusia. Dalam jaringan syaraf manusia, bila suatu sinyal tertentu melalui sinapsis secara berulang-ulang, maka sinapsis tersebut menjadi lebih mampu menghantarkan sinyal pada kesempatan berikutnya, hal ini biasa dikenal dengan istilah penyimpanan informasi dan daya ingat. Hal ini mendasari adanya proses belajar atau pelatihan (*learning*), jadi jaringan syaraf tiruan yang akan digunakan pasti melalui proses pelatihan secara berulang-ulang terlebih dahulu.

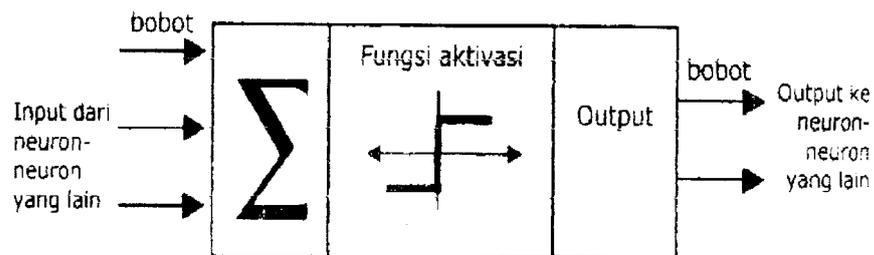
Dalam jaringan syaraf manusia, akson dan dendrit bercabang-cabang sedemikian banyaknya. Hal ini menunjukkan bahwa adanya sistem paralel dan terdistribusi pada dalam jaringan syaraf tiruan. Letak perbedaannya, akson dan dendrit pada jaringan syaraf manusia bercabang-cabang dengan pola yang tidak teratur, sedangkan pada jaringan syaraf tiruan, keparalelan dan kedistribusian cabang-cabang itu membentuk pola tertentu.

Jaringan syaraf tiruan merupakan bagian dari *Artificial Intelligence* atau kecerdasan buatan yang berbasis hubungan, karena cara kerjanya melihat pada jaringan syaraf manusia. Secara garis besar dapat dijelaskan sebagai berikut: beberapa bongkol (baik eksitasi maupun inhibisi) masuk ke suatu neuron, oleh neuron masukan tersebut dijumlahkan, kemudian dibandingkan dengan nilai

ambangya. Hasil penjumlahan baru bisa berarti jika besarnya kecilnya bobot hubungan telah teratur.

2.2.2 Cara Kerja Komponen Jaringan Syaraf Tiruan

Ada beberapa tipe jaringan syaraf tiruan, tetapi hampir semuanya memiliki komponen yang sama. Sama halnya seperti otak manusia, jaringan syaraf juga terdiri dari beberapa neuron, dan ada hubungan antara neuron tersebut. Neuron-neuron tersebut akan mentransformasikan informasi yang diterima melalui sambungan keluaranya menuju ke neuron-neuron yang lain. Pada JST hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut.

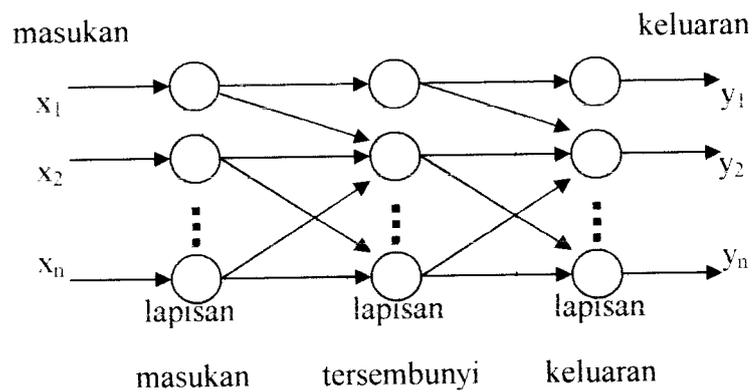


Gambar 2.2 Struktur neuron jaringan syaraf tiruan

Informasi (disebut dengan *input*) akan dikirim ke neuron dengan bobot kedatangan tertentu. *Input* ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila *input* tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan, tetapi jika tidak maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan

mengirimkan *output* melalui bobot-bobot *output*-nya ke semua neuron yang berhubungan dengannya. Hal ini dilakukan secara terus menerus.

Pada jaringan syaraf, neuron-neuron akan dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut dengan lapisan neuron (*neuron layers*). Biasanya neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya (kecuali lapisan *input* dan lapisan *output*). Informasi yang diberikan pada jaringan syaraf akan dirambatkan dari lapisan ke lapisan, mulai dari lapisan *input* sampai ke lapisan *output* melalui lapisan yang lainnya, yang sering dikenal dengan nama lapisan tersembunyi (*hidden layer*), tergantung pada algoritma pembelajarannya, bisa jadi informasi tersebut akan dirambatkan secara mundur pada jaringan.



Gambar 2.3 Jaringan syaraf dengan tiga lapisan

Beberapa jaringan syaraf ada juga yang tidak memiliki lapisan tersembunyi, dan ada juga yang neuron-neuronnya disusun dalam bentuk matriks.

2.2.3 Konsep Belajar Jaringan Syaraf Tiruan

Ciri utama yang dimiliki oleh sistem jaringan syaraf tiruan adalah kemampuan untuk belajar. Agar berfungsi seperti yang diinginkan, jaringan tidak diprogram seperti yang dilakukan pada sistem komputer sekarang ini, melainkan harus diajari.

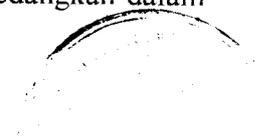
Berdasarkan fungsi masukan keluarannya, fungsi jaringan syaraf tiruan ditentukan oleh parameternya (bobot-bobot koneksi). Untuk kasus yang diketahui fungsi pemetaannya, bobot-bobot tersebut dapat berharga tetap dan ditentukan pada waktu perancangan. Tetapi pada kebanyakan kasus, parameter jaringan yang cocok belum diketahui, dan jaringan harus mencari sendiri besarnya bobot tersebut atau ditentukan secara acak.

Suatu proses penyesuaian parameter secara berurutan dilakukan, dengan tujuan mendekati fungsi yang diinginkan. Proses penyesuaian parameter inilah yang disebut dengan proses belajar dalam sistem jaringan syaraf tiruan. Proses belajar dikategorikan dalam dua jenis :

1. Dengan pengawasan (*supervised learning*),
2. Tanpa pengawasan (*unsupervised learning*).

Proses belajar dengan pengawasan memerlukan keluaran target atau jawaban yang diperlukan dalam proses belajar sebagai dasar penghubung bobot. Jaringan diajar untuk menyelesaikan persoalan-persoalan yang terdapat dalam paket belajarnya.

Selama belajar apabila jaringan mengeluarkan jawaban yang salah, maka besar kesalahan dapat dicari, yaitu beda keluaran aktual dan acuannya. Sedangkan dalam



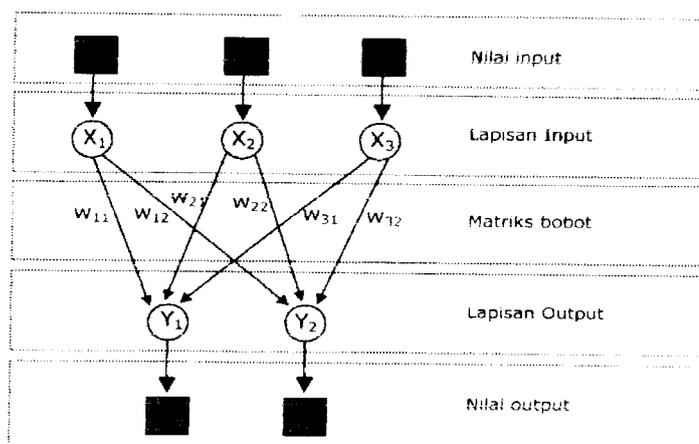
belajar tanpa pengawasan, jaringan akan mengubah bobot-bobotnya, sebagai tanggapan terhadap masukan, tanpa memerlukan keluaran acuan.

2.2.4 Arsitektur Jaringan Syaraf Tiruan

Dalam jaringan syaraf, neuron-neuron dikelompokkan dalam lapisan-lapisan yang umumnya setiap lapisan yang sama akan memiliki keadaan yang sama pula. Apabila neuron-neuron dalam suatu lapisan (misalkan lapisan tersembunyi) akan dihubungkan dengan neuron-neuron pada lapisan lain (misalkan lapisan *output*), maka setiap neuron pada lapisan tersebut (misalkan lapisan tersembunyi) juga harus dihubungkan pada setiap neuron pada lapisan yang lainnya (misalkan lapisan *output*). Ada beberapa arsitektur jaringan syaraf, antara lain :

1. Jaringan dengan lapisan tunggal (*single layer net*).

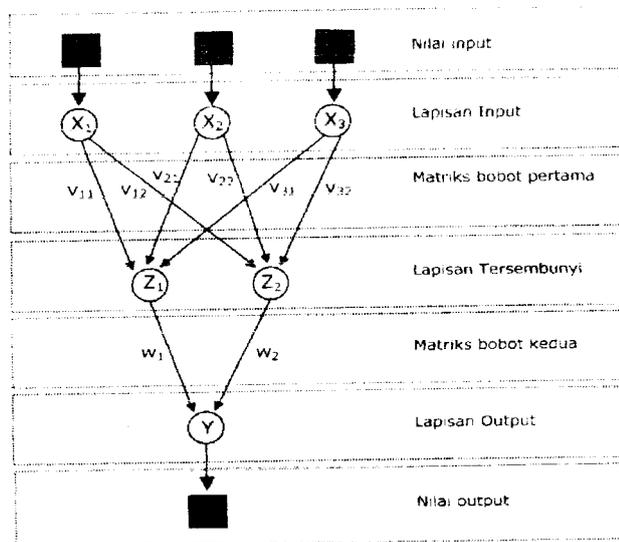
Hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan hanya menerima masukan kemudian secara langsung mengolahnya menjadi keluaran tanpa harus melalui lapisan tersembunyi. Semua unit *input* akan langsung berhubungan dengan unit *output*.



Gambar 2.4 Jaringan syaraf dengan lapisan tunggal

2. Jaringan dengan banyak lapisan (*multilayer net*).

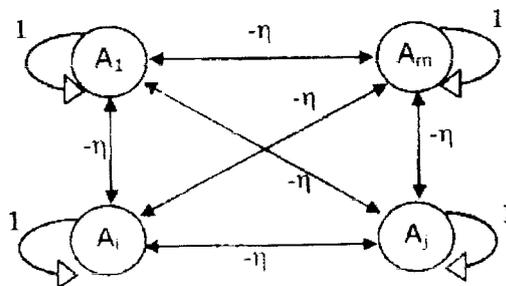
Memiliki satu atau lebih lapisan tersembunyi yang terletak diantara lapisan *input* dan *output*. Dengan pembelajaran yang lebih rumit, jaringan dapat menyelesaikan permasalahan yang lebih sulit dan hasilnya pun lebih sukses dari pada lapisan tunggal.



Gambar 2.5 Jaringan syaraf dengan banyak lapisan

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

Umumnya hubungan antar neuron tidak diperlihatkan pada diagram arsitektur jaringan dengan lapisan kompetitif. Dari gambar berikut bobot-bobotnya ditunjukkan oleh $-\eta$.



Gambar 2.6 Jaringan syaraf dengan lapisan kompetitif

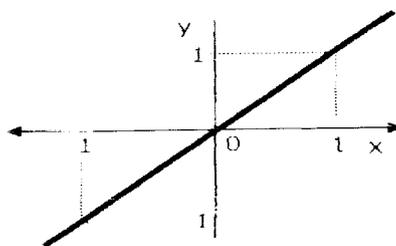
2.2.5 Fungsi Aktifasi

Faktor terpenting dalam menentukan kelakuan suatu neuron adalah fungsi aktivasi dan pola bobotnya. Pada setiap lapisan yang sama, neuron-neuron akan memiliki fungsi aktivasi yang sama. Suatu fungsi aktivasi untuk setiap sel sebuah jaringan saraf tiruan *backpropagation* mempunyai beberapa karakteristik penting sebagai berikut :

1. Harus *continue*, dapat diturunkan.
2. Tidak naik atau turun secara monoton (*monotonically nondecreasing*).

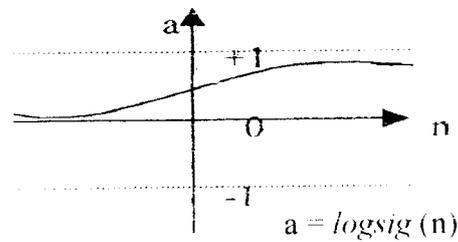
Untuk mendapatkan efisiensi perhitungan, turunannya mudah dihitung. Pada umumnya nilai turunan fungsi aktivasi dapat diperoleh dari nilai rentang tertentu. Ada beberapa fungsi aktivasi yang sering digunakan dalam jaringan syaraf tiruan, tetapi yang biasa digunakan untuk metode *backpropagation* adalah:

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



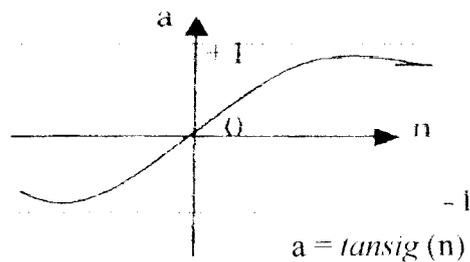
Gambar 2.7 Fungsi aktivasi *linear* identitas

2. *Logsig* (*sigmoid* biner), memiliki nilai *range* antara 0 s/d 1.



Gambar 2.8 Fungsi aktivasi *sigmoid* biner

3. *Tansig* (*sigmoid* bipolar), hampir sama dengan fungsi *sigmoid* biner, hanya saja *output* dari fungsi memiliki *range* antara 1 s/d -1



Gambar 2.9 Fungsi aktivasi *sigmoid* bipolar

2.2.6 Metode *Backpropagation*

Jaringan syaraf tiruan rambat balik atau yang biasa kita kenal dengan nama *backpropagation* merupakan jaringan syaraf tiruan yang paling banyak diterapkan dalam berbagai bidang, seperti pengenalan pola, diagnosa kedokteran, klasifikasi gambar, menerjemahkan kode, dan bermacam-macam analisa pengenalan pola lainnya. Jaringan ini merupakan salah satu jenis yang mudah dipahami dan konsep belajarnya relatif sederhana, yaitu :

1. Belajar dari kesalahan.
2. Memasukan secara umpan maju (*feed forward*) pola-pola masukan.

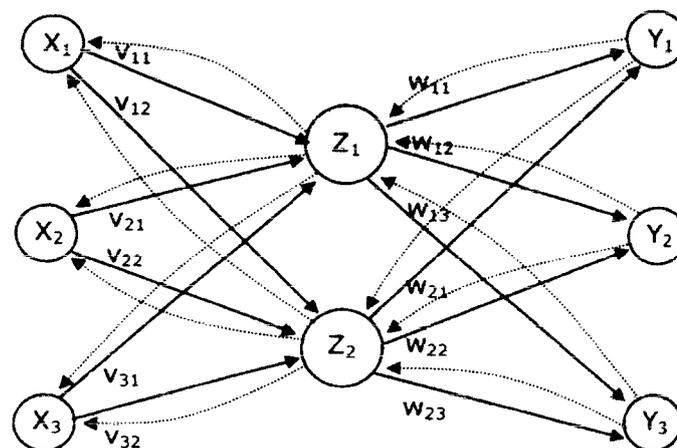
3. Menghitung dan propagasi balik kesalahan yang bersangkutan.
4. Mengatur bobot-bobot koneksi.

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

Setelah pelatihan selesai, komputasi jaringan hanya pada fase umpan maju yaitu memberikan pola-pola masukan. Sekalipun pelatihan lambat, cara kerja jaringan terlatih dapat menghasilkan keluaran sangat cepat.

2.2.6.1 Arsitektur *Backpropagation*

Sebuah arsitektur *backpropagation* dengan jaringan lapis banyak dan memiliki satu lapisan dalam (unit-unit *Z*) ditunjukkan dalam gambar berikut :

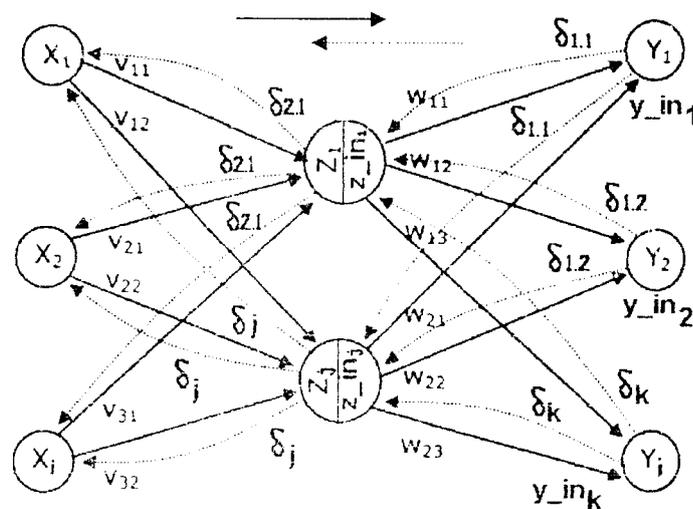


Gambar 2.10 Arsitektur jaringan sederhana

Pada Gambar 2.10 menunjukkan arah sinyal pada fase umpan maju. Selama operasi fase pelatihan perambatan balik, sinyal-sinyal *error* dikirim dengan arah yang sebaliknya. Unit-unit keluaran (y) dan unit-unit lapisan dalam (z) mempunyai bias. Bias pada unit keluaran dinotasikan dengan w_{0k} , bias pada unit lapisan dalam dinotasikan dengan v_{0j} , tindakan bias bekerja seperti pada bobot-bobot koneksi, dimana unit-unit bias selalu mengeluarkan nilai 1. Unit-unit ini bersifat *optional* pada penyajian lain mungkin dihilangkan.

2.2.6.2 Algoritma *Backpropagation*

Jaringan syaraf rambat balik (*backpropagation*) dilatih dengan pembelajaran. Jaringan dilatih dengan contoh *input* dan target *output*. Dalam masing-masing presentasi, bobot diubah untuk mengurangi perbedaan antara *output* jaringan dan target *output*. Setelah pelatihan dilakukan pengujian terhadap jaringan yang telah dilatih. Pembelajaran algoritma jaringan syaraf membutuhkan perambatan maju dan diikuti dengan perambatan mundur. Keduanya dilakukan untuk semua pola pelatihan.



Gambar 2.11 Arsitektur algoritma *backpropagation*

Selama perambatan maju, tiap masukan (x_i) menerima sebuah sinyal masukan dan mengirimkan sinyal ke tiap-tiap unit lapisan tersembunyi $Z_1, 2, \dots, p$. Tiap unit lapisan tersembunyi ini kemudian menghitung aktivasinya (z_{in_j}) dan mengirimkan sinyalnya ke tiap-tiap unit lapisan keluaran (z_j). Tiap unit keluaran (y_{in_k}) menghitung aktivasinya (y_k) untuk membentuk respon pada jaringan terhadap pola masukan yang diberikan.

Selama pelatihan, tiap unit keluaran membandingkan perhitungan aktivasinya (y_k) dengan nilai targetnya (t_k) untuk menentukan kesalahan pola tersebut dengan unit itu. Berdasarkan kesalahan ini, faktor δ_k ($k = 1, \dots, m$) dihitung. δ_k digunakan untuk menyebarkan kesalahan pada unit *output* (y_k) ke semua unit pada lapisan sebelumnya (unit-unit tersembunyi yang dihubungkan ke y_k) dan digunakan nantinya untuk mengubah bobot-bobot antara *output* dan lapisan tersembunyi. Dengan cara yang sama, δ_j ($j = 1, \dots, p$) dihitung untuk tiap unit tersembunyi (z_j). Untuk (δ_j) tidak perlu menyebarkan kesalahan kembali ke lapisan *input*, tetapi digunakan nantinya untuk mengubah bobot-bobot antara lapisan tersembunyi dan lapisan *input*-nya.

Setelah seluruh faktor δ telah ditentukan, bobot untuk semua lapisan diatur secara serentak. Pengaturan bobot w_{jk} (dari tiap-tiap unit lapisan tersembunyi ke tiap-tiap unit *output*) didasarkan pada faktor δ_k dan aktivasi z_{in_j} dari unit tersembunyi z_j . Pengaturan bobot v_{ij} (dari vektor unit *input* ke tiap-tiap unit lapisan tersembunyi) didasarkan pada faktor δ_j dan aktivasi unit *input* x_i .

Suatu jangka waktu (*epoch*) adalah satu set putaran vektor-vektor pelatihan. Beberapa *epoch* diperlukan untuk pelatihan sebuah jaringan syaraf tiruan

backpropagation. Dalam algoritma ini dilakukan perbaikan bobot setelah masing-masing pola pelatihan disajikan. Setelah pelatihan selesai bobot-bobot yang telah diperbaiki disimpan.

2.2.7 Algoritma Levenberg-Marquardt (trainlm)

Algoritma Levenberg-Marquardt dirancang dengan menggunakan pendekatan turunan kedua tanpa harus menghitung matriks Hessian. Apabila jaringan syaraf feedforward menggunakan fungsi kinerja *sum of square*, maka matriks Hessian dapat didekati sebagai:

$$H = J^T * J$$

Dan gradien dapat dihitung sebagai:

$$g_W = J^T * e$$

dengan J adalah matriks Jacobian yang berisi turunan pertama dari error jaringan terhadap bobot, dan e adalah suatu vektor yang berisi error jaringan. Matriks Jacobian dapat dihitung dengan teknik backpropagation standar, yang tentu saja lebih sederhana dibanding dengan menghitung matriks Hessian.

Algoritma Levenberg-Marquardt menggunakan pendekatan untuk menghitung matriks Hessian, melalui perbaikan metode Newton:

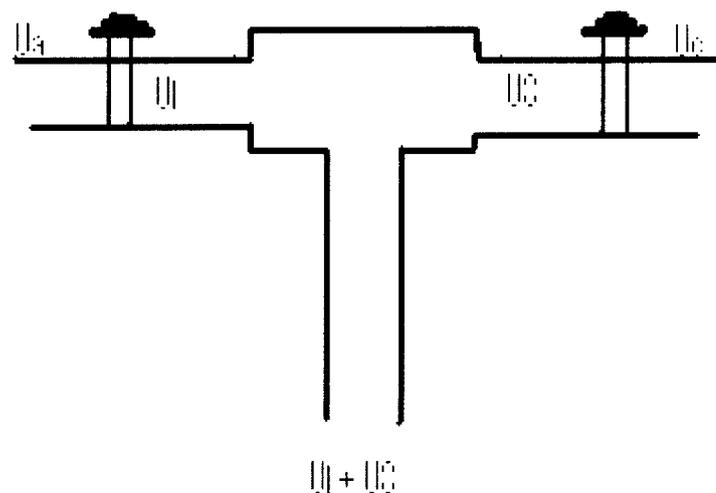
$$W_{k+1} = W_k - [J^T * J + \mu * I]^{-1} * J^T * e$$

Apabila μ bernilai 0, maka pendekatan ini akan sama seperti metode Newton. Namun apabila μ terlalu besar, maka pendekatan ini akan sama halnya dengan gradient descent dengan learning rate yang sangat kecil. Metode Newton sangat cepat dan akurat untuk mendapatkan error minimum, oleh karena itu diharapkan algoritma sesegera mungkin dapat mengubah nilai μ menjadi sama dengan 0. Untuk itu, setelah

2.3.1 Prinsip Shower

Shower terdiri dari tiga tangki, dua tangki berfungsi sebagai tempat air panas dan dingin yang mengatur debit air yang masuk pada bejana ketiga. Tangki ketiga merupakan tempat air dengan suhu yang diinginkan. Katup pada tangki air panas dan dingin berfungsi mengendalikan air yang akan masuk ke tangki ketiga. Air yang masuk pada tangki ketiga merupakan air hasil percampuran dari tangki panas dan tangki dingin. Seberapa besar posisi katup membuka tergantung dari hasil keluaran JST nya. Pada tiap-tiap bejana memiliki suhu sumber yang berbeda beda. Suhu sumber untuk tangki panas 60°C dan suhu sumber tangki dingin 10°C .

Model Shower



Gambar 2.12 Model shower

Definisi yang diperlukan untuk model sistem:

U_a = aliran yang berasal dari *valve setting* panas

U_c = aliran yang berasal dari *valve setting* dingin

U_b = aliran maksimal konstan (1)

U_1 = aliran air dari katup panas

U_3 = aliran air dari katup dingin

$K = 1$

Temp = suhu yang dihasilkan shower

suhu sumber air panas = 60°C

suhu sumber air dingin = 10°C

Besarnya aliran diperoleh dari persamaan sebagai berikut :

$$U_1 = K \times U_a \times ((K \times U_a) \leq U_b) + U_b \times ((K \times U_a) > U_b)$$

$$U_3 = K \times U_c \times ((K \times U_c) \leq U_b) + U_b \times ((K \times U_c) > U_b)$$

Untuk memperoleh besarnya suhu shower digunakan persamaan sebagai berikut :

$$\text{Temp} = \frac{U(1) \times 60 + U(3) \times 10}{U(1) + U(3)}$$

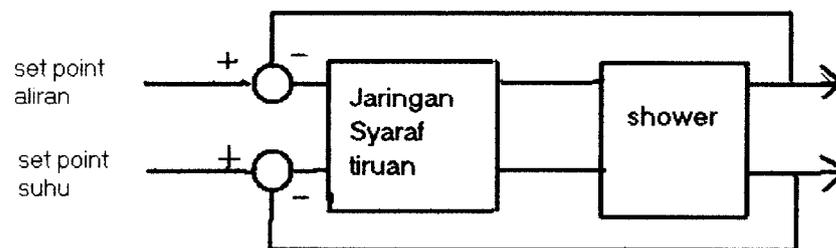
Besarnya aliran pada shower diperoleh dari persamaan berikut :

$$\text{Flow rate} = U(1) + U(3)$$

BAB III

PERANCANGAN SISTEM

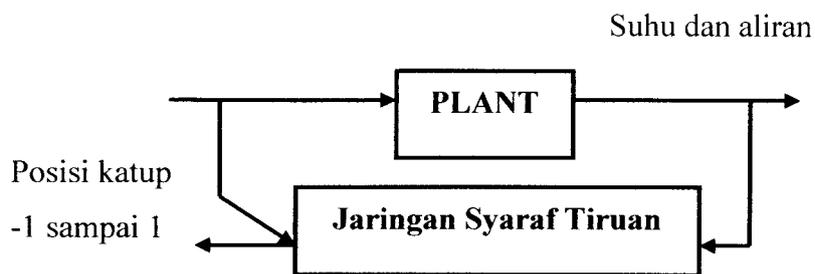
Proses belajar Jaringan syaraf tiruan (JST) dilakukan secara *on-line/continue* (terus-menerus), sehingga JST memerlukan hasil pengendaliannya (suhu yang dihasilkan shower) untuk memperbaiki tanggapan shower. Dalam perancangan sistem, masukan dalam jaringan syaraf adalah berupa suhu yang diinginkan, sedangkan keluaran jaringan syaraf yang juga berfungsi sebagai masukan shower. Sebagai keluaran shower dan juga sebagai hasil akhir dari sistem adalah suhu yang dapat diamati pada tangki. Untuk mengetahui lebih jelas perancangan sistem kendali ini dapat dilihat pada diagram blok berikut :



Gambar 3.1 Blok diagram perancangan sistem

Pelatihan dari sistem pengendalian dirancang dengan menggunakan metode *inverse*, dimana masukan dari *plant*/model shower adalah sebagai target atau keluaran dari jaringan syaraf, sehingga skenario keluarannya akan digunakan kembali sebagai masukan. Karena pada pelatihan menggunakan metode *inverse*, maka masukan dan keluaran dari sistem kendali yang sebenarnya akan dibalik pada

saat pelatihan. Pada saat pelatihan masukan dari jaringan syaraf tiruan adalah keluaran dari shower, yaitu suhu shower, sedangkan keluaran atau target dari jaringan syaraf adalah merupakan masukan shower yang akan mengendalikan katup, yaitu berupa nilai dari -1 sampai 1.

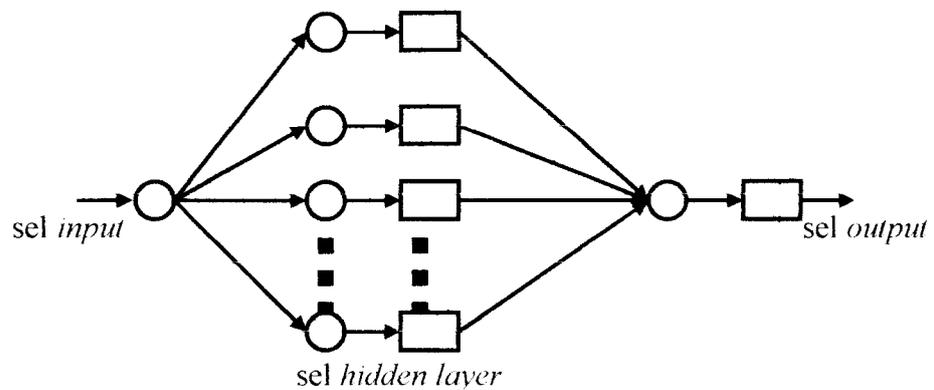


Gambar 3.2 Model *inverse* pelatihan

Kemampuan dari jaringan syaraf tiruan akan dipergunakan untuk mengidentifikasi shower. Selanjutnya hasil proses identifikasi dipergunakan pada proses pengendalian suhu shower. Perangkat lunak yang digunakan dalam perancangan sistem adalah Matlab versi 6.5, karena perangkat lunak ini memiliki bahasa yang bisa digunakan untuk komputasi teknik dan dapat digunakan untuk penghitungan, visualisasi dan pemrograman. Selain itu perangkat lunak Matlab 6.5 juga memiliki *neural network toolbox*, sehingga memudahkan dalam perancangan program jaringan syaraf maupun pensimulasian dari sistem yang telah dilatih. Beberapa kegunaan lain dari Matlab di antaranya adalah untuk pengembangan algoritma, pemodelan, simulasi dan pembuatan antarmuka GUI (*Graphical User Interface*).

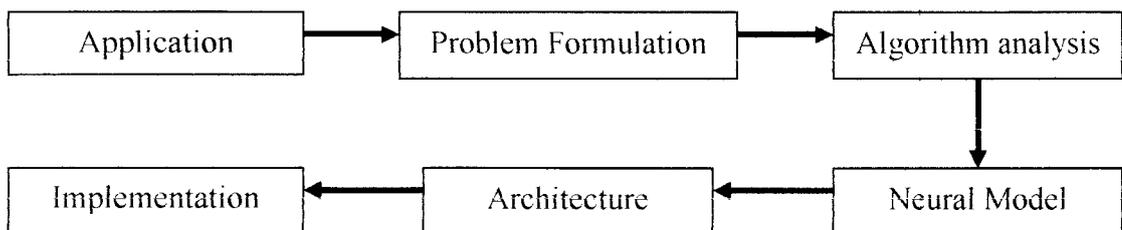
3.1 Perancangan Jaringan Syaraf Tiruan

Jaringan yang digunakan memiliki dua sel neuron pada masukan, dua sel neuron pada pada lapisan keluaran. Sedangkan untuk jumlah dan jumlah lapisan tersembunyi akan di jadikan sebagai pengamatan.



Gambar 3.3 Arsitektur jaringan syaraf tiruan

Jaringan syaraf tiruan yang digunakan sebagai pengendali suhu shower ini menggunakan metode *backpropagation*. Metode *backpropagation* termasuk jenis jaringan yang *autoassociative*, yaitu *range* masukan yang di proses ke dalam jaringan sama dengan *range* hasil yang dikeluarkannya. Proses pembangunan jaringan syaraf tiruan secara umum dapat dilihat dari diagram blok berikut ini :



Gambar 3.4 Proses pembangunan jaringan syaraf tiruan

beberapa iterasi, algoritma akan menuntun nilai μ , kenaikan nilai μ hanya dilakukan apabila dibutuhkan suatu langkah (sementara) untuk menurunkan fungsi kinerja.

2.3 Karakteristik *Plant*

Plant adalah seperangkat peralatan, mungkin hanya terdiri dari beberapa bagian mesin yang bekerja bersama-sama, yang digunakan untuk melakukan suatu operasi tertentu. Perangkat yang digunakan untuk perancangan *plant* ini adalah katup dan sebuah tangki air.

Shower adalah suatu alat yang berfungsi untuk menghasilkan air dengan suhu tertentu sesuai dengan yang kita kehendaki. Mandi menggunakan shower menjadi alternatif lain, apalagi jika menggunakan air hangat. Air yang memancar dari lubang-lubang shower akan menyegarkan badan yang penat setelah seharian beraktivitas. Air yang memancar dari shower bisa diatur debit atau kecepatan semburannya. Kita bisa mengatur suhu shower sesuai dengan suhu yang kita inginkan.

Katup adalah perangkat yang berfungsi untuk mengatur besarnya debit aliran air yang masuk ke tangki penampungan, atau dengan kata lain katup berfungsi sebagai pengontrol aliran air. Katup atau bahasa umum orang sering mengatakan kran banyak sekali bentuk dan jenis-jenisnya salah satunya adalah katup *solenoid*. *Solenoid* adalah peralatan *ON OFF* yang digunakan untuk mengubah sinyal listrik menjadi gerak mekanik linier, *solenoid* yang digunakan dalam sistem pengontrol suhu air adalah suatu katup yang geraknya secara elektris untuk pengaturan kontrol air.

3.1.1 Prosedur Pelatihan

Galat pada jaringan akan di propagasi-balikan selama pelatihan. Galat pada lapisan keluaran akan menentukan galat pada lapisan tersembunyi (*hidden layer*), yang akan digunakan nantinya untuk pengaturan bobot dan bias pada lapisan tersembunyi tersebut. Proses iterasi akan dilakukan berulang kali sampai galat telah sampai pada suatu *level* toleransinya atau mencapai putaran iterasi (*epoch*) maksimum. Langkah pelatihan untuk jaringan syaraf tiruan yang menggunakan metode *backpropagation* untuk lebih jelasnya dapat dilihat melalui langkah-langkah seperti berikut ini :

Langkah 0 : Inisialisasi bobot (v) dan bias (v_0) pada lapisan *input* ke lapisan tersembunyi dan serta bobot (w) dan bias (w_0) pada lapisan tersembunyi ke lapisan keluaran,

Langkah 1 : Menentukan jumlah neurona pada lapisan tersembunyi (hl) kecepatan pelatihan (lr/α), maksimum *epoch* (*goal* me) dan toleransi kesalahan (te), Jika kondisi tidak tercapai, di lakukan langkah 2 – 9,

Langkah 2 : Untuk setiap pasangan pelatihan, di lakukan langkah 3 – 8,

Perambatan maju / *Feedforward* :

Langkah 3 : Tiap unit masukan (x_i ; $i = 1, \dots, n$) menerima sinyal masukan x_i dan menghantarkan sinyal ke semua unit lapisan di atasnya (lapisan tersembunyi),

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} , \quad (3.16)$$

Menggunakan fungsi aktivasi untuk menghasilkan keluaran

$$z_j = f(z_{in_j}) , \quad (3.17)$$

Mengirimkan sinyal ini ke seluruh unit pada lapisan di atasnya (unit keluaran),

Langkah 5 : Tiap unit keluaran (y_k ; $k = 1, \dots, m$) menjumlahkan bobot sinyal masukannya,

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} , \quad (3.18)$$

Menggunakan fungsi aktivasi untuk menghasilkan keluaran

$$y_k = f(y_{in_k}) , \quad (3.19)$$

3.1.3 Pemrograman Jaringan dengan *Procedure* dan *Function*

Penentuan notasi atau tata nama yang digunakan dalam pelatihan untuk jaringan syaraf tiruan metode *backpropagation* dalam perancangan sistem adalah sebagai berikut :

- x Vektor masukan kedalam jaringan.
- t Vektor keluaran target dari jaringan.
- hl Jumlah neuron pada lapisan tersembunyi
- me Maksimum *epoch*
- te Toleransi kesalahan
- v Bobot pada lapisan masukan ke lapisan tersembunyi.
- v0 Bias pada lapisan masukan ke lapisan tersembunyi.

- w Bobot pada lapisan lapisan tersembunyi ke lapisan keluaran.
- w0 Bias pada lapisan tersembunyi ke lapisan keluaran.
- δ_k Koreksi kesalahan untuk pengaturan bobot dan bias pada unit lapisan keluaran ke lapisan tersembunyi.
- δ_i Koreksi kesalahan untuk pengaturan bobot dan bias pada unit lapisan tersembunyi ke lapisan masukan.
- α *Learning rate* atau laju belajar.
- z Sel pada lapisan tersembunyi.
- z_in Fungsi aktivasi untuk sel lapisan tersembunyi.
- y Sel pada lapisan keluaran.
- y_in Fungsi aktivasi untuk sel lapisan keluaran.
- Δ Perubahan
- Σ Penjumlahan

Pada sistem kendali suhu shower, karena data yang digunakan dalam pelatihan terlalu banyak, maka jaringan saraf tiruan dibangun dengan jaringan *feedforward* menggunakan fungsi *newff* yang disediakan oleh *toolbox neural network* pada Matlab. Struktur jaringan saraf tiruan *backpropagation* dengan *newff* adalah sebagai berikut :

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

PR Matriks berukuran Rx2 yang berisi nilai minimum dan maksimum, dengan R adalah jumlah variabel *input*.

Si Jumlah neuron pada lapisan ke-i, dengan $i = 1, 2, \dots, N1$.

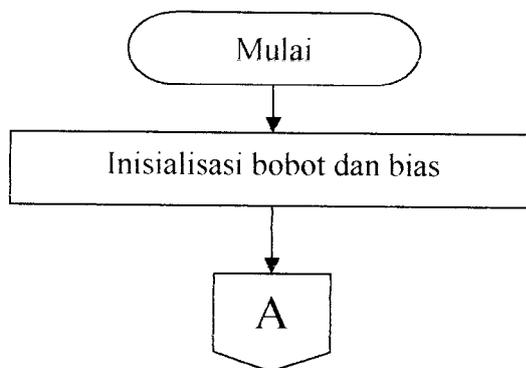
- Tfi Fungsi aktivasi pada lapisan ke-i, dengan $i = 1, 2, \dots, N1$, *default* = *tansig* (*sigmoid bipolar*).
- BTF Fungsi pelatihan jaringan, *default* = *trainlm* (*Levenberg-Marquardt*).
- BLF Fungsi pelatihan untuk bobot, *default* = *learnngdm* (*Gradient descent with momentum*).
- PF Fungsi kinerja kesalahan atau *error*, *default* = MSE (*Mean Square Error*).

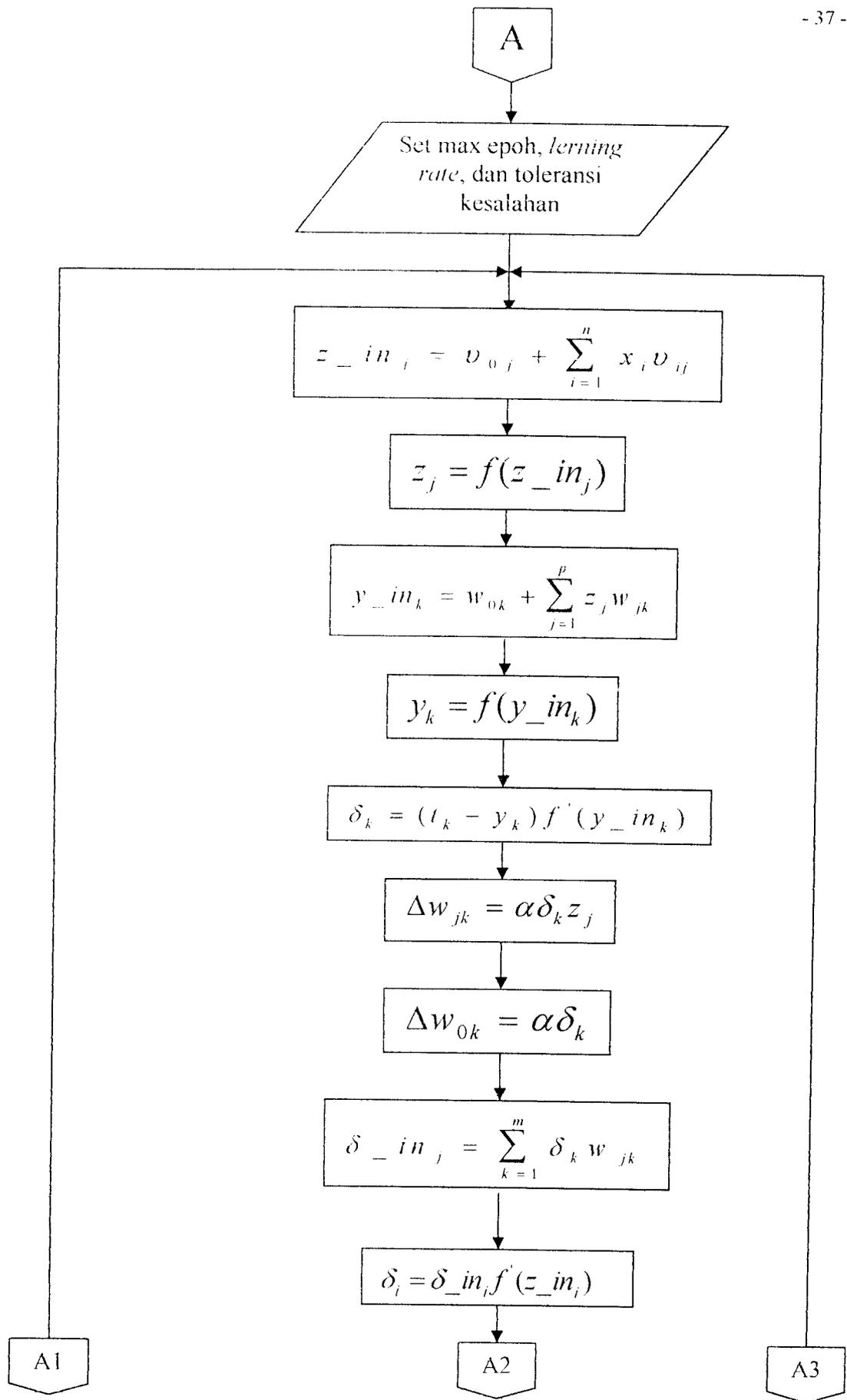
Pelatihan jaringan (BTF) untuk sistem kendali suhu shower menggunakan fungsi *trainlm*.

Beberapa parameter yang ditentukan sebelum melatih jaringan antara lain yaitu jumlah iterasi (*epoch*), *target error* (*goal*), *learning rate* (*lr*) dan tampilan nilai iterasi (*show*).

3.2 Diagram Alir Jaringan Syaraf Tiruan Metode *Backpropagation*

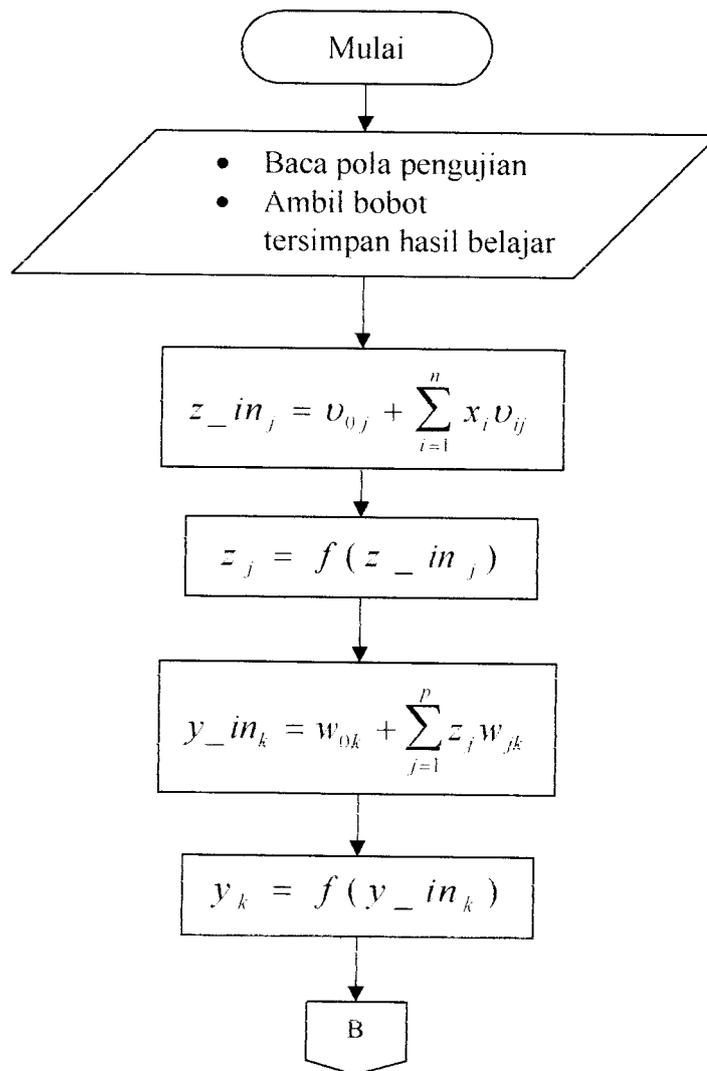
Dengan mengacu pada langkah-langkah prosedur pelatihan 3.1.1 di atas (langkah 0 sampai dengan langkah 9), maka diagram alir untuk prosedur pelatihannya adalah sebagai berikut :

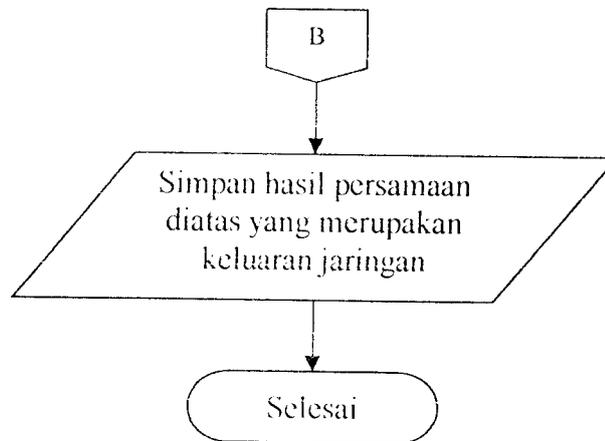




Setelah jaringan belajar dan menyimpan bobot-bobot dan biasanya, baru dapat dilakukan pengujian jaringan. Pada pengujian inilah jaringan yang telah dilatih digunakan untuk menyelesaikan masalah. Sebelum jaringan dilatih, maka jaringan belum bisa digunakan untuk menyelesaikan suatu masalah.

Begitu juga dengan mengacu pada langkah-langkah prosedur pengujian 3.1.2 diatas (langkah 0 sampai dengan langkah 5), maka diagram alir prosedur pengujiannya adalah sebagai berikut :



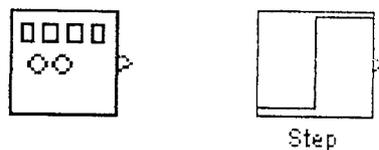


Gambar 3.6 Diagram alir/flowchart prosedur pengujian

3.3 Simulasi Sistem Kendali Suhu Shower

Proses simulasi sistem di kerjakan menggunakan *toolbox simulink*. Sehingga shower yang dimodelkan diambil dari blok yang ada pada *simulink*. Begitu juga dengan komponen-komponen simulasi jaringan syaraf tiruan yang dimodelkan dengan persamaan matematis ataupun yang menggunakan program. Berikut ini adalah beberapa blok *simulink* yang digunakan pada simulasi sistem :

1. *Signal generator* dan *step generator*



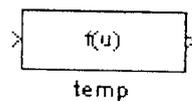
Merupakan pembangkit gelombang dengan tiga bentuk gelombang dasar yang dapat dibangkitkan, dan bisa juga membangkitkan gelombang secara *random/acak*. Tiga bentuk gelombang yang bisa dibangkitkan adalah bentuk gelombang sinus, gelombang kotak dan gergaji. Pada pelatihan akan digunakan gelombang *random*. Setelah jaringan dilatih, maka fungsi masukan gelombang diganti dengan blok *step generator*.

2. Uniform Random Number



Merupakan pembangkit gelombang yang fungsinya sebagai input pengambilan data dan juga merupakan target pelatihan.

3. fcn



Merupakan blok yang berisi persamaan.

4. Constant



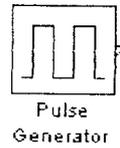
Memberikan sebuah nilai atau bilangan sebenarnya yang tetap agar keluaran dari sistem yang diinginkan memiliki nilai konstan. Pada perancangan sistem digunakan agar besarnya nilai atau gelombang yang dibangkitkan bernilai tetap.

5. Sum



Merupakan blok penjumlah atau pengurang input sesuai dengan tanda operasinya. Blok ini menerima input berupa skalar, vektor, matrik atau elemen dari vektor tunggal. Bila blok memiliki lebih dari satu input, kedua input tersebut harus vektor atau matrik yang memiliki dimensi sama.

6. *Pulse generator*



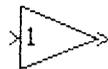
Merupakan blok yang digunakan untuk membangkitkan pulsa pada pengambilan data pelatihan. Data acak kecepatan dibuat dengan fungsi *randperm*.

7. *Multiplexer*



Blok *Mux* mengkombinasikan beberapa input menjadi satu input. Input dapat berbentuk skalar, vektor, atau matrik. Keluaran blok ini tergantung masukannya, bisa berupa vektor atau sinyal gabungan berisi matrik dan vektor. Jika semua inputnya vektor maka outputnya vektor. Jika beberapa inputnya bukan vektor, misalnya matrik. Maka output dari *mix* adalah sinyal bus.

8. *Gain*



Blok *gain* ini merupakan blok penguat yang memiliki nilai keluaran tetap. Masukan pada blok bisa berupa besaran skalar, vektor atau matrik. Pada simulasi sistem.

9. *To Workspace*



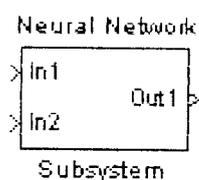
Blok ini menyimpan data dari hasil simulasi ke *workspace* Matlab. Pada saat simulasi, parameter simulasi diatur pada kondisi solver :

Stop time sesuai dengan jangka waktu yang diinginkan.

Type *variable step, discrete (no continuous state).*

Hasil simulasi disimpan dalam *variabel* tegangan dan kecepatan yang di gunakan untuk data pelatihan dan pengujian *offline*.

10. *Subsystem*



Di dalam blok *subsystem* terdapat struktur jaringan syaraf pengujian, struktur dibuat secara manual, karena komputasi dalam *simulink* akan lebih cepat jika masing-masing komponen jaringan syaraf menggunakan blok dari *simulink*.

3.4 Perancangan simulasi pada *plant*

Gambaran dari bejana yang dikendalikan suhu shower nya dimana jika suhu air shower yang diinginkan kurang dari target maka kran akan terbuka dan air akan masuk sehingga akan mempengaruhi suhu yang ada dalam bejana, dan sebaliknya jika suhu sudah melewati batas, maka kran akan tertutup sampai pada batas suhu yang diinginkan terpenuhi. Pada simulasi ini bejana atau tangki air memiliki suatu pembuangan air yang aliran airnya konstan.

3.4.1 Gambaran fisik *plant*

Tangki air panas :

Suhu sumber air panas = 60°C

Aliran air masuk = 1 liter / detik

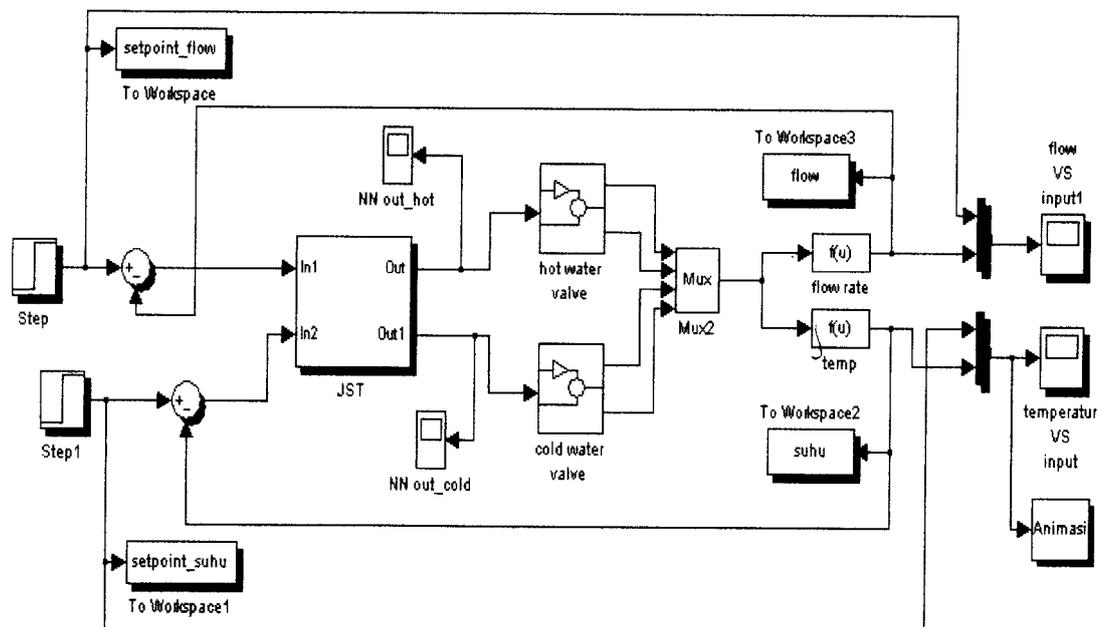
Tangki air dingin :

Suhu sumber air dingin = 10°C

Aliran air masuk = 1 liter / detik

3.4.2 Gambaran *plant* pada simulasi

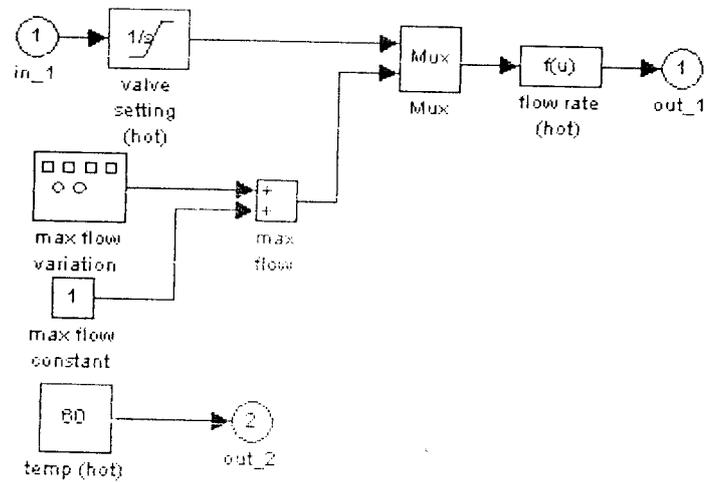
3.4.2.1 Simulink Utama



Gambar 3.8 Blok pada simulasi utama

3.4.2.2 Katup

Katup akan mendapatkan *input* dari Jaringan syaraf tiruan, katup akan mengontrol aliran air sesuai dengan fungsinya. Dengan prinsip kerja diatas maka perancanganya seperti pada gambar dibawah ini :

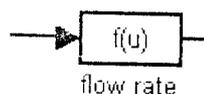


Gambar 3.9 Blok katup pada simulasi

Pada blok katup ini berfungsi sebagai panghasil keluaran dari aliran air panas. Besarnya aliran tergantung dari nilai keluaran jst yang kemudian akan melalui beberapa blok lagi sehingga hasil akhirnya nanti menghasilkan aliran yang akan mempengaruhi besarnya suhu air.

3.4.2.3 flow rate

Besarnya aliran air (*flow rate*) tergantung dari nilai target yang kita harapkan. Pada blok berisi persamaan seperti dibawah ini :



Keterangan :

$$\text{Flow rate} = U(1) + U(3)$$

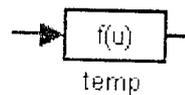
$U(1)$ = aliran air panas

$U(3)$ = aliran air dingin

Aliran air yang masuk merupakan hasil penjumlahan dari aliran air panas dengan aliran air dingin.

3.4.2.4 Temperatur

Temperatur yang dihasilkan dari sistem simulasi tergantung dari aliran air yang telah diatur oleh JST, kemudian aliran tersebut akan masuk kedalam persamaan dibawah ini :



$$\text{Temp} = \frac{U(1) \times 60 + U(3) \times 10}{U(1) + U(3)}$$

Keterangan :

$U(1)$ = aliran air dari katup panas

$U(3)$ = aliran air dari katup dingin

60 = suhu sumber air panas

10 = suhu sumber air dingin

Keluaran blok ini merupakan hasil akhir yang ingin dicapai berupa suhu shower yang diinginkan yang nilainya tergantung dari target yang kita inginkan.

BAB IV

ANALISIS DAN PEMBAHASAN

Sebelum melakukan pelatihan jaringan syaraf tiruan, terlebih dahulu dilakukan penentuan kondisi dari keadaan *plant*. Kondisi ini harus ditentukan karena jaringan yang telah di latih tidak bisa bekerja pada semua kondisi *plant*. Data input pelatihan yang diuji akan mempengaruhi hasil JST nya. Dengan data pelatihan yang baik maka diperoleh hasil keluaran JST nya yang baik pula begitu pula sebaliknya.

Dengan menggunakan data yang ada, langkah pertama yang dilakukan adalah membuat sebuah blok simulasi yang berfungsi untuk mencari data untuk melatih jaringan syaraf tiruan. Dari blok simulasi tersebut akan dihasilkan data *input* dan *output* dari hasil simulasi, kemudian disimpan kedalam *workspace* Matlab untuk dijadikan sebagai masukan dan target pada pelatihan jaringan syaraf tiruan sebagai pengendali suhu.

Pelatihan dengan menggunakan *for-while loops* kurang mendapatkan hasil yang lebih maksimum, disebabkan karena data *input* dan target jaringan syaraf terlalu banyak, kurang lebih sebanyak 166667 data *input* dan dengan jumlah yang sama untuk data targetnya. Sebagai perbandingan, untuk melakukan 1000 iterasi dengan 1 lapisan tersembunyi dan 7 sel neuron pada jaringan syaraf yang menggunakan *for-while loops*, membutuhkan waktu kurang lebih selama 18 jam. Berbeda dengan pelatihan yang menggunakan fungsi *newff* yang disediakan oleh Matlab. Dengan menggunakan struktur jaringan syaraf, data *input* dan target yang

sama seperti diatas, 1000 iterasi dapat dilakukan hanya dalam hitungan menit. Pelatihan dan pengujian jaringan syaraf menggunakan Matlab akan lebih cepat jika semua data *input*, *output* dan bobot-bias dijadikan kedalam bentuk perhitungan matrik seperti yang terdapat pada fungsi *newff*.

4.1 Pelatihan Jaringan Syaraf Tiruan *Backpropagation*

Agar jaringan syaraf tiruan dapat menghasilkan pelatihan yang lebih cepat, dan menghasilkan *mean square error* yang kecil maka perlu dilakukan penyesuaian data *input* dan *output* pelatihan. Dari data yang dihasilkan pada pencarian data untuk pelatihan didapatkan data yang belum bersesuaian antara *input* dan *output* pelatihan. Hal ini terjadi karena *input* data pelatihan diambil dari perubahan suhu pada tangki dan perubahan aliran, sedangkan *output* adalah perubahan posisi katup. Untuk menyesuaikan data pelatihan maka pada *output* atau target pelatihan diberikan delay. Nilai *input* pelatihan bernilai kecil ini bertujuan agar saat *input* bernilai lebih besar dari *input* data pelatihan maka *outputnya* adalah bernilai *output* maksimalnya. Nilai *output* bernilai antara -1 dan 1, untuk mendapatkan nilai *output* maksimal data *output* pelatihan. Sedangkan pada bobot dan bias awal pelatihan, fungsi *newff* pada Matlab nilai bobot dan bias telah di normalisasi dengan metode *nguyen-widrow*. Dibawah ini adalah persamaan untuk normalisasi bobot awal pelatihan dengan metode *nguyen-widrow*

$$\beta = 0.7(p)^{-1/n} = 0.7\sqrt[n]{p} \quad (4.1)$$

$$v_{ij} = \frac{\beta v_{ij}(old)}{\|v_{ij}(old)\|} \quad (4.2)$$

$$v_{oj} = -\beta \text{ and } \beta \quad (4.3)$$



Keterangan :

β : Faktor skala

n : Banyaknya sel masukan

p : Banyaknya sel lapisan tersembunyi

v_{ij} : Bobot yang sudah ternormalisasi

v_{oj} : Bias yang sudah ternormalisasi

$v_{ij}(old)$: Bobot awal, nilai acak antara -0.5 sampai 0.5

Pada pelatihan jaringan syaraf tiruan sebagai pengendali suhu air pada tangki, dilakukan beberapa pengamatan yaitu jumlah fungsi aktivasi, nilai *learning rate*, jumlah lapisan tersembunyi dan jumlah sel neuron pada setiap lapisan tersembunyi yang digunakan.

4.1.1 Pelatihan Menggunakan 1 Lapisan Tersembunyi

Berikut ini adalah data hasil pengamatan dari pelatihan dengan menggunakan 1 lapisan tersembunyi dan dikelompokkan kedalam tabel-tabel sesuai banyaknya neuron yang digunakan pada lapisan tersembunyi.

Penjelasan tabel hasil pelatihan dan tabel hasil pengujian :

No : Nomer.

HL : Jumlah lapisan tersembunyi pada jaringan syaraf (*hidden layer*).

Neuron : Angka pertama adalah jumlah neuron lapisan masukan, angka berikutnya adalah jumlah neuron lapisan tersembunyi. (angka 1 dibelakang menunjukkan jaringan terdiri dari 1 target)

LR : Nilai kecepatan pelatihan (*learning rate*).

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

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

MSE : Nilai rata-rata *error* kuadrat (*mean square error*).

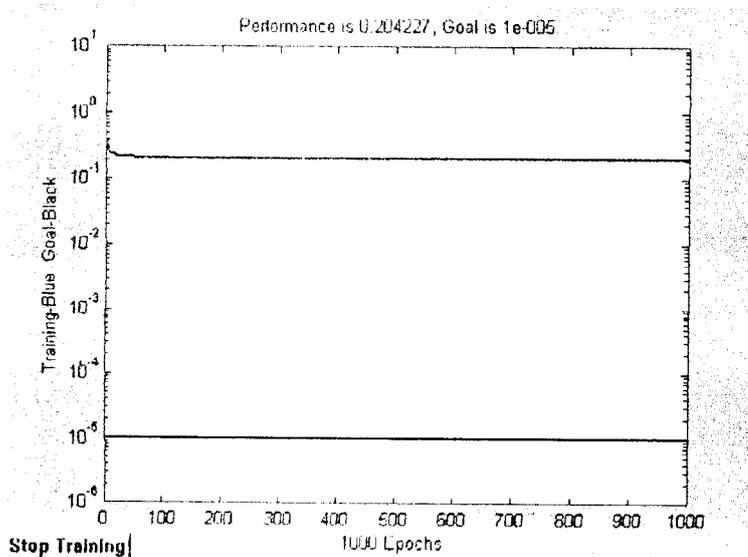
Ket : Keterangan dari berhentinya pelatihan.

Cetak tebal : Hasil pelatihan terbaik dalam satu tabel perbandingan.

4.1.1.1 Berikut ini hasil pelatihan dengan input yang digeser :

Tabel 4.1 Hasil pelatihan dengan 5 neuron dengan berbagai macam fungsi aktivasi

No	HL	Neuron	LR	F. aktivasi	Iterasi	MSE	Ket
1	1	5 – 2	0.7	<i>tansig tansig</i>	1000	0.22710	iterasi
2	1	5 – 2	0.7	<i>tansig logsig</i>	1000	0.30049	iterasi
3	1	5 – 2	0.7	<i>tansig – purelin</i>	1000	0.20422	iterasi
4	1	5 – 2	0.7	<i>logsig purelin</i>	1000	0.20732	iterasi



Gambar 4.1 Hasil pelatihan dengan 1 lapisan tersembunyi, 5 sel neuron, learning rate 0.7

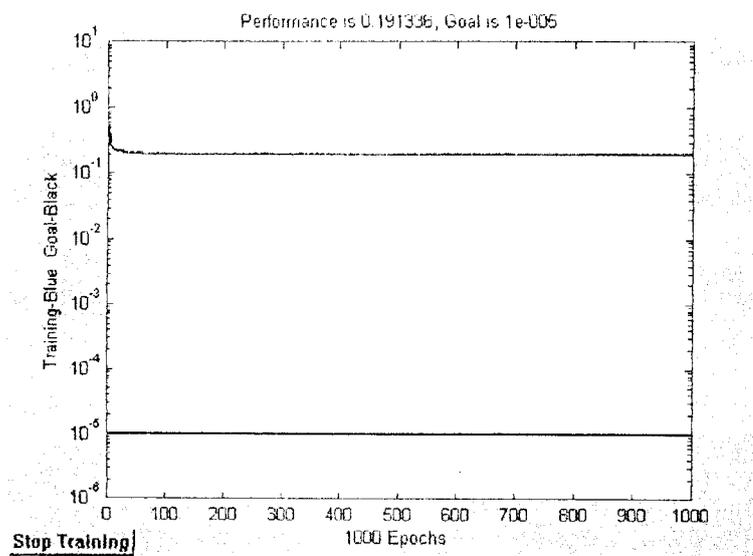
TRAINLM, Epoch 1000/1000, MSE 0.204227/1e-005, Gradient 0.00512777/1e-010

TRAINLM, Maximum epoch reached, performance goal was not met.

Target MSE tidak tercapai, pelatihan berhenti karena target iterasi sudah tercapai.

Tabel 4.2 Hasil pelatihan dengan 7 sel neuron dengan berbagai fungsi aktivasi

No.	HL	Neuron	LR	F. aktivasi	Iterasi	MSE	Ket.
1	1	7-2	0.5	<i>tansig-purelin</i>	1000	0.19133	iterasi
2	1	7-2	0.5	<i>tansig-tansig</i>	1000	0.20290	iterasi
3	1	7-2	0.5	<i>tansig-logsig</i>	1000	0.30898	iterasi



Gambar 4.2 Hasil pelatihan dengan 1 lapisan tersembunyi, 7 sel neuron, learning rate 0.5

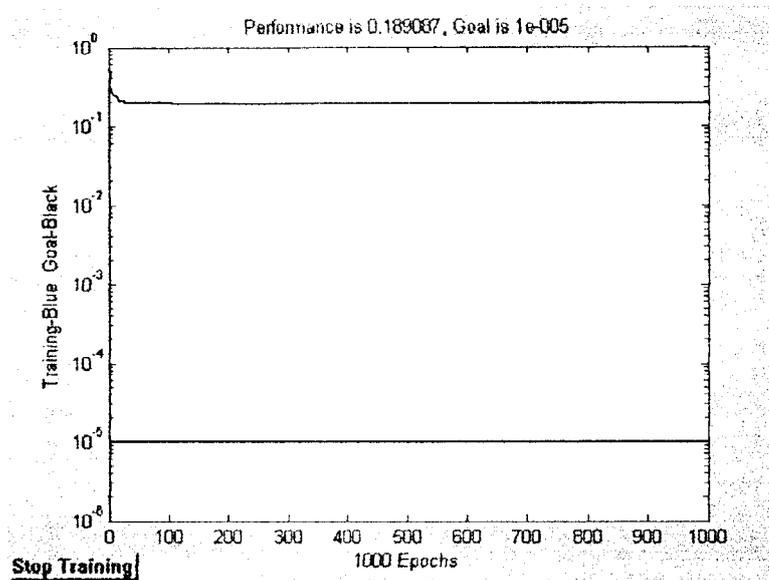
TRAINLM, Epoch 1000/1000, MSE 0.191336/1e-005, Gradient 0.000553606/1e-010

TRAINLM, Maximum epoch reached, performance goal was not met.

Target MSE tidak tercapai, pelatihan berhenti karena target iterasi sudah tercapai.

Tabel 4.3 Hasil pelatihan dengan fungsi *sigmoid* bipolar – *sigmoid* bipolar

No.	HL	Neuron	LR	F. aktivasi	Iterasi	MSE	Ket.
1	1	7 – 2	0.1	<i>tansig - tansig</i>	1000	0.18908	iterasi
2	1	7 – 2	0.4	<i>tansig - tansig</i>	1000	0.24101	iterasi
3	1	7 – 2	0.5	<i>tansig - tansig</i>	1000	0.20290	iterasi
4	1	7 – 2	0.7	<i>tansig - tansig</i>	1000	0.19116	iterasi



Gambar 4.3 Hasil pelatihan dengan 1 lapisan tersembunyi, 7 sel neuron, *learning rate* 0.1

TRAINLM, Epoch 1000/1000, MSE 0.189087/1e-005, Gradient 0.119563/1e-010

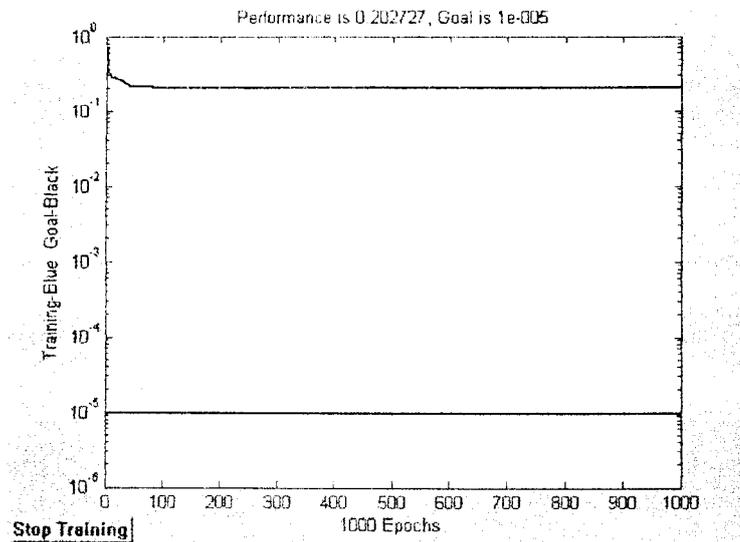
TRAINLM, Maximum epoch reached, performance goal was not met.

Target MSE tidak tercapai, pelatihan berhenti karena target iterasi sudah tercapai.

4.1.1.2 Berikut ini hasil pelatihan dengan input yang tidak digeser :

Tabel 4.4 Hasil pelatihan dengan 5 neuron dengan berbagai macam fungsi aktivasi

No	HL	Neuron	LR	F. aktivasi	Iterasi	MSE	Ket
1	1	5-2	0.7	<i>tansig - tansig</i>	1000	0.20272	iterasi
2	1	5-2	0.7	<i>tansig logsig</i>	1000	0.29640	iterasi
3	1	5-2	0.7	<i>tansig purelin</i>	1000	0.21996	iterasi
4	1	5-2	0.7	<i>logsig purelin</i>	1000	0.21432	iterasi



Gambar 4.4 Hasil pelatihan dengan 1 lapisan tersembunyi, 5 sel neuron, learning rate 0.7

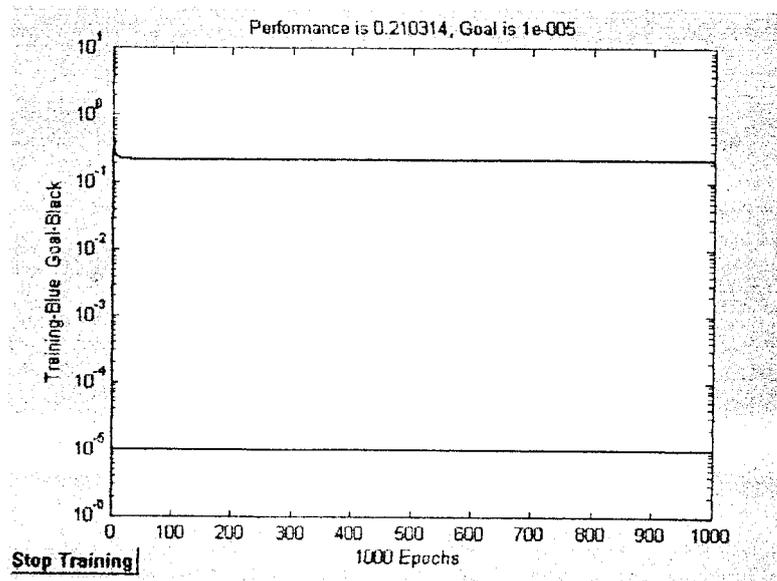
TRAINLM, Epoch 1000/1000, MSE 0.202727/1e-005, Gradient 2.97185/1e-010

TRAINLM, Maximum epoch reached, performance goal was not met.

Target MSE tidak tercapai, pelatihan berhenti karena target iterasi sudah tercapai.

Tabel 4.5 Hasil pelatihan dengan 7 sel neuron dengan berbagai fungsi aktivasi

No.	HL	Neuron	LR	F. aktivasi	Iterasi	MSE	Ket.
1	1	7-2	0.5	<i>tansig - purelin</i>	1000	0.21031	iterasi
2	1	7-2	0.5	<i>tansig tansig</i>	1000	0.69593	iterasi
3	1	7-2	0.5	<i>tansig logsig</i>	1000	0.45250	iterasi



Gambar 4.5 Hasil pelatihan dengan 1 lapisan tersembunyi, 7 sel neuron,
learning rate 0.5

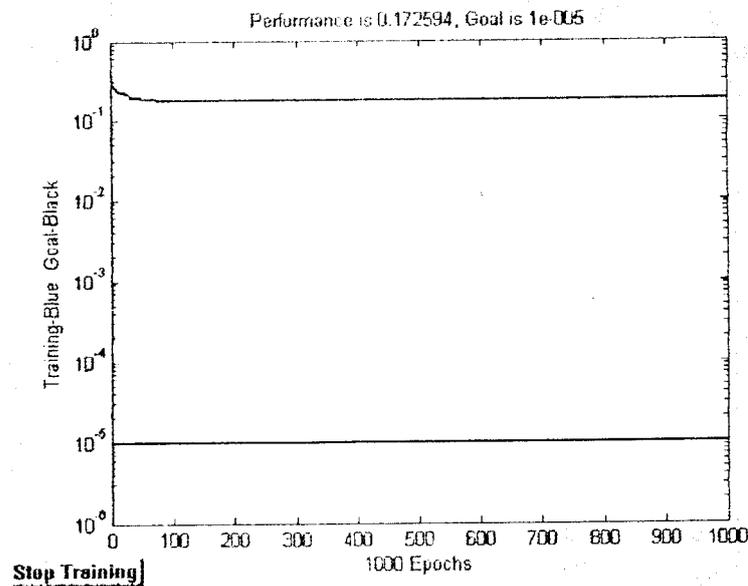
TRAINLM, Epoch 1000/1000, MSE 0.210314/1e-005, Gradient 0.146513/1e-010

TRAINLM, Maximum epoch reached, performance goal was not met.

Target MSE tidak tercapai, pelatihan berhenti karena target iterasi sudah tercapai.

Tabel 4.6 Hasil pelatihan dengan fungsi *sigmoid* bipolar -- *sigmoid* bipolar

No.	HL	Neuron	LR	F. aktivasi	Iterasi	MSE	Ket.
1	1	7 - 2	0.1	<i>tansig - tansig</i>	1000	0.18932	iterasi
2	1	7 - 2	0.4	<i>tansig - tansig</i>	1000	0.17259	iterasi
3	1	7 - 2	0.5	<i>tansig - tansig</i>	1000	0.58326	iterasi
4	1	7 - 2	0.7	<i>tansig - tansig</i>	1000	0.18861	iterasi



Gambar 4.6 Hasil pelatihan dengan 1 lapisan tersembunyi, 7 sel neuron, *learning rate* 0.4

TRAINLM, Epoch 1000/1000, MSE 0.172594/1e-005, Gradient 0.162241/1e-010
TRAINLM, Maximum epoch reached, performance goal was not met.

Target MSE tidak tercapai, pelatihan berhenti karena target iterasi sudah tercapai.

4.1.2 Pelatihan Menggunakan Lebih dari 1 lapisan Tersembunyi

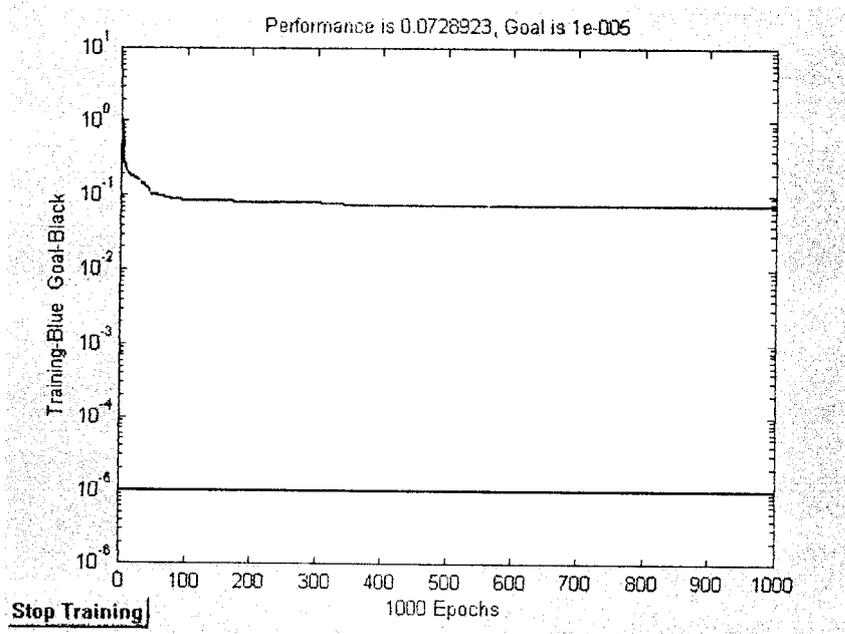
4.1.2.1 Berikut hasil pelatihan dengan input yang digeser :

Berikut ini adalah data hasil pengamatan dari pelatihan dengan menggunakan 2 lapisan tersembunyi dan dikelompokkan kedalam tabel-tabel sesuai fungsi aktivasi yang digunakan.

Tabel 4.7 Hasil pelatihan dengan fungsi aktivasi *sigmoid* bipolar –

sigmoid bipolar - *sigmoid* bipolar

No.	HL	Neuron	LR	F. aktivasi	Iterasi	MSE	Ket.
1	2	4 - 2 - 2	0.5	<i>tansig-tansig-tansig</i>	1000	0.79955	MSE
2	2	5 - 4 - 2	0.3	<i>tansig-tansig-tansig</i>	1000	0.17110	MSE
3	2	7 - 5 - 2	0.6	<i>tansig-tansig-tansig</i>	1000	1.22028	MSE
4	2	7 - 7 - 2	0.6	<i>tansig-tansig-tansig</i>	1000	0.07986	MSE
5	2	9 - 7 - 2	0.4	<i>tansig-tansig-tansig</i>	1000	0.07879	MSE
6	2	13 - 10 -	0.4	<i>tansig-tansig-</i>	1000	0.07289	MSE



Gambar 4.7 Hasil pelatihan dengan 2 lapisan tersembunyi, 13 - 10 sel neuron, *learning rate* 0.4

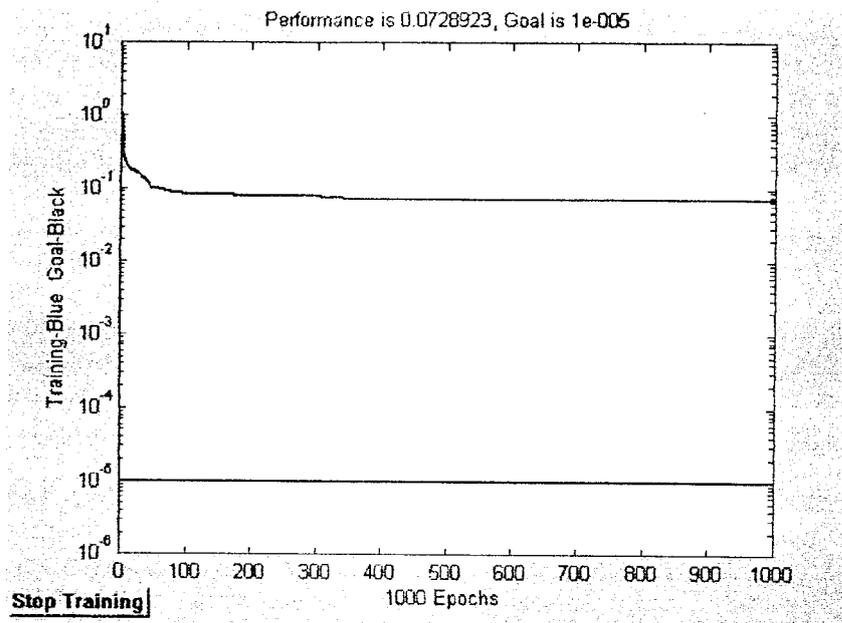
TRAINLM, Epoch 1000/1000, MSE 0.0728923/1e-005, Gradient 0.00149285/1e-010

TRAINLM, Maximum epoch reached, performance goal was not met.

Target MSE tidak tercapai, pelatihan berhenti karena target iterasi sudah tercapai.

Tabel 4.8 Hasil pelatihan dengan jaringan 13 – 10 – 2 dengan fungsi aktivasi *sigmoid bipolar – sigmoid bipolar - sigmoid bipolar*

No.	HL	Neuron	LR	F. aktivasi	Iterasi	MSE	Ket.
1	2	13-10-2	0.2	<i>tansig-tansig-tansig</i>	1000	0.07418	iterasi
2	2	13-10-2	0.4	<i>tansig-tansig-tansig</i>	1000	0.07289	iterasi
3	2	13-10-2	0.5	<i>tansig-tansig-tansig</i>	1000	0.07465	iterasi
4	2	13-10-2	0.6	<i>tansig-tansig-tansig</i>	1000	0.07417	Iterasi
5	2	13-10-2	0.7	<i>tansig-tansig-tansig</i>	1000	0.66244	iterasi



Gambar 4.8 Hasil pelatihan dengan 2 lapisan tersembunyi, 13 - 10 sel neuron, *learning rate 0.4*

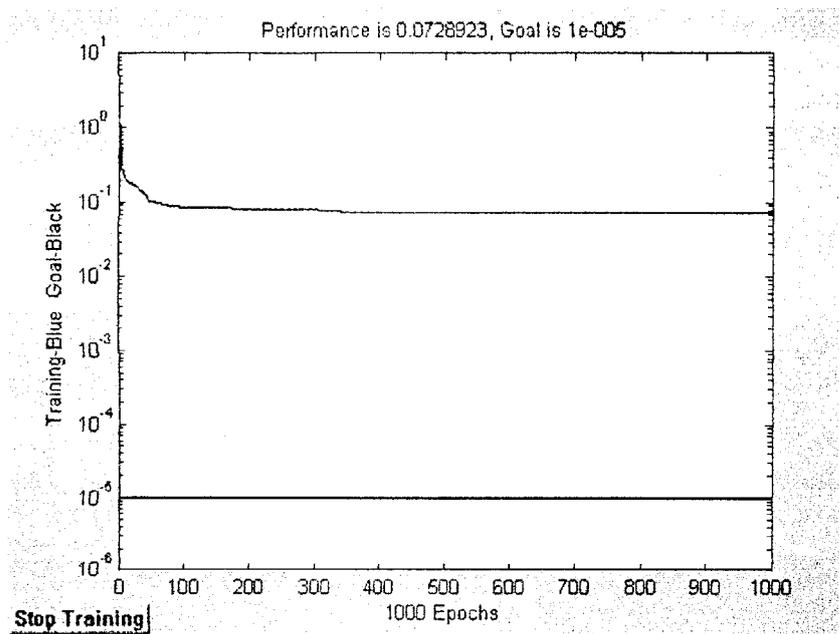
TRAINLM, Epoch 1000/1000, MSE 0.0728923/1e-005, Gradient 0.00149285/1e-010

TRAINLM, Maximum epoch reached, performance goal was not met.

Target MSE tidak tercapai, pelatihan berhenti karena target iterasi sudah tercapai.

Tabel 4.9 Hasil pelatihan dengan jaringan 13 – 10 – 2 dengan berbagai macam fungsi aktivasi

No.	HL	Neuron	LR	F. aktivasi	Iterasi	MSE	Ket.
1	2	13-10-2	0.4	<i>tansig-tansig-tansig</i>	1000	0.07289	iterasi
2	2	13-10-2	0.4	<i>tansig-purelin-urelin</i>	1000	0.11739	iterasi
3	2	13-10-2	0.4	<i>tansig-purelin-logsig</i>	1000	0.23950	iterasi
4	2	13-10-2	0.4	<i>purelin-logsig-tansig</i>	1000	0.17655	Iterasi
5	2	13-10-2	0.4	<i>purelin-tansig-tansig</i>	1000	0.12876	iterasi



Gambar 4.9 Hasil pelatihan dengan 2 lapisan tersembunyi, 13 - 10 sel neuron, *learning rate* 0.4

TRAINLM, Epoch 1000/1000, MSE 0.0728923/1e-005, Gradient 0.00149285/1e-010

TRAINLM, Maximum epoch reached, performance goal was not met.

Target MSE tidak tercapai, pelatihan berhenti karena target iterasi sudah tercapai.

Tabel 4.10 Hasil pelatihan dengan 3 - 4 lapisan tersembunyi dengan berbagai macam jumlah neuron

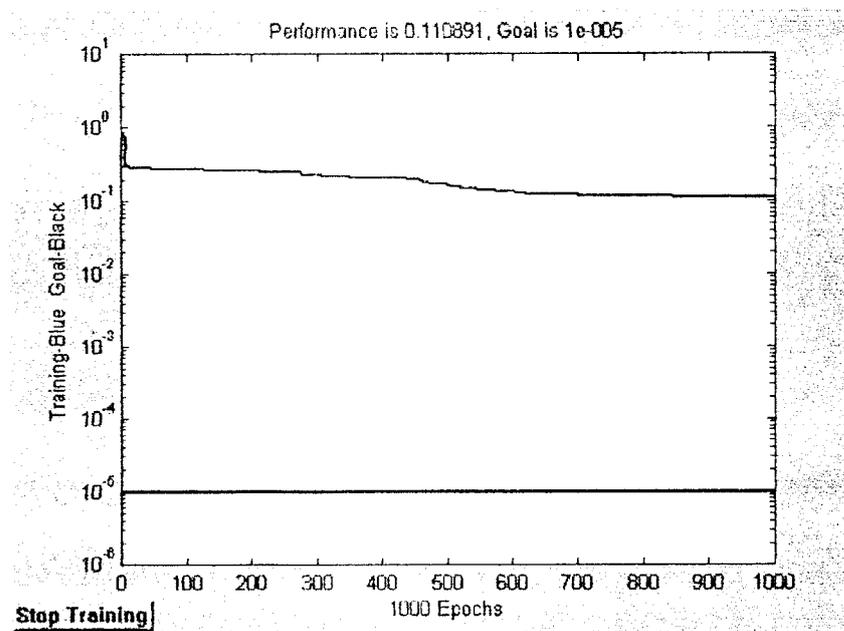
No.	HL	Neuron	LR	F. aktivasi	Iterasi	MSE	Ket.
1	3	5 - 5 - 3 - 2	0.5	<i>tansig</i> - <i>tansig</i> <i>tansig</i> <i>tansig</i>	1000	0.21059	iterasi
2	3	7 - 4 - 3 - 2	0.5	<i>tansig</i> <i>tansig</i> <i>tansig</i> <i>tansig</i>	1000	0.21026	iterasi
3	3	7 - 7 - 5 - 2	0.3	<i>tansig</i> <i>tansig</i> <i>tansig</i> <i>tansig</i>	1000	0.12727	iterasi
4	4	10 - 5 - 3 - 2	0.4	<i>tansig</i> <i>tansig</i> <i>tansig</i> - <i>tansig</i>	1000	0.12311	iterasi
5	4	7 - 7 - 5 - 4 - 2	0.6	<i>tansig</i> - <i>tansig</i> <i>tansig</i> - <i>tansig</i> <i>tansig</i>	1000	0.11089	iterasi

Untuk gambar hasil prlatihan dapat dilihat digambar no.4.10

TRAINLM, Epoch 1000/1000, MSE 0.110891/1e-005, Gradient 14.9909/1e-010

TRAINLM, Maximum epoch reached, performance goal was not met.

Target MSE tidak tercapai, pelatihan berhenti karena target iterasi sudah tercapai.



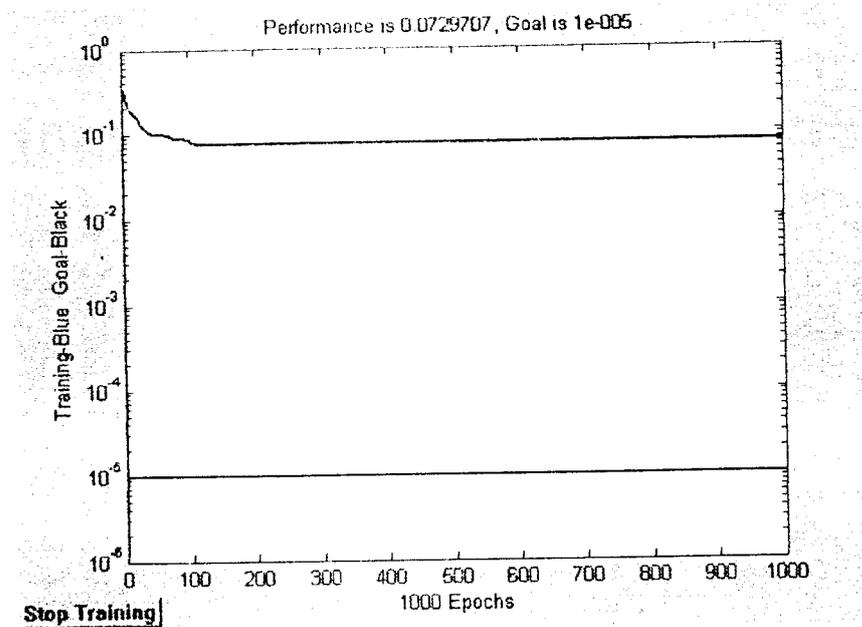
Gambar 4.10 Hasil pelatihan dengan 4 lapisan tersembunyi, 7 – 7 – 5 - 4 sel neuron, *learning rate* 0.6

4.1.2.2 Berikut hasil pelatihan dengan input yang tidak digeser :

Berikut ini adalah data hasil pengamatan dari pelatihan dengan menggunakan 2 lapisan tersembunyi dan dikelompokkan kedalam tabel-tabel sesuai fungsi aktivasi yang digunakan.

Tabel 4.11 Hasil pelatihan dengan fungsi aktivasi *sigmoid* bipolar – *sigmoid* bipolar - *sigmoid* bipolar

No.	HL	Neuron	LR	F. aktivasi	Iterasi	MSE	Ket.
1	2	4 - 2 - 2	0.5	<i>tansig-tansig-tansig</i>	1000	0.24953	MSE
2	2	5 - 4 - 2	0.3	<i>tansig-tansig-tansig</i>	1000	0.25267	MSE
3	2	7 - 5 - 2	0.6	<i>tansig-tansig-tansig</i>	1000	0.11547	MSE
4	2	7 - 7 - 2	0.6	<i>tansig-tansig-tansig</i>	1000	0.12479	MSE
5	2	10 - 8 -	0.6	<i>tansig-tansig-</i>	1000	0.07280	MSE
6	2	13 - 10-	0.4	<i>tansig-tansig-tansig</i>	1000	0.07808	MSE



Gambar 4.11 Hasil pelatihan dengan 2 lapisan tersembunyi, 10 - 8 sel neuron, *learning rate* 0.6

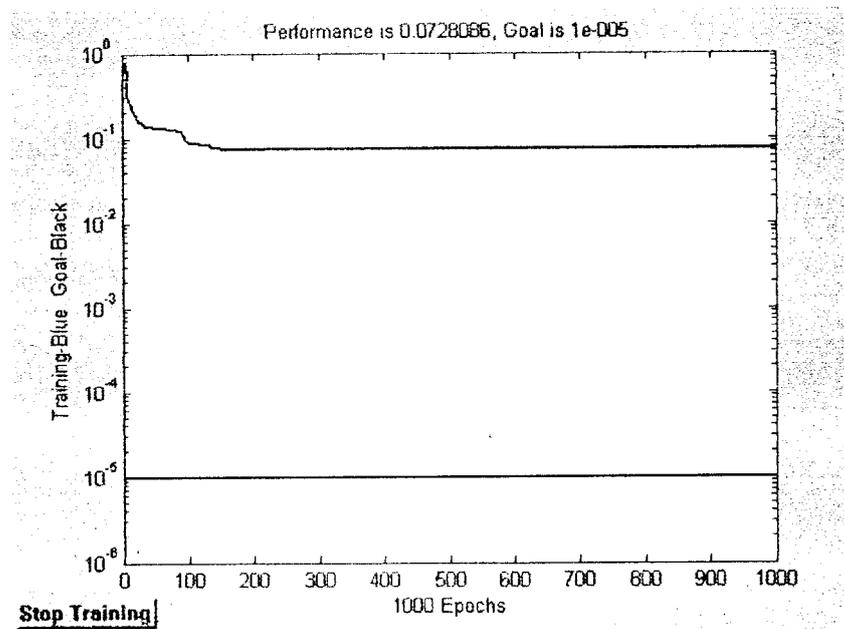
TRAINLM, Epoch 1000/1000, MSE 0.0729707/1e-005, Gradient 0.107485/1e-010

TRAINLM, Maximum epoch reached, performance goal was not met.

Target MSE tidak tercapai, pelatihan berhenti karena target iterasi sudah tercapai.

Tabel 4.12 Hasil pelatihan dengan jaringan 10 – 8 – 2 dengan fungsi aktivasi *sigmoid bipolar sigmoid bipolar - sigmoid bipolar*

No.	HL	Neuron	LR	F. aktivasi	Iterasi	MSE	Ket.
1	2	10- 8 -2	0.2	<i>tansig-tansig-tansig</i>	1000	0.08005	iterasi
2	2	10- 8 -2	0.4	<i>tansig-tansig-tansig</i>	1000	0.07297	iterasi
3	2	10- 8 -2	0.5	<i>tansig-tansig-tansig</i>	1000	0.07281	iterasi
4	2	10- 8 -2	0.6	<i>tansig-tansig-tansig</i>	1000	0.07280	Iterasi
5	2	10- 8 -2	0.7	<i>tansig-tansig-tansig</i>	1000	0.07440	iterasi



Gambar 4.12 Hasil pelatihan dengan 2 lapisan tersembunyi, 10 - 8 sel neuron,
learning rate 0.6

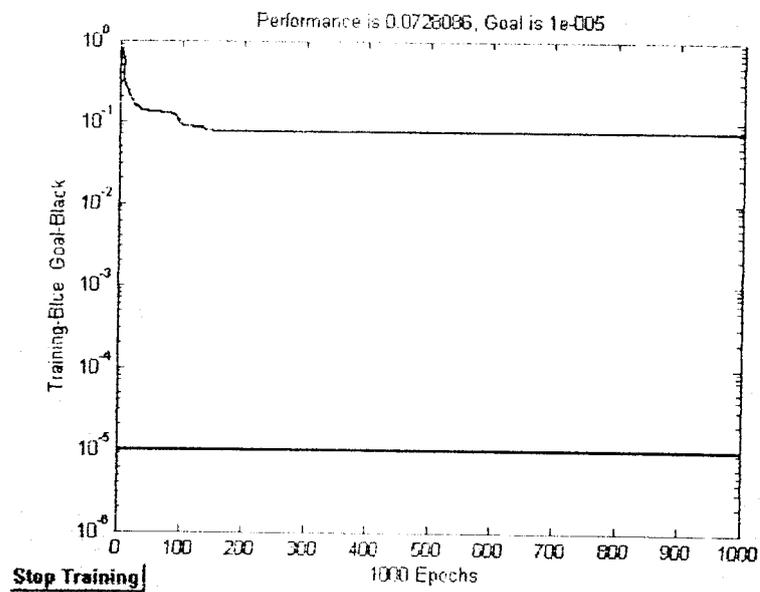
TRAINLM, Epoch 1000/1000, MSE 0.0728086/1e-005, Gradient 0.0236802/1e-010

TRAINLM, Maximum epoch reached, performance goal was not met.

Target MSE tidak tercapai, pelatihan berhenti karena target iterasi sudah tercapai.

Tabel 4.13 Hasil pelatihan dengan jaringan 10 – 8 – 2 dengan berbagai macam fungsi aktivasi

No.	HL	Neuron	LR	F. aktivasi	Iterasi	MSE	Ket.
1	2	10- 8 -2	0.6	<i>tansig-tansig-tansig</i>	1000	0.07280	iterasi
2	2	10- 8 -2	0.6	<i>tansig-purelin-</i>	1000	0.13924	iterasi
3	2	10- 8 -2	0.6	<i>tansig-purelin-logsig</i>	1000	0.84432	iterasi
4	2	10- 8 -2	0.6	<i>purelin-logsig-tansig</i>	1000	0.20110	Iterasi
5	2	10 -8 -2	0.6	<i>purelin-tansig-tansig</i>	1000	0.17008	iterasi



Gambar 4.13 Hasil pelatihan dengan 2 lapisan tersembunyi, 10 - 8 sel neuron, *learning rate* 0.6

TRAINLM, Epoch 1000/1000, MSE 0.0728923/1e-005, Gradient 0.00149285/1e-010

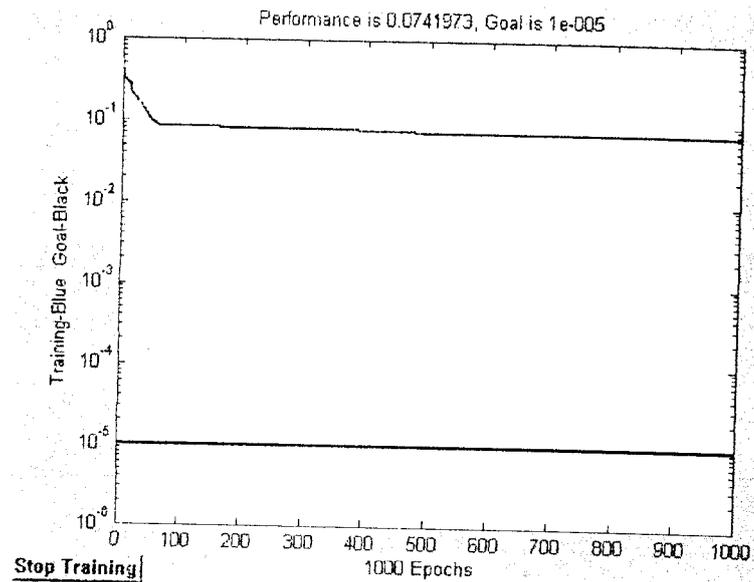
TRAINLM, Maximum epoch reached, performance goal was not met.

Target MSE tidak tercapai, pelatihan berhenti karena target iterasi sudah tercapai.

Tabel 4.14 Hasil pelatihan dengan 3 - 4 lapisan tersembunyi dengan berbagai macam jumlah neuron

No.	HL	Neuron	LR	F. aktivasi	Iterasi	MSE	Ket.
1	3	5 - 5 - 3 - 2	0.5	<i>tansig tansig</i> <i>tansig - tansig</i>	1000	0.31997	iterasi
2	3	7 - 4 - 3 - 2	0.5	<i>tansig tansig</i> <i>tansig - tansig</i>	1000	0.20199	iterasi

3	3	7 - 7 - 5 - 2	0.3	<i>tansig</i> – <i>tansig</i> <i>tansig</i> – <i>tansig</i>	1000	0.08054	iterasi
4	3	10 - 5 - 3 - 2	0.4	<i>tansig</i> – <i>tansig</i> <i>tansig</i> – <i>tansig</i>	1000	0.07419	iterasi
5	4	7 - 7 - 5 - 4 - 2	0.6	<i>tansig</i> – <i>tansig</i> <i>tansig</i> – <i>tansig</i> <i>tansig</i>	1000	0.80696	iterasi



Gambar 4.14 Hasil pelatihan dengan 3 lapisan tersembunyi, 10 – 5 - 3 sel neuron, *learning rate* 0.4

TRAINLM, Epoch 1000/1000, MSE 0.0741973/1e-005, Gradient 1.18785/1e-010

TRAINLM, Maximum epoch reached, performance goal was not met

Target MSE tidak tercapai, pelatihan berhenti karena target iterasi sudah tercapai.

Berikut ini adalah hasil pengamatan dari pelatihan dengan menggunakan berbagai lapisan tersembunyi dan dikelompokkan kedalam tabel-tabel sesuai pelatihan terbaik dari tiap-tiap tabel diatas .

Tabel 4.15 Hasil pelatihan terbaik dari masing – masing pengelompokan pada input yang digeser

No.	HL	Neuron	LR	F. aktivasi	Iterasi	MSE	Ket.
1	1	5 – 2	0.7	<i>tansig purelin</i>	1000	0.20422	Iterasi
2	1	7 – 2	0.5	<i>tansig - tansig</i>	1000	0.19133	Iterasi
3	1	7 – 2	0.1	<i>tansig - tansig</i>	1000	0.18908	Iterasi
4	2	13 – 10 -2	0.4	<i>tansig-tansig- tansig</i>	1000	0.72892	iterasi
5	4	7 – 7 - 5 - 4 -2	0.6	<i>tansig-tansig- tansi -tansig</i>	1000	0.11089	iterasi

Tabel 4.16 Hasil pelatihan terbaik dari masing – masing pengelompokan dengan input yang tidak digeser

No.	HL	Neuron	LR	F. aktivasi	Iterasi	MSE	Ket.
1	1	5 – 2	0.7	<i>logsig purelin</i>	1000	0.20272	Iterasi
2	1	7 – 2	0.5	<i>tansig - logsig</i>	1000	0.21031	Iterasi

3	1	7 - 2	0.4	<i>tansig - tansig</i>	1000	0.17259	Iterasi
4	2	10 - 8 - 2	0.6	<i>tansig-tansig- tansig</i>	1000	0.07280	iterasi
5	3	10 - 5 - 3 - 2	0.4	<i>tansig-tansig- logsig</i>	1000	0.74197	iterasi

Dari tabel 4.15 diatas, untuk input yang digeser diperoleh hasil pelatihan terbaik adalah pelatihan pada tabel nomor 4. Pelatihan tersebut berhenti ketika iterasi yang telah ditentukan sudah tercapai dengan MSE 0.0728923. Sehingga untuk pengujian jaringan akan digunakan struktur yang sama dengan hasil pelatihan pada tabel nomer 4 tersebut. Dari tabel 4.16 diatas, untuk input yang tidak digeser diperoleh hasil pelatihan terbaik adalah pelatihan pada tabel nomor 4. Pelatihan tersebut berhenti ketika iterasi yang telah ditentukan sudah tercapai dengan MSE 0.0728086. Sehingga untuk pengujian jaringan akan digunakan struktur yang sama dengan hasil pelatihan pada tabel nomer 4 tersebut.

Pada tabel semua pelatihan tidak ada pelatihan yang berhenti karena *gradient* sudah mencapai target, jika terdapat pelatihan yang berhenti karena *gradient* sudah tercapai artinya MSE yang dihasilkan sudah mencapai nilai yang paling minimum untuk arsitektur jaringan syaraf sebagai pengendali suhu shower dengan 1 lapisan tersembunyi. Nilai *gradient* menggunakan nilai *default* yang di tentukan fungsi *newff* yaitu $1e-10$. Nilai *gradient* yang dihasilkan dan ditampilkan akan selalu dipengaruhi oleh perubahan dari permukaan MSE. Jika diamati dari hasil pelatihan diatas maka akan menemukan pelatihan dengan fungsi aktivasi

sigmoid bipolar (*tansig*) menghasilkan nilai MSE yang lebih kecil dibandingkan dengan menggunakan fungsi aktivasi yang lain. Fungsi aktivasi *sigmoid* bipolar memiliki nilai *range output* antara 1 sampai -1, dengan *range output* seperti itu maka nilai keluarannya dapat digunakan sebagai pengendali katup. Oleh karena itu sangat diutamakan saat pelatihan menggunakan fungsi aktivasi *sigmoid* bipolar, dan dari hasil pelatihannya terlihat bahwa dengan menggunakan fungsi aktivasi *sigmoid* bipolar akan menghasilkan nilai MSE yang lebih kecil. Jumlah lapisan dan sel neuron pada masing-masing lapisan tersembunyi berpengaruh besar terhadap nilai MSE yang dihasilkan, dan juga sangat dipengaruhi oleh *learning rate* dan *momentum coefficient* yang digunakan sesuai dengan arsitektur jaringan syaraf

Nilai *learning rate* akan berpengaruh terhadap perubahan MSE pada setiap iterasi. Semakin besar nilai *learning rate*, akan semakin cepat pelatihan mendekati nilai *error* minimum, tetapi menghasilkan perubahan MSE yang tidak stabil. Jika nilai *learning rate* digunakan terlalu kecil, maka akan menyebabkan pelatihan lebih lama mendekati nilai *error* minimum yang ditentukan dan iterasi semakin besar.

Penentuan jumlah target iterasi (*epoch*) dilihat dari struktur pelatihan jaringan. Jika jaringan memiliki lapisan tersembunyi dan jumlah neuron yang banyak, maka target iterasi di set tidak terlalu besar agar pelatihan tidak terlalu menggunakan memori pada PC (*personal computer*) terlalu banyak. Semakin banyak jumlah lapisan dan jumlah sel neuron pada masing-masing lapisan, komputasi-pun akan semakin banyak. Semakin banyak komputasi, semakin besar

n ditentu
melakuka

ate Mome



data input



data input

syaraf tir

e Momentu



data input



ta input

af tiruan

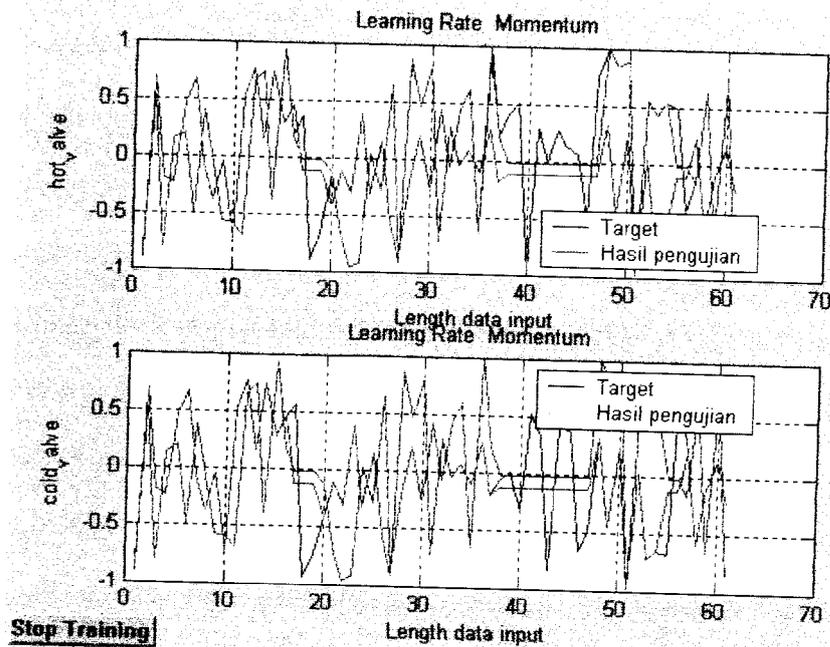
memori PC akan digunakan dan akan semakin lama waktu yang ditempuh untuk mencapai *error* minimum.

Dari tabel pelatihan 4.8, maka struktur jaringan yang akan digunakan adalah struktur jaringan pada tabel nomor 2. Dimana struktur jaringan terdiri dari 2 sel neuron *input*. *Input* pertama adalah *error* aliran yang dihasilkan dari selisih antara *set point* dengan keluaran dari tangki, sel neuron kedua adalah *error* suhu . Lapisan *input* (*v*) terdiri dari 13 sel neuron, sedangkan lapisan tersembunyi terdiri dari 2 lapisan. Lapisan tersembunyi pertama (*w*) terdiri dari 10 sel neuron dan sesuai dengan target sistem jaringan syaraf, maka lapisan tersembunyi kedua atau lapisan *output* (*w_out*) terdiri dari 2 sel neuron. Fungsi aktivasi yang digunakan pada setiap lapisannya adalah fungsi *sigmoid* bipolar. Dan dari tabel pelatihan 4.12, maka untuk *input* yang tidak digeser akan digunakan struktur jaringan pada tabel nomor 4. Dimana struktur jaringan terdiri dari 2 sel neuron *input*. *Input* pertama dan kedua sama dengan *input* yang digunakan untuk pelatihan yang digeser yaitu *error* aliran dan *error* suhu. Lapisan input terdiri dari 10 sel neuron, sedangkan lapisan tersembunyi terdiri dari 2 lapisan. Lapisan tersembunyi pertama terdiri 8 sel neuron dan lapisan tersembunyi kedua terdiri dari 2 sel neuron yang juga merupakan lapisan *output*.

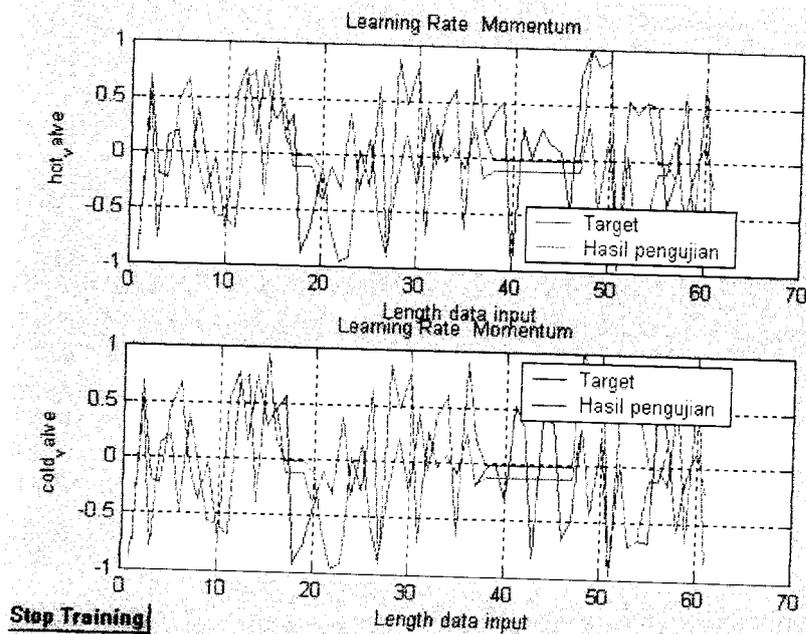
4.2 Pengujian Jaringan Syaraf Tiruan *Backpropagation*

Setelah mendapat struktur jaringan terbaik maka hal pertama yang akan dilakukan adalah menguji jaringan tersebut dengan tujuan untuk mengetahui apakah jaringan yang telah didapatkan dengan target *error* yang telah ditentukan

sudah dapat menyamai target yang telah ditentukan juga. Pengujian menggunakan *input* dan target yang sama seperti saat melakukan pelatihan mencari jaringan.



Gambar 4.15 Hasil pengujian jaringan syaraf tiruan untuk input yang digeser

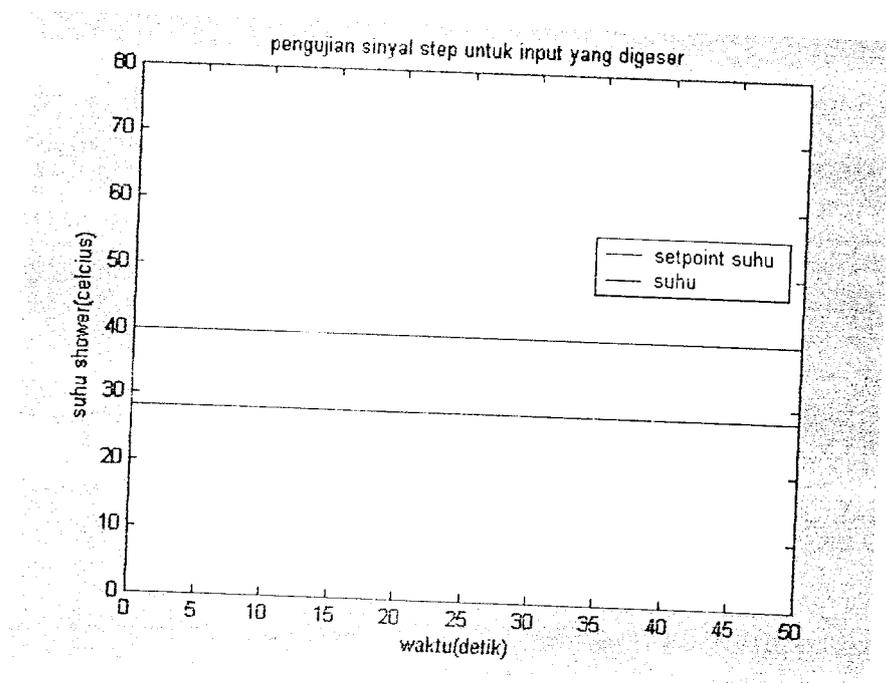


Gambar 4.16 Hasil pengujian jaringan syaraf tiruan untuk input yang tidak digeser



Dari gambar diatas dapat terlihat jelas bahwa struktur jaringan yang telah didapatkan kurang baik. Hal ini dibuktikan dengan hasil pengujian yang (warna biru) kurang dapat mengikuti dengan target (warna merah) yang telah ditentukan. Dari gambar pengujian diatas terlihat bahwa saat medapatkan *input* -1 posisi katup dalam keadaan nol atau tertutup. Dengan didapatkan struktur jaringan yang ini diharapkan jaringan ini dapat mengontrol suhu shower meskipun kurang baik dan mendapatkan *error* yang belum optimal.

Struktur terbaik jaringan pelatihan seperti yang telah disebutkan di atas, kemudian di uji dengan menggunakan masukan *step*. Berikut adalah grafik pengujiannya dengan target $40^{\circ}C$:



Gambar 4.17 Hasil pengujian menggunakan *step input* yang digeser

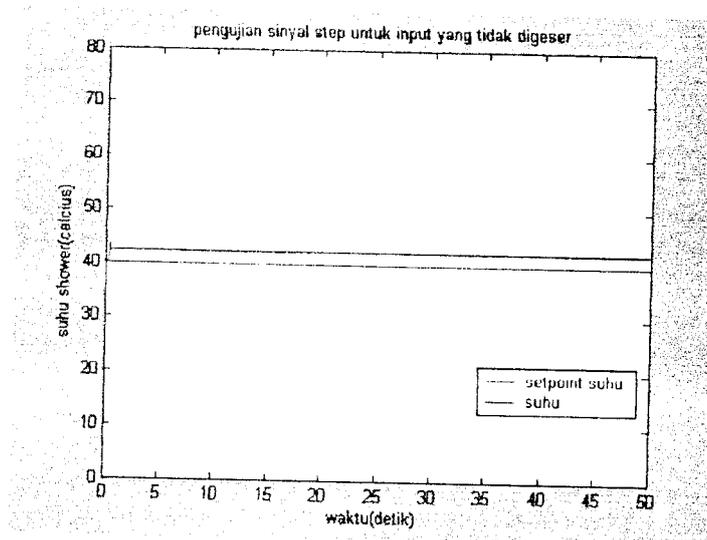
Dari gambar 4.17 diatas, dapat diketahui bahwa waktu (t) adalah dalam satuan detik Matlab, bukan detik dalam *real time*. Karakteristik *respon transien*

dari jaringan syaraf sebagai pengendali suhu shower dengan menggunakan *step input* adalah sebagai berikut :

- Waktu tunda (t_d) adalah waktu yang diperlukan oleh tanggapan untuk mencapai setengah (50%) dari nilai akhirnya yaitu selama 0.5 detik.
- Waktu naik (t_r) adalah waktu yang diperlukan oleh tanggapan untuk naik dari 0 % menjadi 100 % dari nilai akhir, karena grafik pengujian bernilai menurun atau nilainya semakain kecil, maka waktu naik tidak ada.
- Maksimum *overshoot* (m_p) adalah nilai puncak kurva tanggapan diukur dari satuan waktu, digunakan untuk mengukur kestabilan relatif dari sistem. Karena pada pengujian tidak terjadi *overshoot* maka tidak ada nilai maksimum *overshoot* (m_p).
- Waktu puncak adalah waktu yang diperlukan tanggapan untuk mencapai puncak atau maksimum *overshoot*. Karena tidak terjadi *Overshoot* maka waktu puncak tidak ada.
- Waktu turun (t_s) adalah waktu yang diperlukan untuk menanggapi kurva agar dapat mencapai dan tetap berada dalam persentase nilai akhir tertentu dan biasanya digunakan batasan 5 %. Seperti terlihat pada gambar 4.19 diatas, maka waktu turun adalah 0,28 detik.

Dari gambar 4.18 dibawah ini, dapat diketahui bahwa waktu (t) adalah dalam satuan detik Matlab, bukan detik dalam *real time*. Karakteristik *respon transien* dari jaringan syaraf sebagai pengendali suhu shower dengan menggunakan *step input* adalah sebagai berikut :

- Waktu tunda (t_d) adalah waktu yang diperlukan oleh tanggapan untuk mencapai setengah (50%) dari nilai akhirnya yaitu selama 0,61 detik.
- Waktu naik (t_r) adalah waktu yang diperlukan oleh tanggapan untuk naik dari 0 % menjadi 100 % dari nilai akhir yaitu selama 1,17 detik.
- Maksimum *overshoot* (m_p) adalah nilai puncak kurva tanggapan diukur dari satuan waktu, digunakan untuk mengukur kestabilan relatif dari sistem. Pada grafik besarnya *overshoot* adalah 3,44.
- Waktu puncak adalah waktu yang diperlukan tanggapan untuk mencapai puncak atau maksimum *overshoot*. Karena besarnya *overshoot* adalah 3,45 maka waktu puncak (T_p) adalah 1,045 detik.
- Waktu turun (t_s) adalah waktu yang diperlukan untuk menanggapi kurva agar dapat mencapai dan tetap berada dalam persentase nilai akhir tertentu dan biasanya digunakan batasan 5 %. Seperti terlihat pada gambar 4.12 diatas, maka waktu turun adalah 0,015 detik.



Gambar 4.18 Hasil pengujian menggunakan *step input* yang tidak digeser

Dari gambar hasil pengujian menggunakan step untuk input yang tidak digeser terlihat adanya *overshoot*. *Overshoot* terjadi karena JST yang digunakan dalam pengujian belum mencapai error minimal, hal itu menyebabkan pada saat pengujian dengan target tertentu posisi katup yang terbuka melebihi dari posisi katup yang seharusnya. Misalnya diambil *set point* 40 seperti pada pengujian diatas dan dari gambar diatas didapatkan nilai *Overshoot* sebesar 3.46.

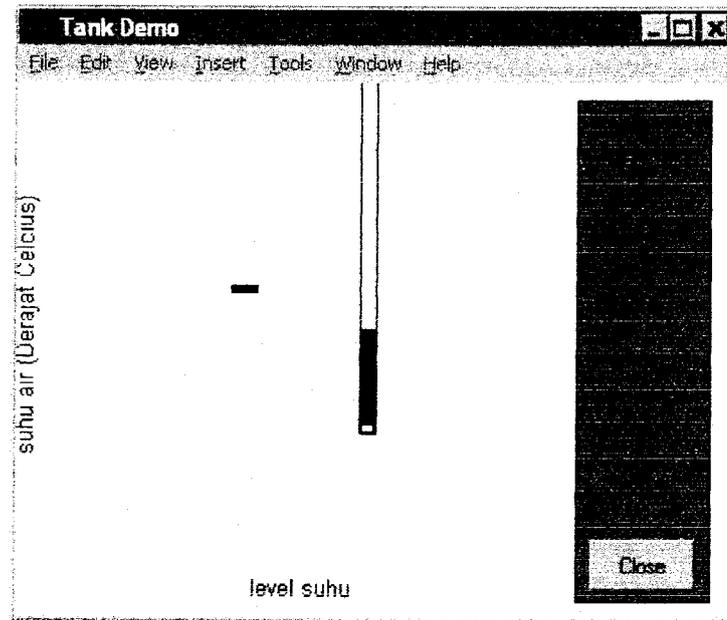
Dari hasil pelatihan diperoleh bobot yang berbeda beda antara input yang digeser dan yang tidak digeser. Hal ini dikarenakan jumlah sel neuron pada tiap lapisan tersembunyi, *learning rate* nya berbeda – beda.

Setelah mendapatkan keluaran JST maka nilai tersebut digunakan sebagai *input* katup. Pengujian *online* dilakukan menggunakan *toolbox simulink* pada Matlab dan pengujian menggunakan GUI (*Graphical User Interface*).

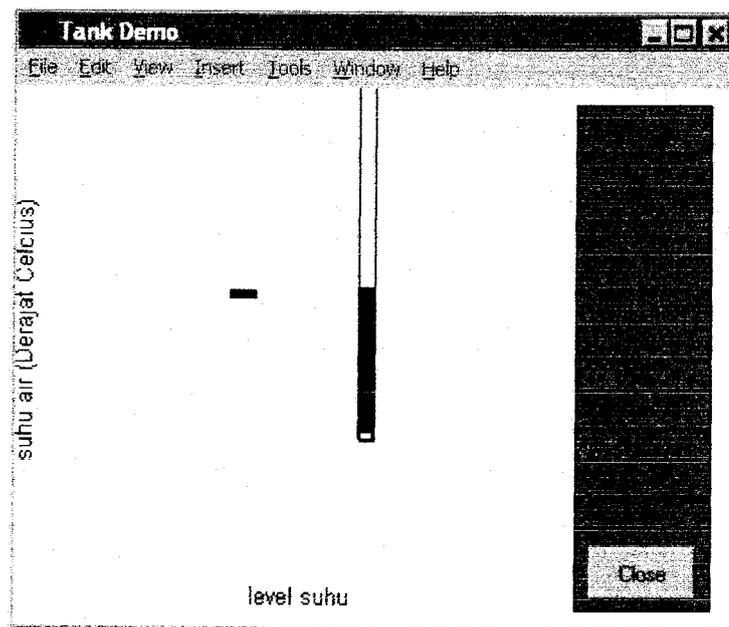
Pengujian *online* dilakukan dengan memberikan *set point* berupa suhu dan menghasilkan keluaran jaringan berupa nilai antara -1 sampai 1, nilai inilah yang akan memberi masukan pada katup sehingga katup dapat terbuka untuk menghasilkan aliran air dan air ini akan bercampur sehingga akan menghasilkan suhu. Kemudian suhu yang dihasilkan di-*feedback* menjadi *input* jaringan kedua. input pertama diperoleh dari perubahan aliran yang didapat dari setpoint aliran dikurangi *feedback*nya.

Set point yang berupa suhu diinginkan di masukan melalui blok *step*. suhu dihasilkan belum optimal mengikuti suhu dari *set point*. Ini disebabkan karena pada saat suhu mendekati *set point*. Selain dengan tampilan grafik, suhu air juga ditampilkan dalam bentuk gambar animasi sederhana yang hanya menampilkan

proses perubahan suhu saja tanpa menampilkan aliran. Berikut ini adalah tampilan gambar animasinya dengan suhu sebesar 40 °C



Gambar 4.19 tampilan animasi suhu untuk input yang digeser

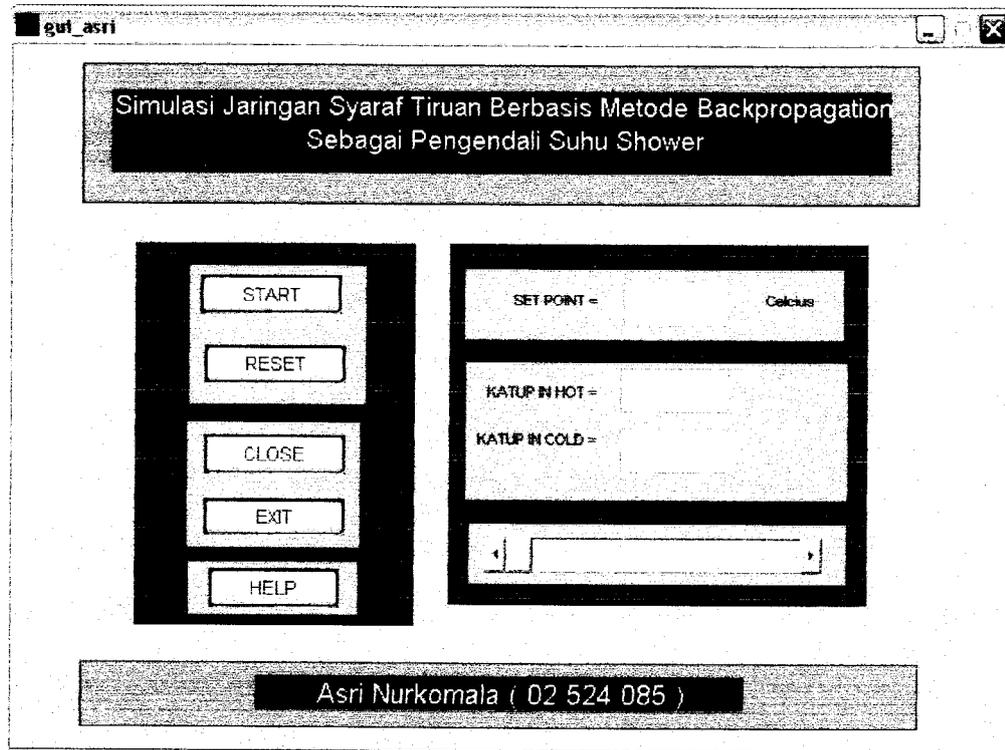


Gambar 4.20 tampilan animasi suhu untuk input yang tidak digeser



Pada simulasi menggunakan *Graphical User Interface* (GUI), *set point* suhu air di masukan pada kotak *edit set point* atau dengan cara mengatur masukannya melalui *slider* diatas kotak *edit set point*. Untuk menjalankan simulasi dapat dilakukan dengan cara menekan *button START*. Hasil dari pengujian akan ditampilkan pada kotak *edit katup input*.

Gambar dari tampilan GUI dapat dilihat pada gambar dibawah ini :



Gambar 4.21 Tampilan pengujian menggunakan GUI

Set point pada GUI adalah masukan pada jaringan syaraf yaitu berupa ketinggian air yang di inginkan. Sedangkan keluaran jaringan syaraf adalah masukan pada katup yaitu berupa nilai posisi katup dimana nilai -1 katup dalam kondisi tertutup dan 1 katup dalam kondisi terbuka. Untuk memulai simulasi tinggal menekan *botton START*, sedangkan *Button RESET* akan berfungsi

mengembalikan keadaan *set point*, *katup in* dan *slider* pada nilai awal menjadi sama dengan nol. *Button HELP* merupakan menu panduan dan bantuan untuk melakukan simulasi dengan menggunakan GUI. *Button CLOSE* berfungsi untuk menutup tampilan GUI, sedangkan *Button EXIT* berfungsi menutup semua tampilan *window* yang ada pada Matlab atau keluar dari program Matlab. Karena tampilan GUI mengendalikan simulasi dengan *simulink*, maka saat simulasi dijalankan, tampilan perbandingan antara suhu *set point* dan suhu keluaran dari tangki akan ditampilkan berupa grafik yang dapat di lihat dengan cara *men-double* klik blok *scope* paling kanan pada *simulink*.

Berbeda dengan pengujian menggunakan *simulink*, masukan nilai *set point* di masukan melalaui blok *step* dan jaringan diolah dengan blok-blok terpisah yang dijadikan satu kedalam blok *subsystem*. Berikut ini adalah tabel perbandingan hasil pengujian menggunakan *simulink*, dimana nilai *input* suhu dipilih atau ditentukan secara acak.

Tabel 4.17 Perbandingan suhu air *set point* dengan suhu air pada tangki

Untuk input yang digeser

No	Masukan(m)	Keluaran(m)	Selisih (m)
1	30	42.3	-12.3
2	40	28.3	11.7
3	45	31.4	13.6
4	50	30.2	19.8
5	60	35	25

Tabel 4.18 Perbandingan suhu air *set point* dengan suhu air pada tangki

Untuk input yang tidak digeser

No	Masukan(m)	Keluaran(m)	Selisih (m)
1	30	32.3	-2.3
2	40	42.3	-2.3
3	45	35	5
4	50	37.7	12.3
5	60	42.3	17.7

Dari tabel 4.17 dan 4.18 diatas, dengan 5 data *masukan* suhu air yang diambil secara acak, diperoleh rata-rata selisih suhu air sebesar 11.56 derajat celsius untuk input yang digeser dan 6.08 derajat celsius untuk input yang tidak digeser.

$$selisih\ rata - rata = \frac{\sum selisih}{banyaknya\ data}, \tag{4.4}$$

Hasil penelitian ini membuktikan bahwa simulasi jaringan syaraf tiruan sebagai kendali suhu shower kurang baik, dengan persentase *mean square error* pengujian sebesar 0.44. Persentase ini jika dibandingkan dengan nilai *mean square error* hasil pelatihan maka persentase ini lebih besar dibandingkan dengan hasil pelatihan yang sebesar 0.0728086 untuk input yang digeser dan persentase *mean square error* pengujian sebesar 0.12. Persentase ini jika dibandingkan dengan nilai *mean square error* hasil pelatihan maka persentase ini lebih besar dibandingkan dengan hasil pelatihan yang sebesar 0.0741973 untuk input yang tidak digeser.

$$MSE \text{ selisih} = \frac{\sum(\text{selisih})^2}{\text{banyaknya data}} \times \frac{100}{\text{nilai data max}} \% \quad (4.5)$$

Persentase ini belum mencapai nilai minimum karena struktur yang digunakan saat pengujian adalah hasil pelatihan yang berhenti disebabkan karena MSE minimum pelatihan belum mencapai target MSE. Hal ini juga disebabkan keterbatasan komputer yang digunakan saat pelatihan dan simulasi, sehingga dengan data masukan yang banyak, perlu dicoba menggunakan komputer dengan tingkat proses komputasi yang lebih tinggi atau dengan menggunakan metode pembelajaran yang lebih cepat agar dapat mencapai MSE target yang sekecil mungkin.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah pelatihan dan pengujian dilakukan, maka dapat disimpulkan bahwa :

1. JST sebagai pengendali *invers* pada suhu shower kurang bekerja dengan baik.
2. Terjadi overshoot pada input yang digeser yang tidak dapat dihindari, hal ini dapat terjadi karena hasil pelatihan jaringan terbaik yang didapatkan belum maksimal, sehingga pada saat pengujian nilai error masih besar bahkan terjadi *overshoot*.
3. Banyaknya jumlah data yang dijadikan sebagai data pelatihan, berpengaruh terhadap lamanya waktu iterasi untuk mencapai target *error* minimum, jumlah iterasi akan semakin banyak dan nilai *mean square error* yang dihasilkan.
4. Struktur terbaik jaringan syaraf tiruan untuk input yang digeser pada sistem kendali suhu shower terdiri dari 13 sel neuron lapisan *input*. Lapisan tersembunyi terdiri dari 2 lapisan, dimana lapisan tersembunyi pertama memiliki 10 sel neuron, lapisan tersembunyi kedua terdiri dari 2 sel neuron. (*Mean Square Error*) MSE yang dihasilkan adalah 0.0728086 dengan fungsi aktivasi menggunakan *tansig* (fungsi *sigmoid* bipolar).
5. Struktur terbaik jaringan syaraf tiruan untuk input yang tidak digeser pada sistem kendali suhu shower terdiri dari 10 sel neuron lapisan *input*.

Lapisan tersembunyi terdiri dari 2 lapisan, dimana lapisan tersembunyi pertama memiliki 8 sel neuron, lapisan tersembunyi kedua terdiri dari 2 sel neuron. (*Mean Square Error*) MSE yang dihasilkan adalah 0.0741973 dengan fungsi aktivasi menggunakan fungsi *tansig* (fungsi *sigmoid bipolar*).

6. Hasil perbandingan nilai MSE antara input yang digeser dan yang tidak digeser adalah 0.44 dan 0.12. Dari nilai tersebut maka untuk sistem kendali shower lebih baik menggunakan input yang tidak digeser.
7. Jumlah lapisan tersembunyi sangat mempengaruhi kecepatan pelatihan, karena semakin banyak proses komputasinya, dan akan menyebabkan lama interaksi untuk mencapai target MSE semakin besar.
8. Persentase MSE untuk selisih suhu air pada tangki adalah sebesar 0.44 untuk input yang digeser dan untuk input yang tidak digeser persentase MSE untuk selisih suhu air sebesar 0.12.

5.2 Saran

1. Pelatihann jaringan dilakukan dengan metode yang berbeda, agar menghasilkan nilai *Mean Square Error* yang lebih kecil lagi.
2. Menggunakan basis ilmu pengontrol lain untuk dapat membandingkan hasil yang lebih baik.

DAFTAR PUSTAKA

- Andriyanti Dwi K., *Pengolahan Informasi Secara Cerdas*, 1993, Artikel.
- Brooks/Cole, *Simulations of Machines*.
- Fausett, Laurance, *Fundamental Of Neural Network*, 1994, Prentice Hall International Edition.
- Harvey, Robert L., *Neural Network Principles*, 1993, Prentice Hall International Edition.
- Kung, S.Y., *Digital Neural Networks*, 1993, Prentice Hall International Edition.
- Kuo, Benjamin C., *Automatic Control System*, edisi bahasa indonesia jilid 1, 1995, Prentice Hall International Edition.
- Kusumadewi, Sri, *Artificial Intelligence*, 2003, Graha Ilmu.
- Kusumadewi, Sri, *Membangun Jaringan Syaraf Tiruan dengan Matlab dan Exellink*, 2003, Graha Ilmu.
- Ogata, Katsuhiko, *Teknik Kontrol Automatik*, jilid 1, 1997, Erlangga.
- Suwandi, Adang, dkk, *Sistem Penggerak Cerdas Berbasis Jaringan Syaraf Tiruan*, Tabloid Elektron, No. 49, Th XVIII.
- _____, *Neural Network Toolbox User's Guide*, 2002, The MathWorks international.