

**Pengendali Suhu Water Heater Dengan Jaringan Syaraf  
Tiruan Backpropagation**

**TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat

Untuk Memperoleh Gelar Sarjana Teknik Elektro



Oleh:

Nama : Hanung Pratama N

No. Mahasiswa : 04524012

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

**2011**

**LEMBAR PENGESAHAN PEMBIMBING**

**PENGENDALI SUHU WATER HEATER DENGAN  
JARINGAN SYARAF TIRUAN  
BACKPROPAGATION**

**TUGAS AKHIR**



Oleh:

Nama : Hanung Pratama N

No. Mahasiswa : 04524012

Yogyakarta, September 2011

Menyetujui,

Pembimbing I

Pembimbing II

Dwi Ana Ratna Wati, ST, M.Eng.

Wahyudi Budi Pramono, ST, M.Eng.

LEMBAR PENGESAHAN PEMBIMBING

PENGENDALI SUHU WATER HEATER DENGAN  
JARINGAN SYARAF TIRUAN  
BACKPROPAGATION

TUGAS AKHIR

Oleh:

Nama : Hanung Pratama N

No. Mahasiswa : 04524012

Yogyakarta, September 2011

Menyetujui,

Pembimbing I

Dwi Ana Ratna Wati, ST, M.Eng.

Pembimbing II

Wahyudi Budi Pramono, ST, M.Eng.

**LEMBAR PENGESAHAN PENGUJI**  
**PENGENDALI SUHU WATER HEATER DENGAN**  
**JARINGAN SYARAF TIRUAN**  
**BACKPROPAGATION**

**TUGAS AKHIR**

Oleh:

Nama : Hanung Pratama N

No. Mahasiswa : 04524012

**Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat**

**untuk Memperoleh Gelar Sarjana Teknik Elektro**

**Fakultas Teknologi Industri Universitas Islam Indonesia**

Yogyakarta, September 2011

Tim Penguji,

Dwi Ana Ratna Wati, ST., M. Eng

Ketua

Dr. Eng. Hendra Setiawan, ST., MT

Anggota I

Medilla K, ST., M.Eng

Anggota II

Mengetahui,

Ketua Jurusan Teknik Elektro

Universitas Islam Indonesia



Tito Yuwono, ST., M.Sc

**LEMBAR PERNYATAAN KEASLIAN**

Saya yang bertanda tangan dibawah ini.

Nama : Hanung Pratama N

No. Mahasiswa : 04524012

Menyatakan bahwa Tugas Akhir ini adalah hasil pekerjaan saya sendiri, dan sepanjang sepengetahuan saya, tidak berisi materi yang ditulis oleh orang lain sebagai persyaratan penyelesaian studi di Universitas Islam Indonesia atau perguruan tinggi lain, kecuali bagian-bagian tertentu yang saya ambil sebagai acuan dengan mengikuti tata cara dan etika penulisan karya ilmiah yang lazim. Jika ternyata terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab saya.

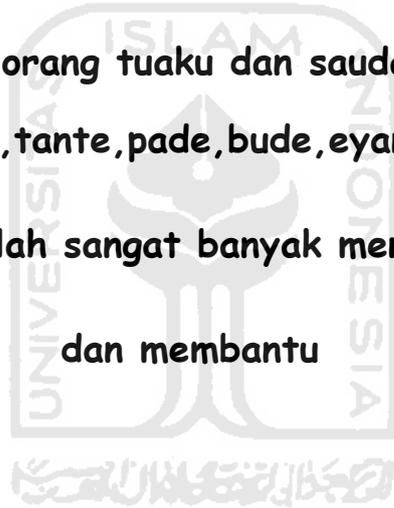
Yogyakarta, September 2011

Hanung Pratama N



## HALAMAN PERSEMBAHAN

**Karya Ini Aku Persembahkan**  
**Allah SWT Atas segala nikmat dan kemudahan**  
**yang Engkau berikan,**  
**Kepada orang tuaku dan saudaraku :**  
**(om, tante, pade, bude, eyang)**  
**yang telah sangat banyak mensupport**  
**dan membantu**



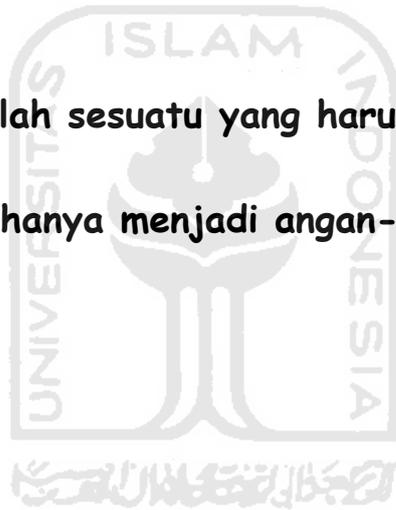
**HALAMAN MOTTO**

**Kemudahan datang bersama kesulitan**

**Janganlah berharap tanpa doa dan usaha**

**Keinginan adalah sesuatu yang harus diusahakan**

**Bukan hanya menjadi angan-angan**



## 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 “Pengendali Water Heater Dengan Jaringan Syaraf Tiruan”. 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 syarat menjadi sarjana 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.

Ucapan terima kasih yang sebesar-besarnya kepada berbagai pihak yang telah memberikan bantuan, bimbingan, dukungan, kerjasama, fasilitas dan kemudahan lainnya kepada beberapa pihak antara lain:

1. Bapak Tito Yuwono, ST, M.Sc. selaku Ketua Jurusan Teknik Elektro Universitas Islam Indonesia, sekaligus Dosen Pembimbing II Tugas Akhir.
2. Ibu Dwi Ana Ratna Wati, ST, M.Eng. selaku Pembimbing I Tugas Akhir yang telah meluangkan waktunya untuk memberikan masukan dan bantuan sampai terselesaikannya Tugas Akhir ini.
3. Wahyudi Budi Pramono, ST, M.Eng. selaku Pembimbing II Tugas Akhir yang telah meluangkan waktunya sampai terselesaikannya Tugas Akhir ini.
4. Seluruh Dosen Jurusan Teknik Elektro, terima kasih atas bimbingan selama saya kuliah di Jurusan Teknik Elektro.
5. Bapak dan Ibuku yang selalu memberikan semangat, mendukung dan memberikan doanya.
6. Om Tofa dan bulik yang memberikan bantuan besar memotifasi dan bantuannya.
7. Keluarga besarku, Pakde, Bude, Om, bulik, yang telah memberi motifasi dan bantuannya.
8. Subhan yang telah banyak membantu selama ini.
9. Mas Heri yang membantu menyediakan alat dan tempat di lab kendali.
10. Teman-teman seperjuangan yang telah memberikan dukungan dan bantuannya.
11. Dan banyak pihak lain yang tidak dapat kami sebutkan seluruhnya yang telah membantu dalam penyelesaian Tugas Akhir ini.

Penulis menyadari sepenuhnya bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Hal ini disebabkan karena keterbatasan kemampuan dan kurangnya pengetahuan yang dimiliki penulis. Oleh karena itu penulis mengharapkan kritik dan saran yang bersifat konstruktif dan solutif dari semua pembaca untuk kebaikan dan kesempurnaan tugas akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi siapa saja yang membutuhkan.

*Wassalaamu'alaikum Wr.Wb.*

Yogyakarta, September 2011



Penulis

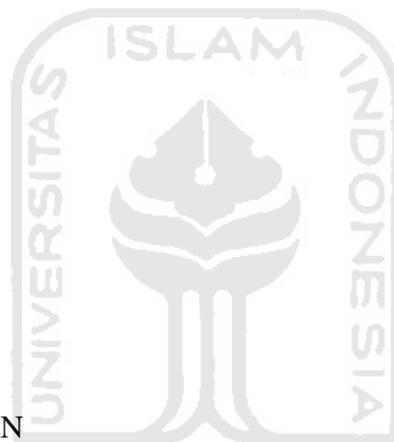
## ABSTRAKSI

Water heater banyak digunakan dalam kehidupan sehari-hari, baik di rumah tangga, industri maupun lingkungan pendidikan yang membutuhkan pengaturan pemanas secara otomatis. Jaringan Syaraf Tiruan merupakan salah satu kendali water heater yang dapat diaplikasikan menggunakan *neural network toolbox* pada *software* Matlab 7.4 dengan menggunakan NI DAQ USB 6009 untuk mengakuisisi data dan identifikasi data dari pemanas air. Data yang diperoleh dari pembacaan tanggapan sensor suhu LM35 menggunakan program matlab dengan *simulink*, yang diambil dari perangkat keras NI DAQ USB 6009 dengan memberikan nilai tegangan kontrol antara 0 – 5 volt secara *random*. Data yang diperoleh disimpan pada program matlab. Dari program matlab diperoleh input berupa nilai perubahan suhu ( $e$ ) dan output tegangan kontrol. Data yang diperoleh dilatih menggunakan metode *backpropagation*. Struktur jaringan terbaik yang diperoleh dari pelatihan, terdiri dari 10 sel neuron lapisan *input*, 5 sel neuron lapisan tersembunyi dan 1 sel neuron lapisan *output*. Fungsi aktivasi pada setiap lapisan menggunakan fungsi sigmoid bipolar-sigmoid bipolar-identitas, diperoleh *Mean Square Error* (MSE)  $9.68815 \cdot 10^{-8}$ . Pengujian sistem dengan memberikan suhu referensi sebesar 50°C, 70°C dan 90°C, diperoleh *rise time* masing-masing 152 detik untuk suhu referensi 50°C, 291 detik untuk suhu referensi 70°C dan 416 untuk suhu referensi 90°C. JST yang dirancang mempunyai respon yang cukup cepat dan mampu beroperasi pada rentang suhu 40 °C sampai 90 °C. Pemberian gangguan berupa air es pada suhu 70 °C membuat suhu *plant* turun sampai 63.2855°C dan dibutuhkan waktu 149 detik untuk kembali ke suhu referensi.

Kata kunci: Jaringan syaraf tiruan, kendali suhu, *water heater*

## DAFTAR ISI

Halaman Judul	i
Lembar Pengesahan Pembimbing	ii
Lembar Pernyataan Keaslian	iii
Lembar Pengesahan Penguji	iv
Halaman Persembahan	v
Halaman Motto	vi
Kata Pengantar	vii
Abstraksi	x
Daftar Isi	xi
Daftar Gambar	xviii
Daftar Tabel	xix
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Metodologi Penelitian	3
1.6 Sistematika Penulisan	3
<b>BAB II TINJAUAN PUSTAKA</b>	
2.1 Tinjauan Pustaka	5



2.2	Sistem Jaringan Syaraf Tiruan	7
2.2.1	Jaringan Syaraf Manusia Sebagai Dasar JST	8
2.2.2	Cara Kerja Komponen JST	9
2.2.3	Konsep Belajar JST	11
2.2.4	Arsitektur JST	13
2.2.5	Fungsi Aktifasi	15
2.2.6	Metode Backpropagation	16
2.2.6.1	Arsitektur <i>Backpropagation</i>	17
2.2.6.2	Algoritma <i>Backpropagation</i>	18
2.2.7	Prosedur Pelatihan	20
2.2.8	Prosedur Pengujian	24
2.2.9	Pemrograman Jaringan dengan <i>Procedure</i> dan <i>Function</i>	26
2.3	NI USB-6009	28
2.3.1	Konstruksi NI USB-6009	30
2.3.2	<i>Input</i> Analog	30
2.3.3	<i>Output</i> Analog	33
2.4	Sensor Suhu LM35	33
2.5	IC TCA 785	34
2.6	Kontrol Fasa Tegangan AC 1 $\Phi$	38
2.7	TRIAC	40
2.8	<i>Optocoupler</i>	42
2.9	Penguat Operasional (Op-Amp)	43

<b>BAB III PERANCANGAN SISTEM</b>	
3.1	Rangkaian Penguat Sensor LM 35 49
3.2	Penguat Tegangan Kontrol 51
3.3	Rangkaian Driver AC 53
3.4	Analog Input Pada Matlab 57
3.5	Analog Output Pada Matlab 59
3.6	Perancangan Jaringan Syaraf Tiruan 60
3.6.1	Diagram Alir JST <i>Backpropagation</i> 61
3.6.2	Diagram Alir Prosedur Pengujian 64
3.2	Perancangan Simulink 65
<b>BAB IV PENGUJIAN, ANALISIS DAN PEMBAHASAN</b>	
4.1	Pelatihan JST <i>Backpropagation</i> 69
4.1.1	Pelatihan Menggunakan 1 Lapisan Tersembunyi 69
4.1.2	Pelatihan Menggunakan Lebih 1 Lapisan Tersembunyi 73
4.1.3	Pelatihan Menggunakan Lebih 2 Lapisan Tersembunyi 76
4.2	Pengujian Sistem Pengendali JST <i>Backpropagation</i> 82
4.3.1	Pengujian Dengan Variasi <i>Set Point</i> 82
4.3.2	Pengujian Sistem Pada Suhu Referensi Naik 86
4.3.3	Pengujian Sistem Dengan Gangguan 88
4.3.4	Perbandingan Kendali Jaringan Syaraf dengan PID 89
<b>BAB V PENUTUP</b>	
5.1	Kesimpulan 92
5.2	Saran 93

DAFTAR PUSTAKA

LAMPIRAN



**DAFTAR GAMBAR**

Gambar 2.1	Struktur neuron jaringan syaraf tiruan	10
Gambar 2.2	Jaringan syaraf dengan tiga lapisan	11
Gambar 2.3	Jaringan syaraf dengan lapisan tunggal	13
Gambar 2.4	Jaringan syaraf dengan banyak lapisan	14
Gambar 2.5	Jaringan syaraf dengan lapisan kompetitif	14
Gambar 2.6	Fungsi aktivasi <i>linear</i> /identitas	15
Gambar 2.7	Fungsi aktivasi <i>sigmoid</i> biner	16
Gambar 2.8	Fungsi aktivasi <i>sigmoid</i> bipolar	16
Gambar 2.9	Arsitektur jaringan sederhana	18
Gambar 2.10	Arsitektur algoritma <i>backpropagation</i>	20
Gambar 2.11	NI USB-6009	28
Gambar 2.12	Blok Diagram NI USB-6009	30
Gambar 2.13	Skema mode differential	32
Gambar 2.14	Skema mode RSE	32
Gambar 2.15	<i>Connecting load</i> AO	33
Gambar 2.16	<i>Aplikasi dasar LM35 sebagai sensor suhu</i>	34
Gambar 2.17	Bentuk fisik IC TCA 785	34
Gambar 2.18	Konfigurasi pin IC TCA 785	35
Gambar 2.19	Diagram pulsa IC TCA 785	37
Gambar 2.20	Rangkaian pengontrol gelombang penuh satu fasa	38
Gambar 2.21	Bentuk gelombang pengontrol gelombang penuh satu fasa	39
Gambar 2.22	(a). Rangkaian ekivalen TRIAC, (b). Simbol TRIAC	40

## (b). Simbol TRIAC

Gambar 2.23	Karakteristik TRIAC	41
Gambar 2.24	Skema <i>optocoupler</i>	42
Gambar 2.25	Diagram skematis simbol Op-Amp	43
Gambar 2.26	Penguat <i>non-inverting</i>	44
Gambar 2.27	Hubungan sinyal <i>input-output</i> penguat <i>non-inverting</i>	45
Gambar 3.1	Blok diagram sistem kendali suhu <i>water heater</i>	46
Gambar 3.2	Blok diagram sistem pengendali jaringan syaraf tiruan <i>backpropagation</i>	47
Gambar 3.3	Gambar invers pelatihan	48
Gambar 3.4	Rangkaian penguat sensor LM35	49
Gambar 3.5	Grafik hubungan tegangan keluaran LM35 dengan Op-Amp	50
Gambar 3.6	Rangkaian penguat tegangan control	51
Gambar 3.7	Grafik hubungan tegangan kanal AO NI USB-6009 dengan Op-Amp	52
Gambar 3.8	Rangkaian pengontrol sudut fasa	53
Gambar 3.9	Grafik hubungan tegangan kontrol dengan tegangan beban	56
Gambar 3.10	Blok untuk <i>analog input</i>	57
Gambar 3.11	Blok manipulasi suhu pada Matlab	59
Gambar 3.12	Blok untuk <i>analog output</i>	59
Gambar 3.13	Arsitektur jaringan syaraf tiruan	60
Gambar 3.14	Proses pembangunan jaringan syaraf tiruan	60

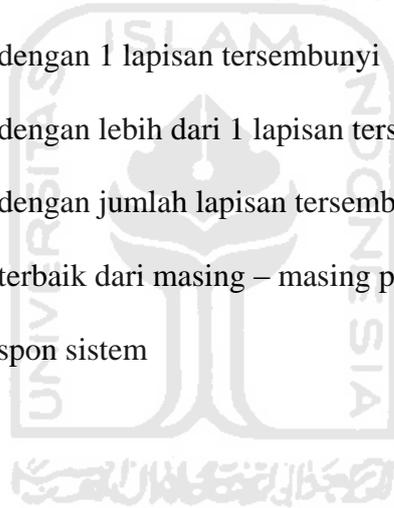
Gambar 3.15	Diagram alir/ <i>flowchart</i> prosedur pelatihan	63
Gambar 3.16	Diagram alir/ <i>flowchart</i> prosedur pengujian	64
Gambar 3.17	Blok rangkaian simulink	65
Gambar 4.1	Hasil pelatihan dengan 1 lapisan tersembunyi fungsi aktivasi <i>sigmoid bipolar-identitas</i>	71
Gambar 4.2	Hasil pelatihan dengan 1 lapisan tersembunyi fungsi aktivasi <i>sigmoid biner-identitas</i>	72
Gambar 4.3	Hasil pelatihan dengan 1 lapisan tersembunyi fungsi aktivasi <i>sigmoid bipolar-sigmoid bipolar-identitas</i>	74
Gambar 4.4	Hasil pelatihan dengan 1 lapisan tersembunyi fungsi aktivasi <i>sigmoid biner-sigmoid bipolar-identitas</i>	75
Gambar 4.5	Hasil pelatihan dengan 1 lapisan tersembunyi fungsi aktivasi <i>sigmoid biner-identitas-identitas</i>	75
Gambar 4.6	Hasil pelatihan dengan 1 lapisan tersembunyi fungsi aktivasi <i>sigmoid biner-sigmoid biner-identitas</i>	76
Gambar 4.7	Hasil pelatihan dengan 1 lapisan tersembunyi fungsi aktivasi <i>sigmoid biner-sigmoid bipolar-sigmoid biner-sigmoid bipolar- identitas</i>	77
Gambar 4.8	Respon sistem pada <i>set point</i> 50 °C	83
Gambar 4.9	Respon sistem pada <i>set point</i> 70 °C	83

Gambar 4.10 Respon sistem pada <i>set point</i> 90 °C	84
Gambar 4.11 Grafik respon sistem pada suhu referensi naik	87
Gambar 4.12 Grafik respon sistem dengan gangguan	88
Gambar 4.13 Grafik respon sistem kendali PID	90
Gambar 4.14 Grafik respon sistem kendali Jaringan Syaraf	90



**DAFTAR TABEL**

Tabel 2.1 Fungsi setiap pin terminal analog	31
Tabel 2.2 Deskripsi fungsi pin-pin IC TCA 785	35
Tabel 3.1 Penguatan sensor LM35	50
Tabel 3.2 Penguat tegangan kontrol	52
Tabel 3.3 Pengujian blok driver AC	55
Tabel 3.4 Data penunjukkan termometer dan tegangan pada Matlab	58
Tabel 4.1 Hasil pelatihan dengan 1 lapisan tersembunyi	70
Tabel 4.2 Hasil pelatihan dengan lebih dari 1 lapisan tersembunyi	73
Tabel 4.3 Hasil pelatihan dengan jumlah lapisan tersembunyi lebih dari 2	76
Tabel 4.4 Hasil pelatihan terbaik dari masing – masing pengelompokan	78
Tabel 4.5 Karakteristik respon sistem	84



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Teknik pengontrolan otomatis saat ini memberikan peranan yang sangat besar dalam kehidupan sehari-hari. Khususnya dibidang ilmu pengetahuan dan industri. Beberapa diantaranya adalah pengaturan suhu ruangan, pengaturan kelembaban, pengaturan level air, pengaturan kecepatan motor, pengaturan tekanan udara dan masih banyak lainnya.

Di bidang pengaturan suhu dengan *water heater* dibutuhkan untuk kepentingan rumah tangga dan industri. Di dalam kebutuhan industri digunakan untuk memperlancar dan mempercepat jalanya proses produksi. Sistem pemanas banyak digunakan dalam industri pengolahan minyak, pengolahan pupuk, pengolahan makanan, pengolahan tambang, dan sebagainya.

Dalam prosesnya *water heater* membutuhkan sistem pengendalian yang dapat mengatur suhu. Pengendalian bisa dilakukan manual dan otomatis. Untuk mempermudah dan mempercepat proses digunakan sistem pengendali yang berjalan secara otomatis. Sebuah sistem kendali tersebut dibutuhkan untuk menjaga kondisi supaya dalam keadaan suhu yang diinginkan.

Model-model sistem dapat dibuat sebaik mungkin sesuai dengan kondisi yang diinginkan. Dengan identifikasi sistem yang baik akan memperkecil kesalahan pada suatu sistem kendali.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang dipaparkan sebelumnya, dapat diambil rumusan masalah sebagai berikut:

1. Bagaimana merancang dan merealisasikan pengendali *water heater* dengan menggunakan algoritma Jaringan Syaraf Tiruan Backpropagation.
2. Bagaimana merancang pengendali Jaringan Syaraf Tiruan sehingga dihasilkan sistem dengan respon yang cepat, sekaligus mempunyai kestabilan yang baik.

## 1.3 Batasan Masalah

Adapun batasan-batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Hanya dibahas pengendalian suhu pada air bervolume 1.000 mL menggunakan metode JST dengan batas suhu 40 °C sampai 90 °C (suhu naik).
2. JST yang digunakan adalah model *backpropagation*.
3. Respon sistem yang diamati adalah respon sistem *closed loop*
4. *Software* yang digunakan adalah Matlab.
5. *Hardware* Akuisisi data menggunakan NI DAQ USB-6009.

## 1.4 Tujuan Penelitian

Penelitian dan perancangan sistem memiliki beberapa tujuan, yaitu :

1. Merancang sebuah sistem pengendali suhu naik air dengan algoritma jaringan syaraf tiruan metode *backpropagation*.
2. Membuat suatu pelatihan untuk menghilangkan atau memperkecil *error* pada sistem.

### **1.5 Metodologi Penelitian**

Penelitian dalam tugas akhir ini akan dilakukan dengan tahapan berikut :

1. Studi literatur, yaitu mengumpulkan data dengan cara mempelajari literatur/buku, artikel yang berkaitan dengan topik tugas akhir. Termasuk artikel dan data dari internet.
2. Konsultasi, yaitu memperoleh data dari pihak yang mengetahui tentang topik tugas akhir.
3. Perancangan dan pengujian sistem, yaitu merancang sistem berdasarkan teori yang diperoleh saat studi literatur dan menguji sistem agar dapat dianalisis dan diambil kesimpulan.

### **1.6 Sistematika Penulisan**

Sistematika penulisan dan pembahasan laporan tugas akhir ini dapat dijelaskan sebagai berikut:

#### **BAB I PENDAHULUAN**

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

## **BAB II TINJAUAN PUSTAKA**

Memuat dasar-dasar teori yang berhubungan dengan penelitian dan juga dasar teori yang berhubungan dengan jaringan syaraf tiruan, NI DAQ, sensor, IC TCA 785, kontrol fasa tegangan AC 1 fasa, triac, optocoupler, dan penguat op-amp yang akan digunakan dalam penelitian.

## **BAB III PERANCANGAN SISTEM**

Menjelaskan tentang pemrograman jaringan syaraf tiruan yang digunakan untuk *system control water heater*, pembagian fungsi kerja dalam diagram blok dan diagram alir, perancangan dan pengujian sistem yang dibuat dengan menggunakan *simulink* pada program matlab.

## **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 KESIMPULAN DAN SARAN**

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

### TINJAUAN PUSTAKA

#### 2.1 Tinjauan Pustaka

Penelitian sebelumnya berjudul “*Perancangan Pengendali PID Digital pada Heater*” (Andika, Roni: 2010). Pada penelitian ini telah dirancang dan diimplementasikan sistem kendali tipe PID untuk pengendalian suhu pada pemanas (heater) air. Sistem kendali tipe PID dibuat dengan menggunakan labview 8.6 dengan nilai setpoint sebagai masukan. Analog input (AI) NIDAQ 6009 terhubung dengan sensor suhu LM35 dan analog output (AO) terhubung dengan driver AC. Dari perancangan sistem kendali PID yang telah dibuat nilai  $K_p=1$ ,  $K_i=7$ , dan  $K_d=5$  dapat menghasilkan kestabilan yang baik. Pengujian sistem keseluruhan dilakukan dengan menggunakan nilai setpoint  $70^0$  C dari suhu awal air sebesar  $29^0$  C. dari hasil pengujian, diperoleh waktu untuk mencapai kestabilan rata-rata sebesar 371 *second*. Secara umum hasil yang diperoleh menunjukkan bahwa *control* PID mampu mengendalikan suhu air dengan kestabilan yang baik dan tanggapan yang cepat terhadap perubahan suhu yang dikendalikan.

Penelitian juga dilakukan (Priawan, Rizki: 2010), dimana penelitian ini mengakuisisi data dan identifikasi data dari pemanas air. Data yang diperoleh dari nilai tegangan sensor suhu LM35 yang diambil oleh perangkat keras akuisisi data NI

DAQ 6009 dan hasilnya disimpan dalam PC menggunakan labview. Setelah data diperoleh diidentifikasi menggunakan matlab. Dari matlab diperoleh input berupa nilai tegangan yang dikeluarkan NI DAQ 6009 dan data output dari op-amp sensor LM35, kemudian data diidentifikasi dengan dibagi dua yaitu data estimasi dan data validasi. Kedua data dibandingkan untuk memperoleh persamaan matematis yaitu fungsi alih dan proses model. Pada tahap ini parameter dapat diubah sampai mendapatkan hasil terbaik. Hasil yang diperoleh pada proses ini yaitu fungsi alih orde satu dengan nilai  $K=48,484$ ,  $TPL=167,29$  dan  $TD=30$ . Grafik *step respon open loop* yang ditampilkan dari fungsi alih yang diperoleh dihasilkan nilai *steady state* sebesar  $90^{\circ}\text{C}$  suhu air.

Penelitian tentang '*Aplikasi Logika Fuzzy pada Water Heater Berbasis Labview*' (Isnanto: 2011). Penelitian ini memfokuskan pada *control* dan *monitoring* suhu air dengan menggunakan logika fuzzy. *Control* dan *monitoring* sistem dilakukan secara *real time* pada sebuah PC dengan menggunakan program LabVIEW sebagai *Graphical User Interface* (GUI)-nya, sedangkan proses akuisisi datanya menggunakan NI USB-6009. Perancangan *Fuzzy Logic Controller* (FLC) menggunakan LabVIEW *PID and Fuzzy Logic Toolkit*. FLC yang dirancang mempunyai 2 buah *crisp input* yaitu *error* ( $e$ ) dan *change of error* ( $\Delta e$ ), serta 1 buah *crisp output* yaitu *change of voltage* ( $\Delta u$ ), dengan metode inferensi Max-Min dan metode defuzzifikasi *Center of Area* (COA). Jumlah label yang digunakan 5 buah dengan tipe segitiga dan *trapezoid*, dengan 25 buah aturan.

## 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 diimplementasikan 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.

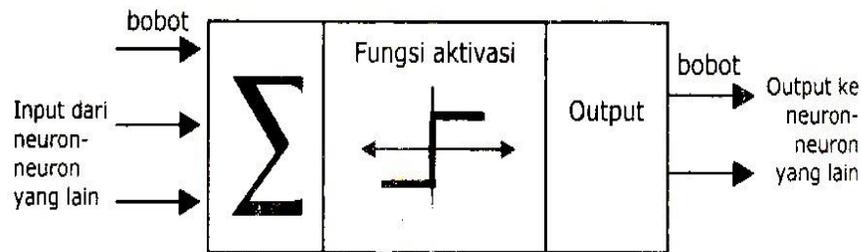
Dalam jaringan syaraf manusia, akson dan dendrit bercabang-cabang sedemikian banyaknya. Hal ini menunjukkan bahwa adanya sistem paralel dan

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 ambangnya. 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 keluarnya menuju ke neuron-neuron yang lain. Pada JST hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut.

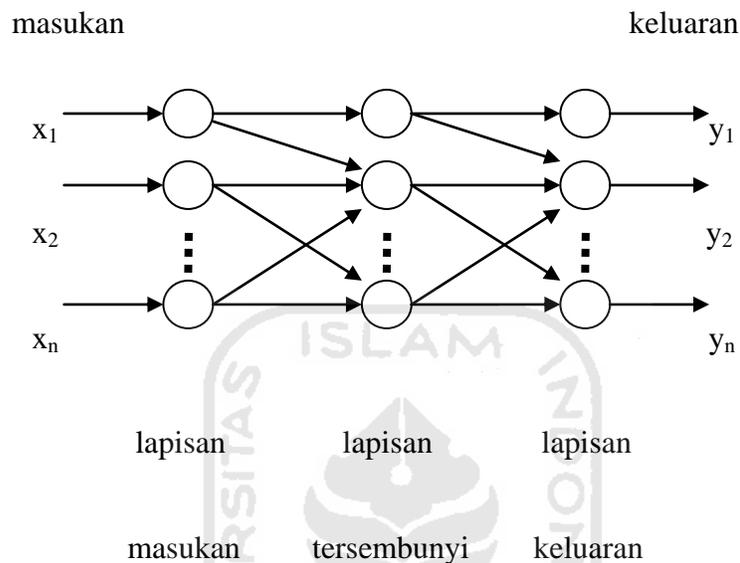


Gambar 2.1 Struktur neuron jaringan syaraf tiruan

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 2.2 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.

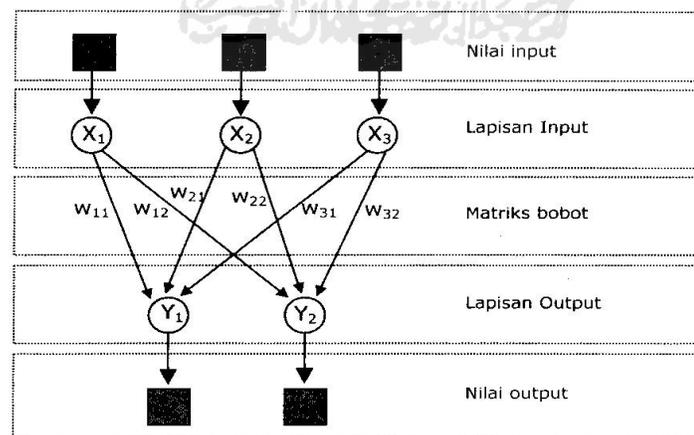
Selama belajar apabila jaringan mengeluarkan jawaban yang salah, maka besar kesalahan dapat dicari, yaitu beda keluaran aktual dan acuannya. Sedangkan dalam belajar tanpa pengawasan, jaringan akan mengubah bobot-bobotnya, sebagai tanggapan terhadap masukan, tanpa memerlukan keluaran acuan.

### 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*).

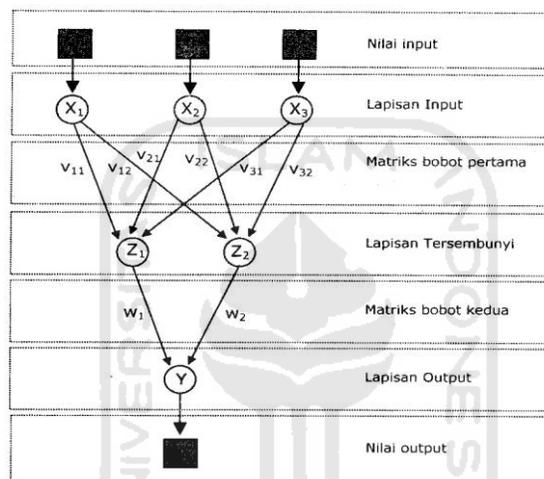
Hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan hanya menerima masukan kemudian secara langsung mengolahnya menjadi keluaran tanpa harus melalui lapisan tersembunyi. Semua unit *input* akan langsung berhubungan dengan unit *output*.



Gambar 2.3 Jaringan syaraf dengan lapisan tunggal

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

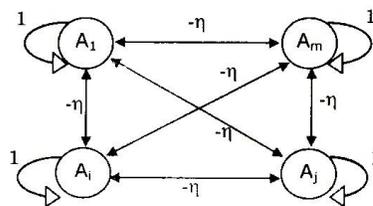
Memiliki satu atau lebih lapisan tersembunyi yang terletak diantara lapisan *input* dan *output*. Dengan pembelajaran yang lebih rumit, jaringan dapat menyelesaikan permasalahan yang lebih sulit dan hasilnya pun lebih sukses dari pada lapisan tunggal.



Gambar 2.4 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 2.5 Jaringan syaraf dengan lapisan kompetitif

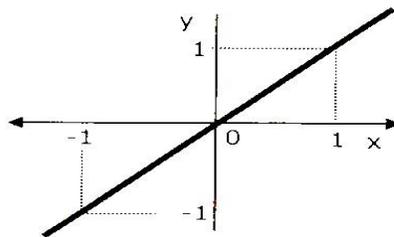
### 2.2.5 Fungsi Aktifasi

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*).

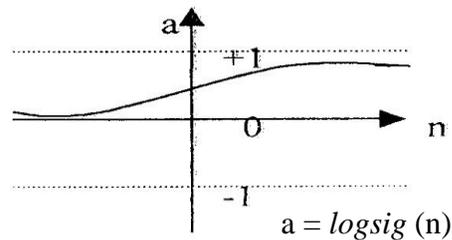
Untuk mendapatkan efisiensi perhitungan, turunannya mudah dihitung. Pada umumnya nilai turunan fungsi aktivasi dapat diperoleh dari nilai rentang tertentu. Ada beberapa fungsi aktivasi yang sering digunakan dalam jaringan syaraf tiruan, tetapi yang biasa digunakan untuk metode *backpropagation* adalah:

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



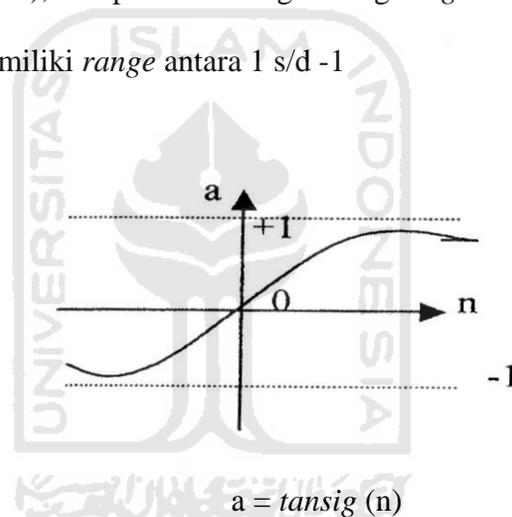
Gambar 2.6 Fungsi aktivasi *linear/identitas*

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



Gambar 2.7 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 2.8 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 :

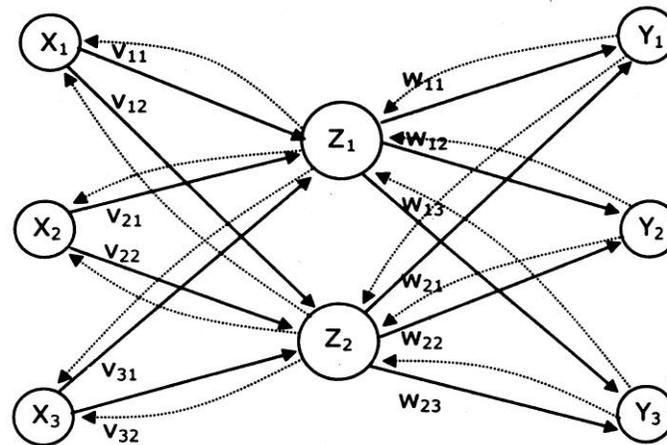
1. Belajar dari kesalahan.
2. Memasukan secara umpan maju (*feed forward*) pola-pola masukan.
3. Menghitung dan propagasi balik kesalahan yang bersangkutan.
4. Mengatur bobot-bobot koneksi.

*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 :



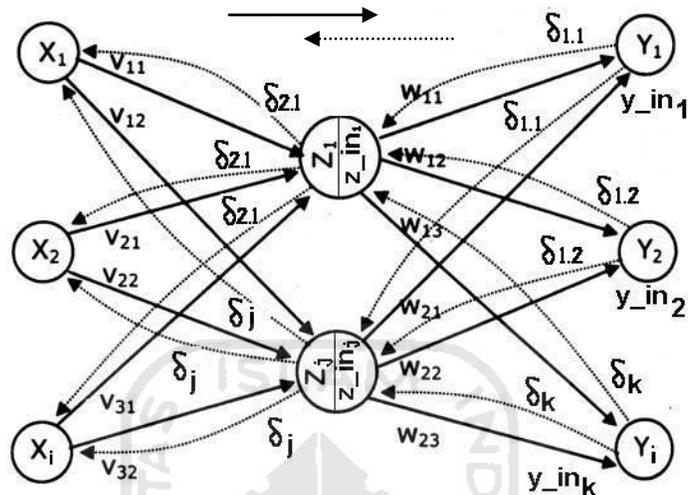
Gambar 2.9 Arsitektur jaringan sederhana

Pada Gambar 2.9 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 2.10 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  $\delta_k$  ( $k = 1, \dots, m$ ) dihitung.  $\delta_k$  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,  $\delta_j$  ( $j = 1, \dots, p$ ) dihitung untuk tiap unit tersembunyi ( $z_j$ ). Untuk ( $\delta_j$ ) 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  $\delta$  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  $\delta_k$  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  $\delta_j$  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 Prosedur Pelatihan

*Error* pada jaringan akan di propagasi-balikan selama pelatihan. *Error* 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 *error* 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 ( $hl$ ), 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),

Langkah 4 : Setiap unit tersembunyi ( $z_j ; j = 1, \dots, p$ ) menjumlahkan bobot sinyal masukan,

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} , \quad (2.1)$$

Digunakan fungsi aktivasi untuk menghasilkan keluaran

$$z_j = f(z_{in_j}), \quad (2.2)$$

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,

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk}, \quad (2.3)$$

Digunakan fungsi aktivasi untuk menghasilkan keluaran

$$y_k = f(y_{in_k}), \quad (2.4)$$

#### **Perambatan balik / *Backward* :**

Langkah 6 : Tiap unit keluaran ( $y_k$  ;  $k = 1, \dots, m$ ) menerima pola target yang saling berhubungan pada masukan pola pelatihan, hitung kesalahan informasinya,

$$\delta_k = (t_k - y_k) f'(y_{in_k}), \quad (2.5)$$

Menghitung koreksi bobotnya (digunakan untuk memperbaharui  $w_{jk}$  nantinya)

$$\Delta w_{jk} = \alpha \delta_k z_j, \quad (2.6)$$

Menghitung koreksi biasnya (digunakan untuk memperbaharui  $v_{0j}$  nantinya)

$$\Delta w_{0k} = \alpha \delta_k , \quad (2.7)$$

Mengirimkan  $\delta_k$  ke unit-unit pada lapisan dibawahnya.

Langkah 7 : Tiap unit lapisan tersembunyi ( $z_j ; j = 1, \dots, hl$ ) menjumlahkan hasil perubahan masukannya (dari unit-unit lapisan diatasnya),

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} , \quad (2.8)$$

Mengalikan dengan turunan fungsi aktivasinya untuk menghitung informasi kesalahannya,

$$\delta_j = \delta_{in_j} f'(z_{in_j}) , \quad (2.9)$$

Menghitung koreksi bobotnya (digunakan untuk memperbaharui  $v_{ij}$  nantinya)

$$\Delta v_{ij} = \alpha \delta_j x_i , \quad (2.10)$$

Menghitung koreksi biasnya (digunakan untuk memperbaharui  $v_{0j}$  nantinya)

$$\Delta v_{0j} = \alpha \delta_j , \quad (2.11)$$

### Perbaiki bobot dan bias :

Langkah 8 : Tiap unit keluaran ( $y_k$  ;  $k = 1, \dots, m$ ) diperbaiki bobot dan biasnya ( $j = 1, \dots, p$ )

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}, \quad (2.12)$$

$$w_{0k}(\text{baru}) = w_{0k}(\text{lama}) + \Delta w_{0k}, \quad (2.13)$$

pada tiap unit lapisan tersembunyi ( $z_j$  ;  $j = 1, \dots, p$ ) diperbaiki bias dan bobotnya ( $i = 1, \dots, n$ )

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}, \quad (2.14)$$

$$v_{0j}(\text{baru}) = v_{0j}(\text{lama}) + \Delta v_{0j}, \quad (2.15)$$

Langkah 9 : Tes kondisi berhenti.

### 2.2.8 Prosedur Pengujian

Bobot dan bias yang telah diperbaiki selama pelatihan, akan digunakan kembali sebagai prosedur pengujian. Prosedur aplikasinya dapat dilihat pada beberapa langkah berikut :

Langkah 0 : Penentuan bobot awal (hasil dari pelatihan),

Langkah 1 : Untuk setiap vektor masukan , lakukan langkah 2 - 4,

Langkah 2 : Untuk setiap masukan, distribusikan masukan  $x_i$  ke setiap unit di atasnya (unit dalam/lapisan tersembunyi),

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

Langkah 4 : Setiap unit tersembunyi ( $z_j ; j = 1, \dots, p$ ) menjumlahkan sinyal masukan terbobot,

$$z\_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} , \quad (2.16)$$

Menggunakan fungsi aktivasi untuk menghasilkan keluaran

$$z_j = f(z\_in_j) , \quad (2.17)$$

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,

$$y\_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} , \quad (2.18)$$

Menggunakan fungsi aktivasi untuk menghasilkan keluaran

$$y_k = f(y\_in_k) , \quad (2.19)$$

### 2.2.9 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.
- $\delta_k$  Koreksi kesalahan untuk pengaturan bobot dan bias pada unit lapisan keluaran ke lapisan tersembunyi.
- $\delta_j$  Koreksi kesalahan untuk pengaturan bobot dan bias pada unit lapisan tersembunyi ke lapisan masukan.
- 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.

- $\Delta$  Perubahan  
 $\Sigma$  Penjumlahan

Pada sistem kendali suhu air, 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)**

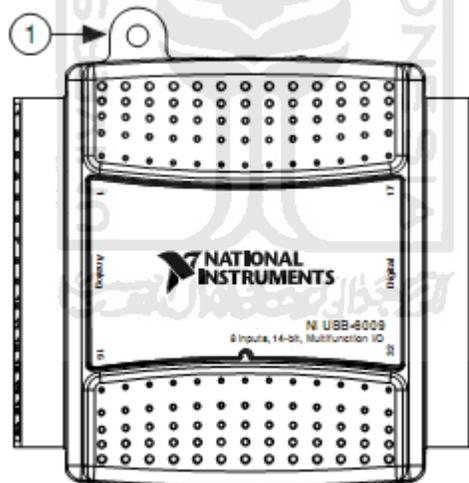
- PR Matriks berukuran  $R \times 2$  yang berisi nilai minimum dan maksimum, dengan  $R$  adalah jumlah variabel *input*.
- Si Jumlah neuron pada lapisan ke- $i$ , dengan  $i = 1, 2, \dots, N1$ .
- Tfi Fungsi aktivasi pada lapisan ke- $i$ , dengan  $i = 1, 2, \dots, N1$ , *default* = *tansig* (*sigmoid bipolar*).
- BTF Fungsi pelatihan jaringan, *default* = *trainlm* (*Levenberg-Marquardt*).
- BLF Fungsi pelatihan untuk bobot, *default* = *learngdm* (*Gradient descent with momentum*).
- PF Fungsi kinerja kesalahan atau *error*, *default* = MSE (*Mean Square Error*).

Pelatihan jaringan (BTF) untuk sistem kendali suhu air menggunakan fungsi *trainlm*. Fungsi ini tidak hanya merespon *gradient* lokal saja, namun juga mempertimbangkan kecenderungan yang baru saja terjadi pada permukaan *error*.

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

### 2.3 NI USB-6009

NI USB-6009, sebagai suatu terobosan teknologi yang fungsi kerjanya hampir sama seperti mikrokontroler, hadir memenuhi kebutuhan (*market need*) dan teknologi baru. Sebagai teknologi baru, NI USB-6009 banyak digunakan oleh para konsumen di bidang industri dan di dalam dunia pendidikan teknik yang dapat mempermudah pekerjaan mereka untuk suatu kepentingan tertentu.



Gambar 2.11 NI USB-6009

Penggunaan NI USB-6009 cukup praktis, karena dimasa sekarang hampir semua komputer memiliki port USB untuk komunikasi data. Sehingga perangkat ini disebut juga sebagai sistem akuisisi data. Karena menggunakan sistem USB maka NI USB-

6009 dapat terkoneksi secara *plug and play* pada komputer. Dengan kemampuan sistem koneksi *plug and play*, modul ini cukup mudah digunakan untuk pengukuran yang cepat dan juga serbaguna untuk aplikasi pengukuran yang lebih kompleks.

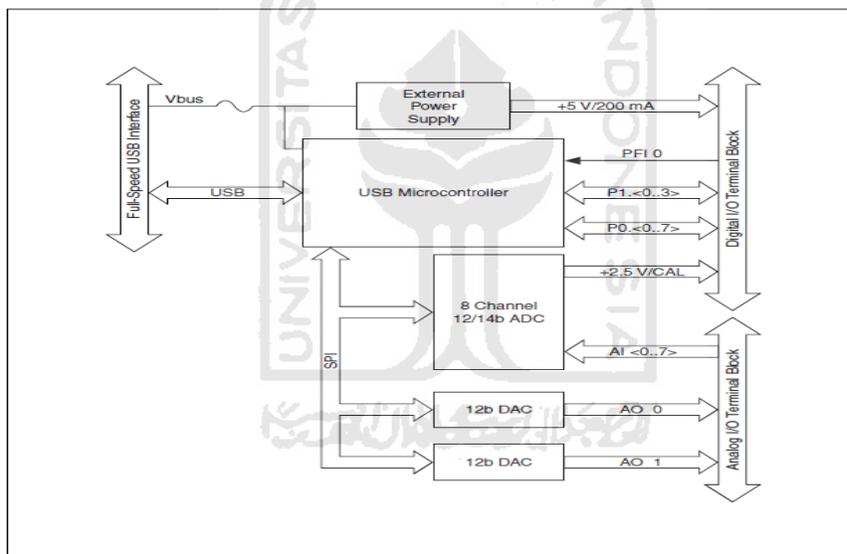
Sistem akuisisi data di sini adalah suatu proses mengumpulkan data dan mengukur sinyal elektrik dari sensor, transduser, *test probe* dan memasukkannya ke komputer untuk kemudian diproses. Data akuisisi mencakup dua tahap yaitu:

1. Membaca sinyal nyata (*real world*) dan menkonversi ke dalam sinyal tegangan.
2. Mengkonversi sinyal tegangan ke dalam bilangan biner sehingga dapat dianalisa oleh komputer.

Perangkat NI USB-6009 dapat digunakan untuk mengumpulkan masukan tegangan ke dalam sebuah komputer, dimana dapat digunakan perangkat lunak untuk memanipulasi tegangan ke dalam besaran angka. Sistem ini menggabungkan semua komponen, perangkat keras dan perangkat lunak untuk menggunakan piranti NI USB-6009. Saat menganalisa atau mengontrol fenomena dunia nyata menggunakan sebuah komputer, diharuskan mengkonversi fenomena tersebut ke dalam data sehingga komputer dapat menerjemahkan atau menghasilkan. Transduser digunakan untuk mengkonversi fenomena ke dalam sinyal elektrik sehingga peralatan NI USB-6009 dapat mengambil dan mengirimkan ke dalam komputer. Disamping itu komputer dapat digunakan untuk menghasilkan sinyal elektrik sehingga piranti NI USB-6009 mengubahnya ke dalam sebuah representasi dunia nyata.

### 2.3.1 Konstruksi NI USB-6009

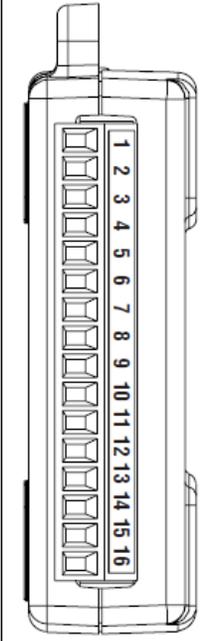
NI USB-6009 menyediakan koneksi 8 *analog input (AI) channel* , 2 *analog output (AO) channel*, 12 *digital input/output (DIO) channel*, dan 32 bit *counter* dengan USB berkecepatan penuh. Blok diagram di bawah memperlihatkan komponen-komponen fungsional yang terdapat pada NI USB-6009. NI USB-6009 memiliki 2 terminal sekrup yang dapat dipisahkan, satu untuk terminal *analog signal* dan yang satu lagi untuk terminal *digital signal*



Gambar 2.12 Blok Diagram NI USB-6009

### 2.3.2 Input Analog

NI USB-6009 dapat menghubungkan sinyal input analog ke PC melalui *I/O connector*. Berikut ini fungsi dari masing-masing pin pada terminal analog.

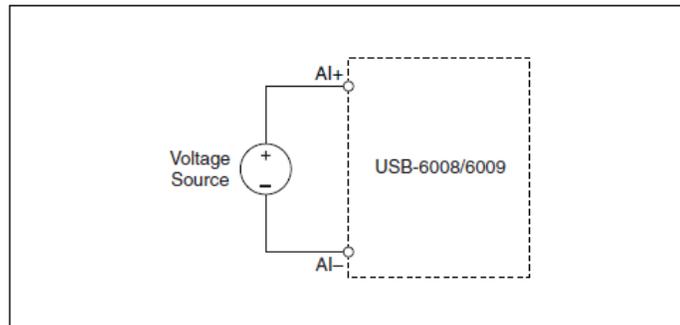
Tabel 2.1 Fungsi setiap pin terminal *analog*


Module	Terminal	Signal, Single-Ended Mode	Signal, Differential Mode
	1	GND	GND
	2	AI 0	AI 0+
	3	AI 4	AI 0-
	4	GND	GND
	5	AI 1	AI 1+
	6	AI 5	AI 1-
	7	GND	GND
	8	AI 2	AI 2+
	9	AI 6	AI 2-
	10	GND	GND
	11	AI 3	AI 3+
	12	AI 7	AI 3-
	13	GND	GND
	14	AO 0	AO 0
	15	AO 1	AO 1
	16	GND	GND

Kanal AI dapat dikonfigurasi dalam dua mode, yaitu:

### 1. Mode Sinyal Tegangan Differential

Ujung sinyal positif dihubungkan ke terminal AI+ dan ujung sinyal negatif ke terminal AI-.

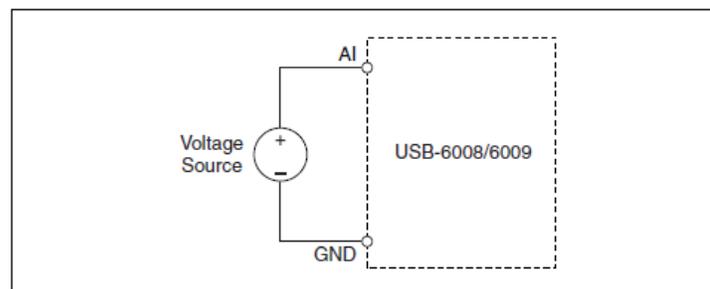


Gambar 2.13 Skema mode differential

Mode input differential dapat mengukur sinyal  $\pm 20$  V pada range  $\pm 20$  V. Bagaimanapun juga, tegangan input pada tiap pin sebesar  $\pm 10$  V yang dihubungkan ke GND. Contoh, jika  $AI1 = +10V$  dan  $AI5 = -10$  V, maka tegangan yang terukur pada device sebesar  $\pm 20$  V.

## 2. Mode *Reference Single-ended* (RSE)

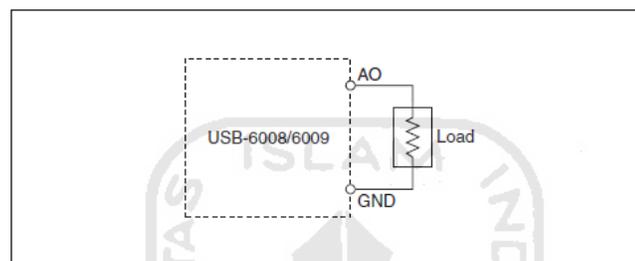
Untuk menggunakan fungsi ini, maka ujung sinyal tegangan positif dihubungkan ke salah satu terminal AI dan ujung yang lain dihubungkan ke GND.



Gambar 2.14 Skema mode RSE

### 2.3.3 Output Analog

NI USB-6009 memiliki dua kanal *Analog Output (AO)* yang terpisah. Kedua AO tersebut dapat membangkitkan output 0 – 5 V. Untuk menghubungkan beban ke NI USB-6009, ujung positif beban dihubungkan ke terminal AO dan *ground* beban dihubungkan ke terminal GND.



Gambar 2.15 *Connecting load AO*

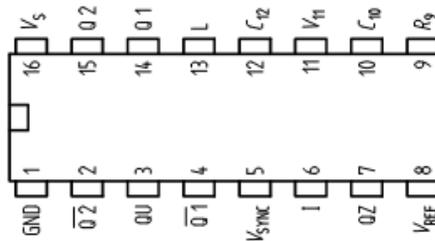
### 2.4 Sensor Suhu LM35

LM35 adalah IC sensor suhu dengan presisi Celcius, dimana tegangan outputnya proporsional linier dengan temperatur dalam derajat Celcius. LM35 memiliki kelebihan dibandingkan sensor suhu berpresisi Kelvin, dimana pemakai tidak perlu mengambil nilai tegangan konstan yang besar untuk mendapatkan skala celcius yang tepat. LM35 memiliki keadaan default yaitu akurasi  $\pm 1/4$  °C pada temperatur ruang dan  $\pm 3/4$  °C pada range maksimum  $-55$  sampai  $+150$  °C.

LM35 memiliki faktor skala linier  $+10.0$  mV/°C, ini berarti untuk tiap kenaikan satu derajat celcius pada suhu sekitar tegangan output akan naik 10 mV. Tegangan kerja dari LM35 adalah 4 sampai 30 Volt dengan kuat arus sebesar 60  $\mu$ A.



terkontrol (*dc chopper*). IC ini memiliki kaki (pin) sejumlah 16 buah, seperti yang terlihat pada gambar 2.18



Gambar 2.18 Konfigurasi pin IC TCA 785

Berikut ini adalah fungsi dari tiap-tiap pin IC TCA 785:

Tabel 2.2 Deskripsi fungsi pin-pin IC TCA 785

Pin	Symbol	Function
1	GND	Ground
2	Q2	Output 2 inverted
3	Q U	Output U
4	Q2	Output 1 inverted
5	$V_{\text{SYNC}}$	Synchronous voltage
6	I	Inhibit
7	Q Z	Output Z
8	$V_{\text{REF}}$	Stabilized voltage
9	$R_9$	Ramp resistance
10	$C_{10}$	Ramp capacitance
11	$V_{11}$	Control voltage
12	$C_{12}$	Pulse extension
13	L	Long pulse
14	Q 1	Output 1
15	Q 2	Output 2
16	$V_s$	Supply voltage

IC TCA 785 memerlukan sumber tegangan antara 8 Volt hingga 18 Volt, frekuensi kerja 10 Hz hingga 500 Hz, serta temperatur kerja antara – 250 hingga 85 °C. Prinsip kerja IC TCA 785 adalah sinyal sinkronisasi dari tegangan sumber dihubungkan pada kaki nomor 5 ( $V_{\text{SYNC}}$ ) melalui resistor berhambatan tinggi. Perab nol (*zero detector*) akan menentukan letak titik nol dan disimpan dalam memori sinkron. *Detector* ini kemudian akan mengendalikan generator gelombang tegangan gigi gergaji yang sesuai dengan frekuensi sumber tegangan. Kapasitor  $C_{10}$  dan resistor  $R_9$  akan menentukan kemiringan dari gelombang gigi gergaji yang dihasilkan. Nilai kapasitansi kapasitor antara 500 pF sampai 1  $\mu\text{F}$  dan  $R_9$  dapat diperoleh dari resistor dengan resistansi 3 k $\Omega$  sampai 100 k $\Omega$ . Gelombang ini kemudian dibandingkan dengan tegangan referensi  $V_{11}$  oleh *comparator* (pembanding). Sinyal *output* dari *comparator* ini kemudian akan diteruskan ke rangkaian logika. Bila tegangan referensi pada kaki 11 ( $V_{11}$ ) pada posisi terendah, maka sudut penyulutan akan menunjukkan  $\alpha = 0^\circ$ . Sehingga untuk mengatur pemicuan dapat dilakukan memvariasi tegangan yang masuk pada kaki 11 (tegangan kontrol). Persamaan yang dapat digunakan untuk mendapatkan sudut pemicuan adalah:

$$\alpha = \frac{V_{\text{CONTROL}}}{V_{\text{SAT}}} \times 180^\circ \quad (2.20)$$

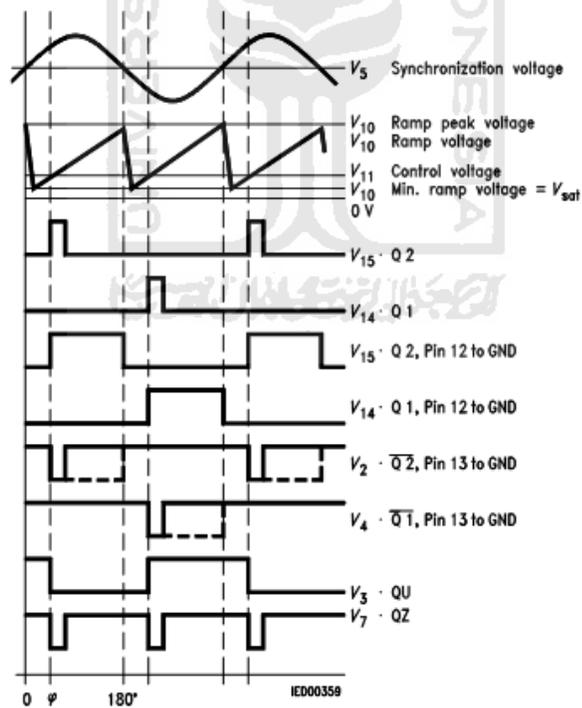
dimana:

$\alpha$  = Sudut pemicuan ( $^\circ$ )

$V_{\text{CONTROL}}$  = Tegangan kontrol,  $V_{11}$  (Volt)

$V_{\text{SAT}}$  = Tegangan pada pin 10,  $V_{10}$  (Volt)

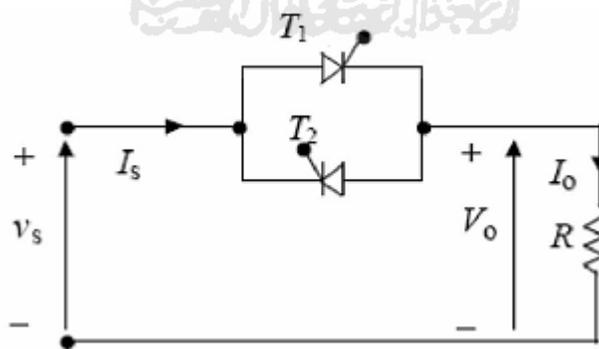
$V_{sat}$  ( $V_{10}$ ), biasanya bernilai 10 V, namun lebih tepatnya pengaturan sudut picu dapat dilakukan dengan menggunakan *oscilloscope*, sehingga dapat diperoleh pengukuran yang lebih baik. Pada IC TCA 785, jika  $V_{sat}$  ( $V_{10}$ ) melebihi tegangan kontrol  $V_{11}$ , maka sebuah sinyal akan diproses dalam logika. Proses ini dapat dilakukan berdasarkan besarnya tegangan kontrol  $V_{11}$  sehingga didapatkan sudut  $\alpha$  antara  $0^\circ - 180^\circ$ . Keluaran dari IC ini adalah pada kaki 14 (negatif) dan 15 (positif). Diagram pulsa dari proses pembangkitan sinyal pada IC TCA 785 dapat dilihat pada gambar 2.19.



Gambar 2.19 Diagram pulsa IC TCA 785

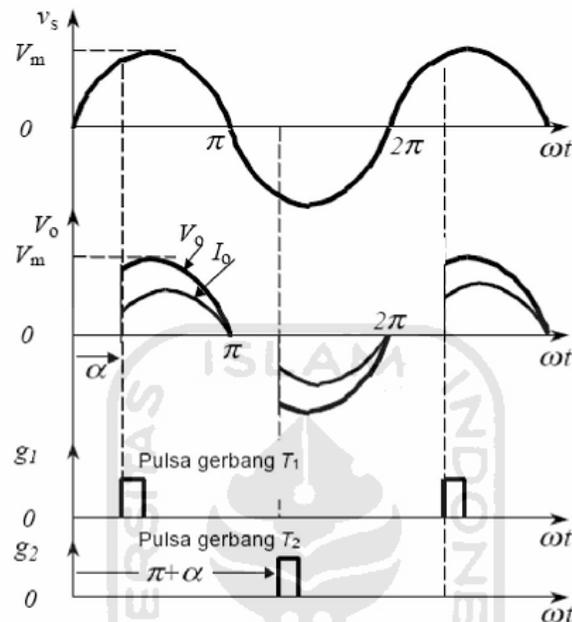
## 2.6 Kontrol Fasa Tegangan AC 1 Fasa

Pengaturan terhadap tegangan AC dapat dilakukan dengan pemotongan gelombang sinusoida. Pemotongan terhadap gelombang sinus menyebabkan nilai efektif arus berkurang sehingga daya yang disupply ke beban juga berkurang. Pemotongan gelombang sinus dapat dilakukan menggunakan *thyristor*. Kontrol tegangan AC menggunakan *thyristor* untuk mengontrol gelombang penuh satu fasa ditentukan oleh kuantitas pemotongan gelombang sinusoida. Prinsip dari kontrol sudut fasa untuk gelombang penuh satu fasa dapat dijelaskan berdasarkan rangkaian pada gambar 2.20. Energi mengalir ke beban dikontrol dengan menunda sudut pemicuan *thyristor*  $T_1$  dan sudut pemicuan *thyristor*  $T_2$ . Selama tegangan masukan setengah siklus positif, daya yang mengalir dikontrol oleh beberapa sudut tunda dari *thyristor*  $T_1$ , dan *thyristor*  $T_2$  mengontrol daya selama tegangan masukan setengah siklus negatif. Pulsa-pulsa yang dihasilkan pada  $T_1$  dan  $T_2$  terpisah  $180^\circ$ .



Gambar 2.20 Rangkaian pengontrol gelombang penuh satu fasa

Bentuk gelombang untuk tegangan masukan, tegangan keluaran, dan sinyal gerbang untuk  $T_1$  dan  $T_2$  ditunjukkan pada Gambar 2.21.



Gambar 2.21 Bentuk gelombang pengontrol gelombang penuh satu fasa

Jika  $V_s = \sqrt{2} V_s \sin \omega t$  adalah tegangan masukan dan sudut tunda *thyristor*  $T_1$  dan  $T_2$  sama ( $\alpha_1 = \alpha_2 = \alpha$ ), tegangan keluaran rms dapat ditentukan dengan:

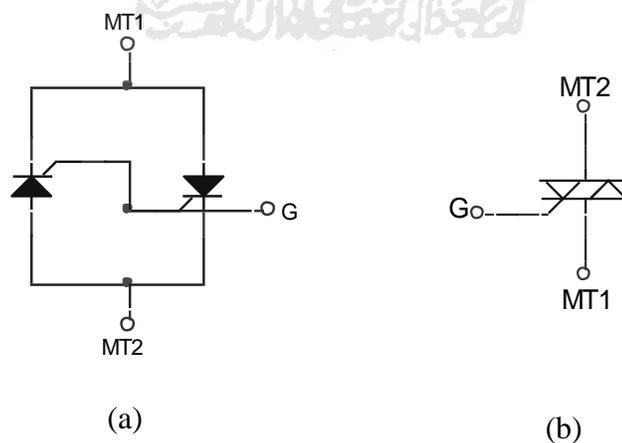
$$\begin{aligned}
 V_o &= \left[ \frac{2}{2\pi} \int_{\alpha}^{\pi} 2 V_s^2 \sin^2 \omega t d(\omega t) \right]^{\frac{1}{2}} \\
 &= \left[ \frac{1}{\pi} \int_{\alpha}^{\pi} V_s^2 (1 - \cos 2\omega t) d(\omega t) \right]^{\frac{1}{2}} \\
 &= V_s \left[ \frac{1}{\pi} \int_{\alpha}^{\pi} (1 - \cos 2\omega t) d(\omega t) \right]^{\frac{1}{2}}
 \end{aligned}$$

$$\begin{aligned}
 &= V_s \left[ \frac{1}{\pi} \left( \pi - \alpha + \frac{\sin 2\alpha}{2} \right) \right]^{1/2} \\
 &= V_s \left[ 1 - \frac{\alpha}{\pi} + \frac{\sin 2\alpha}{2\pi} \right]^{1/2} \quad (2.21)
 \end{aligned}$$

Dengan variasi sudut  $\alpha$  dari 0 sampai  $\alpha$ .  $V_o$  dapat divariasikan dari  $V_s$  sampai 0.

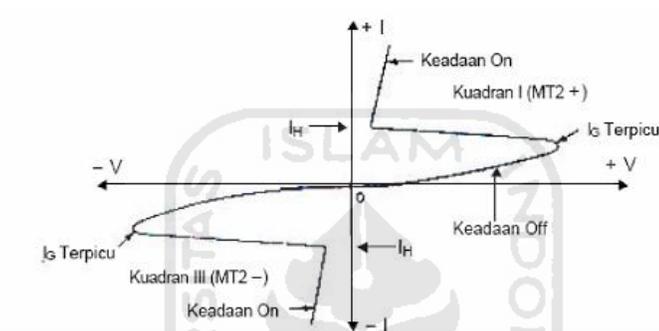
## 2.7 TRIAC

SCR (*Sillicon Controlled Rectifier*) dan TRIAC (*TRIode Alternating Current*) adalah alat *thyristor* yang paling sering digunakan. TRIAC dapat bersifat konduktif dalam dua arah. Dalam hal ini dapat dianggap sebagai dua buah *thyristor* yang terhubung *invers-paralel* dengan koneksi gerbang seperti ditunjukkan pada gambar 2.22 (a). TRIAC mempunyai tiga terminal; terminal utama 2 (MT2), terminal utama 1 atau (MT1) dan gerbang (G). Gambar 2.22 (b) menunjukkan simbol TRIAC.



Gambar 2.22 (a). Rangkaian ekivalen TRIAC, (b). Simbol TRIAK

Jika terminal MT2 positif terhadap MT1, TRIAC dapat di-*on*-kan dengan memberikan sinyal gerbang positif antara G dan MT1. Jika terminal MT2 negatif terhadap MT1 maka TRIAC dapat di-*on*-kan dengan memberikan sinyal pulsa negatif antara G dan MT1. Tidak perlu memiliki kedua sinyal gerbang positif dan negatif, TRIAC dapat dihidupkan baik oleh sinyal gerbang positif maupun negatif.



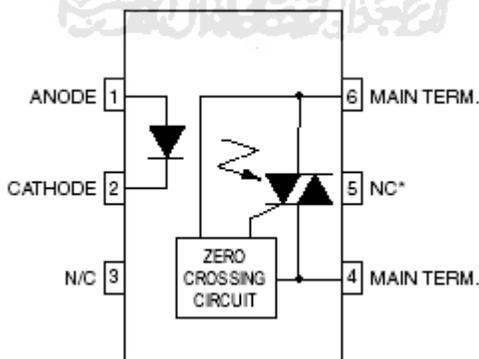
Gambar 2.23 Karakteristik TRIAC

Karakteristik V-I dari TRIAC diberikan Gambar 2.23. Arus  $I_H$ , disebut *holding current* adalah arus minimum yang dibutuhkan untuk mempertahankan TRIAC tetap *on*. TRIAC merupakan komponen yang simetris dan mampu memberikan performansi yang sama pada daerah kerja kuadran III dari grafik dengan kerja kuadran I. Sehingga TRIAC dapat dioperasikan di kuadran I (tegangan dan arus gerbang positif) atau di kuadran III (arus dan tegangan gerbang negatif).

## 2.8 Optocoupler

Guna mengendalikan peralatan listrik yang mempunyai konsumsi tegangan 220 Volt AC, diperlukan antarmuka antara tegangan tinggi dan tegangan rendah, yaitu dengan menggunakan *optocoupler*. *Optocoupler* memungkinkan konstruksi relai elektronik sederhana. *Optocoupler* dapat digunakan untuk menghindari noise dan memiliki tanggapan kerja yang cepat.

Keunggulan *optocoupler* sebagai sistem pengontrol adalah antara pengontrol dengan yang dikontrol tidak tersambung secara langsung, sehingga apabila ada kerusakan pada terkontrol dalam hal ini arus AC bolak-balik, maka tidak akan merusakkan rangkaian pengontrol. Hal ini terjadi karena secara fisik jaringan antara led dengan optodioda tidak terhubung, sehingga meskipun rangkaian yang dikontrol cukup membahayakan rangkaian pengontrol apabila terjadi hubungan langsung, dengan menggunakan *optocoupler* masalah ini dapat diatasi.



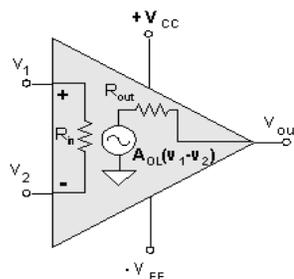
Gambar 2.24 Skema *optocoupler*

## 2.9 Penguat Operasional (Op-Amp)

Op-Amp adalah piranti yang berfungsi mengindera dan memperkuat sinyal masukan baik DC maupun AC. Op-Amp terdiri atas tiga rangkaian dasar, yaitu penguat diferensial impedansi masukan tinggi, penguat tegangan penguatan tinggi dan penguat keluaran impedansi rendah. Karakteristik Op-Amp yang terpenting adalah:

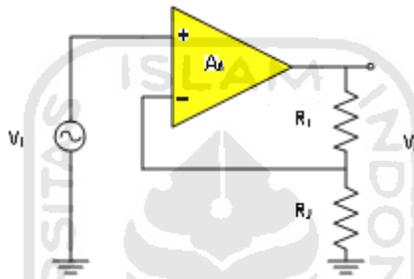
1. Impedansi masukan sangat tinggi.
2. Penguatan loop terbuka sangat tinggi.
3. Impedansi keluaran amat rendah, sehingga keluaran penguat tidak terpengaruh pembebanan.

Simbol Op-Amp standar digambarkan dengan sebuah segitiga, terminal-terminal masukan ada pada bagian atas segitiga. Masukan membalik (*inverting*) dinyatakan dengan tanda *minus* (-), tegangan DC atau AC yang diberikan pada masukan ini akan digeser fasanya  $180^\circ$  pada keluarannya. Sedangkan masukan tak membalik (*non-inverting*) dinyatakan dengan tanda *plus* (+) dan jika masukan ini diberi tegangan DC atau AC maka masukan akan sefasa dengan keluaran.



Gambar 2.25 Diagram skematis simbol Op-Amp

Op-Amp fungsi umumnya adalah sebagai penguat, tetapi dengan penambahan beberapa komponen pasif eksternal dapat digunakan untuk berbagai macam aplikasi. Salah satu aplikasi Op-Amp sebagai penguat adalah penguat *non-inverting*. Rangkaian penguat *non-inverting* ditunjukkan gambar 2.26. Sinyal yang ingin dikuatkan dimasukkan ke input *non-inverting*.



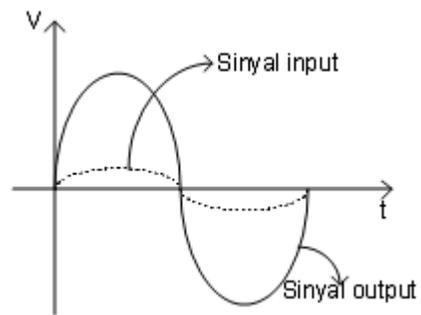
Gambar 2.26 Penguat *non-inverting*

Hubungan tegangan masukan dan keluaran diperlihatkan pada persamaan di bawah ini.

$$V_o = A \cdot V_i$$

$$V_o = \left( 1 + \frac{R_1}{R_2} \right) \cdot V_i \quad (2.22)$$

Dari persamaan di atas, tegangan keluaran mempunyai fase yang sama dengan tegangan masukan. Ilustrasi sinyal *input* dan *output* dijelaskan gambar 2.27.



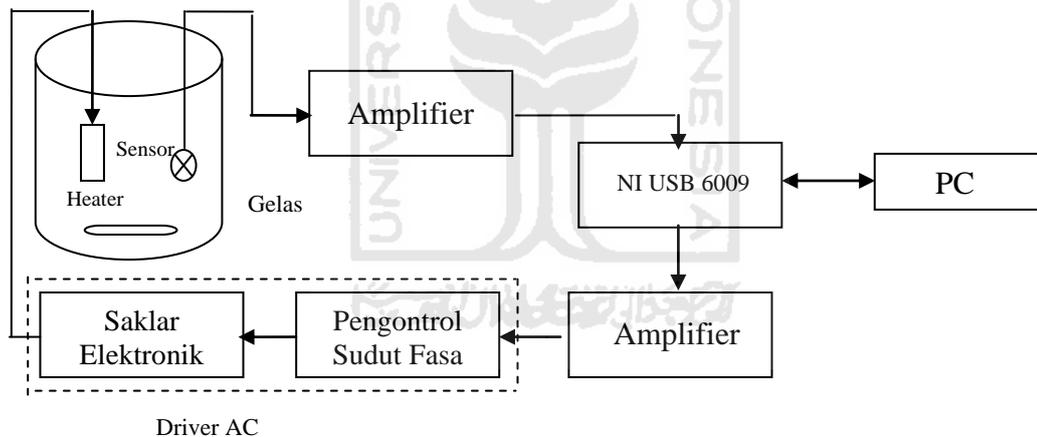
Gambar 2.27 Hubungan sinyal *input-output* penguat *non-inverting*



### BAB III

#### PERANCANGAN SISTEM

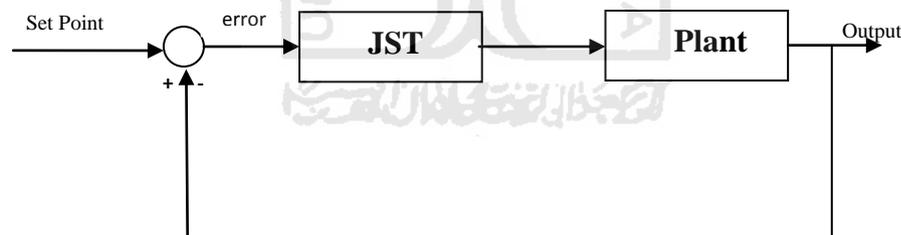
Perancangan sistem pengendali suhu *water heater* dengan algoritma jaringan syaraf tiruan hanya dirancang akan membentuk suatu sistem pengendali suhu air. Pengendalian suhu air dilakukan dengan mengatur tegangan yang diberikan ke *heater*. Pengaturan tegangan dilakukan oleh blok driver AC berdasarkan output dari pengendali dengan algoritma jaringan syaraf tiruan *backpropagation*. Blok diagram sistem ditunjukkan gambar 3.1.



Gambar 3.1 Blok diagram sistem kendali suhu *water heater*

Tabung/tangki air menggunakan sebuah gelas ukur tahan panas sampai 200 °C dengan kapasitas maksimal 1 Liter. Sebagai pemanas digunakan sebuah elemen *heater* yang biasa digunakan pada pemanas air konvensional. *Plant* juga dilengkapi dengan *stirrer*/kipas yang akan memutar air sehingga penyebaran

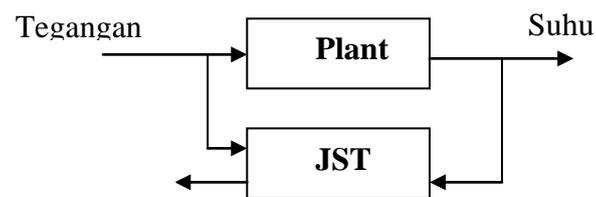
panasnya dapat lebih cepat merata. Penginderaan suhu air dilakukan oleh sensor suhu LM35 yang dibungkus dengan bahan anti air untuk menghindari terjadinya *short circuit*. Sensor suhu LM35 memiliki keluaran cukup kecil yaitu tiap kenaikan 1 °C akan menghasilkan kenaikan tegangan output sebesar 10 mV. Tegangan output sensor suhu LM35 masih dalam orde mV, sehingga perlu dikuatkan dengan menggunakan amplifiier (Op-Amp) agar dapat dibaca oleh NI USB-6009 dan dikirim ke PC sebagai data masukan. Selanjutnya data tersebut akan diproses dengan bantuan program Matlab. Sebagai pengendalian tegangan ke pemanas digunakan driver AC. Jaringan syaraf digunakan untuk mengatur besarnya tegangan yang digunakan pemanas, sehingga dapat digunakan untuk mengendalikan suhu air. Sistem kendali suhu berbasis algoritma jaringan syaraf tiruan *backpropagation* dapat dilihat pada gambar 3.2.



Gambar 3.2 Blok diagram sistem pengendali jaringan syaraf tiruan  
*backpropagation*

Saat SP (*Setting Point*) diinputkan ke dalam sistem, maka sistem akan menghitung *error* (e). Variabel *error* (e) ini akan menjadi *input* dalam JST yang berupa perangkat lunak sistem. *Output* dari JST memberikan aksi kendali pada

*plant*, dalam hal ini mengaktifkan driver AC untuk mengatur supply tegangan 0 - 220 Volt AC ke heater. Suhu *output plant* akan diumpanbalikkan guna menghitung sinyal *error*.



Gambar 3.3 Gambar invers model

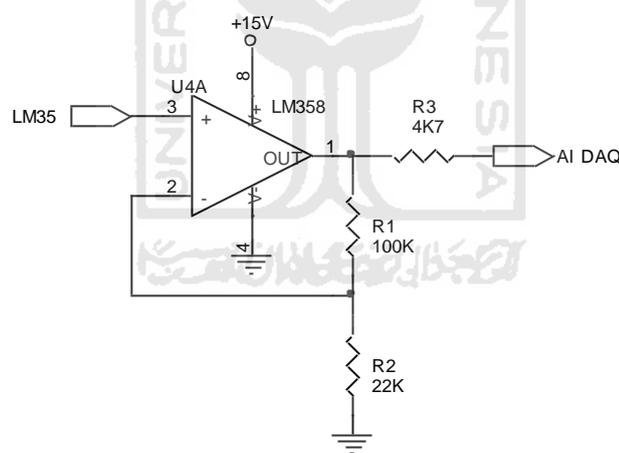
Pelatihan dari sistem pengendalian dirancang dengan menggunakan metode *inverse*, dimana masukan dari *plant* adalah sebagai target atau keluaran dari JST, sehingga skenario keluarannya akan digunakan kembali sebagai masukan. Karena pada pelatihan menggunakan metode *inverse*, maka masukan dan keluaran dari sistem kendali yang sebenarnya akan dibalik pada saat pelatihan. Pada saat pelatihan masukan dari JST adalah suhu, sedangkan keluaran atau target dari JST adalah merupakan tegangan control.

Kemampuan dari JST akan dipergunakan untuk mengidentifikasi tegangan sensor suhu. Selanjutnya hasil proses identifikasi digunakan pada proses pengendalian tegangan yang dihasilkan heater. Perangkat lunak yang digunakan adalah Matlab dan akuisisi data menggunakan NI DAQ USB-6009.

### 3.1 Rangkaian Penguat Sensor LM 35

Sensor suhu LM35 memiliki keluaran cukup kecil yaitu tiap kenaikan 1 °C akan menghasilkan kenaikan tegangan *output* sebesar 10 mV. Jadi perlu ditambahkan rangkaian penguat agar nilai tegangan yang dibaca tidak terlalu kecil. Penguatan sinyal input sebesar 5,5 kali. Keluaran dari sensor LM35 akan masuk ke Op-Amp LM358. Rangkaian penguat ini termasuk jenis *non inverting*. Perhitungan nilai penguatan dari rangkaian ini yaitu:

$$\begin{aligned}
 A &= V_o/V_i \\
 &= (1 + R_1/R_2) \\
 &= (1 + 100K/22K) = 5.5x
 \end{aligned}$$

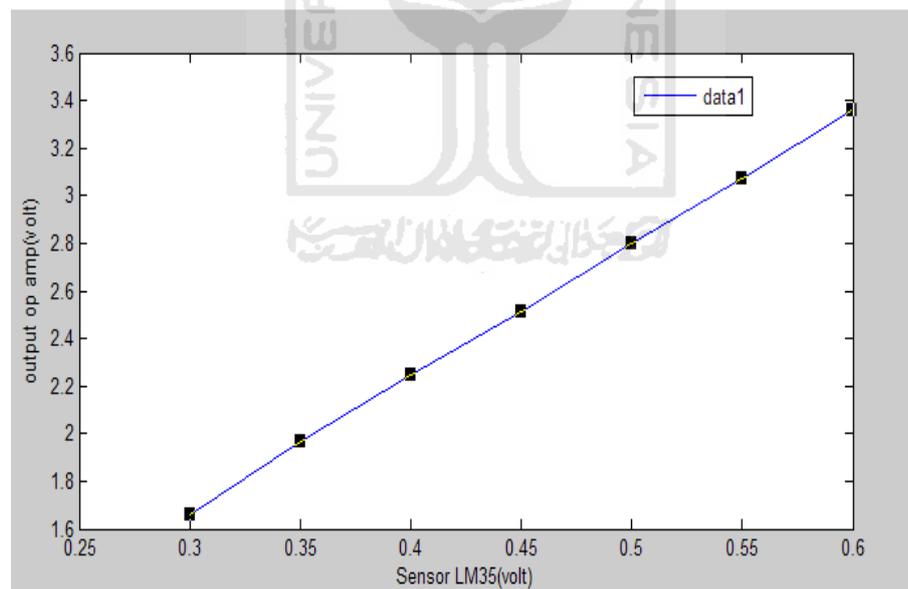


Gambar 3.4 Rangkaian penguat sensor LM35

Tabel 3.1 Penguatan sensor LM35

Sensor LM 35 (Volt)	Output Op-Amp (Volt)	Penguatan (kali)
0.30	1.66	5.53
0.35	1.97	5.62
0.40	2.25	5.62
0.45	2.51	5.57
0.50	2.80	5.60
0.55	3.07	5.58
0.60	3.36	5.60

Dari table di atas, dapat dilihat bahwa tegangan keluaran sensor LM35 cukup kecil. Tegangan input dapat dinaikkan sekitar  $\pm 5.5$  kali.

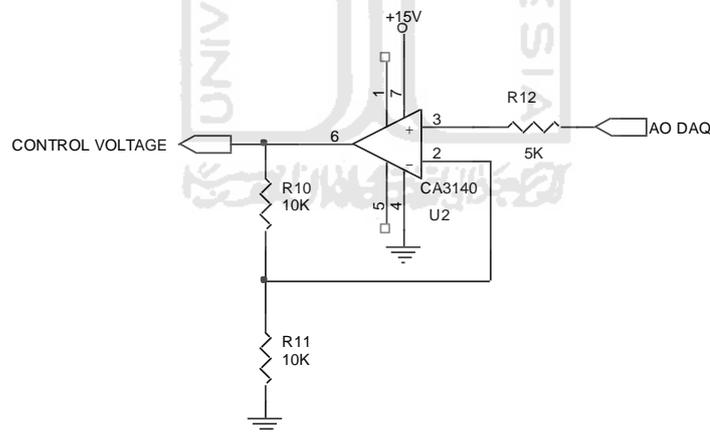


Gambar 3.5 Grafik hubungan tegangan keluaran LM35 dengan Op-Amp

Dari gambar grafik hubungan di atas dapat diketahui bahwa, semakin besar nilai tegangan yang dikeluarkan sensor LM35 maka, akan semakin bertambah besar nilai tegangan yang dikeluarkan oleh Op-Amp.

### 3.2 Penguat Tegangan Kontrol

Pada pengontrolan fasa tegangan AC memerlukan tegangan kontrol sebesar 0 sampai 10 V<sub>DC</sub> untuk menggeser pulsa 0° sampai 180°, dimana tegangan kontrol ini bersumber dari kanal *analog output* NI USB-6009. NI USB-6009 hanya dapat membangkitkan tegangan 0 sampai 5 V<sub>DC</sub>, sehingga perlu dikuatkan 2x agar dapat menggeser pulsa secara penuh. Pada rangkaian ini digunakan Op-Amp CA3140 yang difungsikan sebagai penguat *non inverting*.



Gambar 3.6 Rangkaian penguat tegangan kontrol

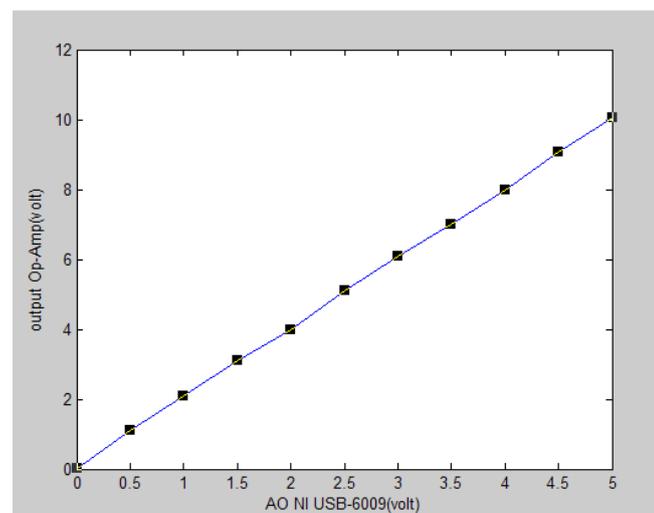
Perhitungan nilai penguatan dari rangkaian diatas adalah:

$$\begin{aligned}
 A &= V_o/V_i \\
 &= (1+ R_{10}/R_{11}) \\
 &= (1+ 10K/10K) = \mathbf{2x}
 \end{aligned}$$

Tabel 3.2 Penguat tegangan kontrol

<i>Analog Output</i> NI USB-6009 (Volt)	<i>Output Op-Amp</i> (Volt)	Penguatan
0	0.01	0
0.5	1.1	2.2
1	2.1	2.1
1.5	3.09	2.06
2	4	2
2.5	5.1	2.04
3	6.08	2.02
3.5	7	2
4	8	2
4.5	9.07	2.01
5	10.06	2.012

Berdasarkan data pada tabel 3.2, dapat dibuat grafik hubungan antara tegangan keluaran kanal *analog output* NI USB-6009 dengan tegangan setelah melalui rangkaian penguat (Op-Amp).

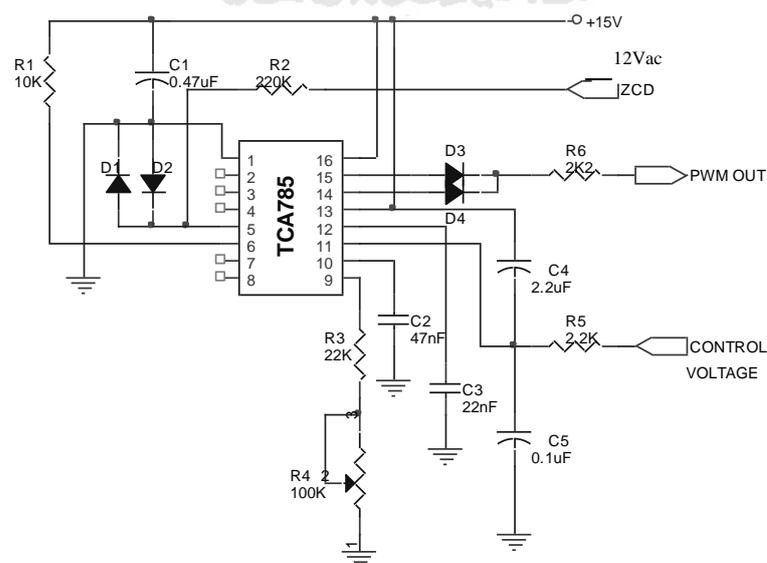


Gambar 3.7 Grafik hubungan tegangan kanal AO NI USB-6009 dengan Op-Amp

Dari table 3.2 dan gambar 3.7 di atas, dapat dilihat bahwa tegangan keluaran kanal *analog output* NI USB-6009 dapat dikuatkan dua kali dengan hubungan yang linier.

### 3.3 Rangkaian Driver AC

Rangkaian pengontrol sudut fasa pada penelitian ini menggunakan IC TCA 785, dimana didalamnya sudah terintegrasi *Zero Crossing Detector* (ZCD) dan penggeser pulsa dari  $0^\circ$  sampai  $180^\circ$ . ZCD adalah rangkaian yang digunakan untuk mendeteksi gelombang sinus AC 220 volt saat melewati titik tegangan nol. Seberangan titik nol yang dideteksi adalah peralihan dari positif menuju negatif dan peralihan dari negatif menuju positif. Seberangan-seberangan titik nol ini merupakan acuan yang digunakan sebagai awal pemberian nilai waktu tunda untuk pemicuan triac. Rangkaian pengontrol sudut fasa dapat dilihat pada gambar 3.8.



Gambar 3.8 Rangkaian pengontrol sudut fasa

Sinyal sinkronisasi berasal dari tegangan 12 V<sub>AC</sub> dari ZCD yang dihubungkan pada kaki nomor 5. Untuk menghindari arus yang sangat besar pada IC maka sebelum masuk ke kaki 5, tegangan 12 V<sub>AC</sub> dari ZCD harus diberi hambatan terlebih dahulu sebesar 220 kΩ dan juga diberi rangkaian antiparalel diode untuk memperbolehkan aliran arus reverse karena beban induktif dan untuk meningkatkan kinerja saat turn-off dari rangkaian. Diode memotong tegangan balik blocking dari ZCD pada kondisi tunak. Akan tetapi pada kondisi transien, tegangan balik dapat ditingkatkan karena tegangan induksi pada rangkaian. Besarnya arus yang masuk ke IC menjadi sangat kecil, yaitu:

$$I_{SYNC} = \frac{V_{SYNC}}{R}$$

$$I_{SYNC} = \frac{12}{220000} = 5,4 \mu A$$

dimana:

$V_{SYNC}$  = Tegangan sinkronisasi (Volt)

$I_{SYNC}$  = Arus sinkronisasi (Ampere)

$R$  = Hambatan (Ohm)

Driver AC digunakan untuk pengaturan nilai tegangan kontrol terhadap pengaturan tegangan yang di *supply* ke beban (*heater*). Tegangan kontrol yang dimasukkan merupakan konversi dari sudut picu (fasa) sinyal tegangan AC nya. Untuk mendapatkan sudut pemicuan dapat dilakukan secara manual, yaitu dengan

menggunakan persamaan yang ada. Persamaan yang digunakan untuk mendapatkan sudut pemicuan adalah:

$$\alpha = \frac{V_{CONTROL}}{V_{SAT}} \times 180^\circ \quad (3.1)$$

Tegangan yang terukur pada pin 10 IC TCA 785 adalah 5.05 V (multimeter) dan 10 V<sub>pp</sub> (CRO). Untuk mengatur sudut pemicuan dapat dengan cara mengatur tegangan kontrolnya pada pin 11 IC TCA 785. Variasi sudut yang akan dilakukan adalah 0°, 15°, 30°, 45°, 60°, 75°, 90°, 105°, 120°, 135°, 150°, 165° dan 180°.

Jika  $V_s = \sqrt{2} V_s \sin \omega t$  adalah tegangan sumber AC dan sudut picu  $\alpha$ . Dari pengukuran diketahui  $V_s = 219$  Volt. Maka kita dapat mencari tegangan keluarannya, yaitu:

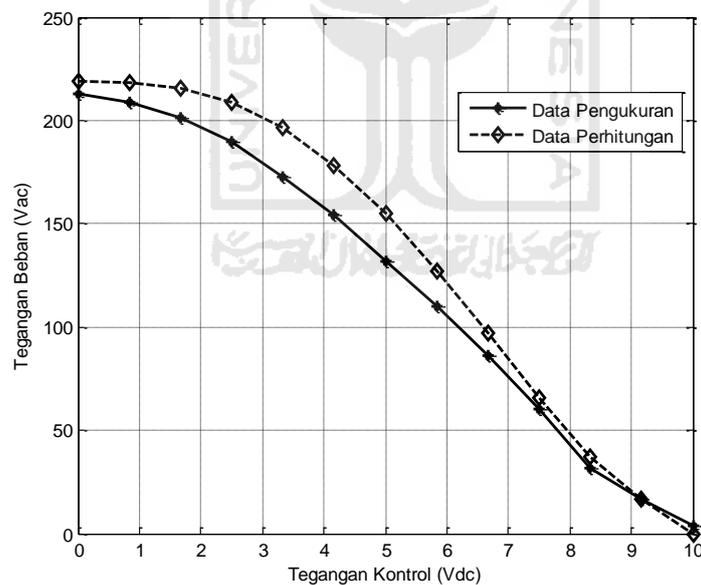
$$V_o = V_s \left[ 1 - \frac{\alpha}{\pi} + \frac{\sin 2\alpha}{2\pi} \right]^{\frac{1}{2}} \quad (3.2)$$

Tabel 3.3 Pengujian blok driver AC

Sudut Picu / $\alpha$ (°)	Tegangan Kontrol (Volt)	Tegangan Beban / $V_o$ (Volt)		Error (%)
		Pengukuran	Perhitungan	
0	0	213	219	2.7
15	0.83	209	218.45	4.3
30	1.66	201	215.91	6.9
45	2.5	190	208.79	9
60	3.33	173	196.5	11.96
75	4.16	154	178.45	13.7
90	5	132	154.85	14.7
105	5.83	110	127.13	13.4

Sudut Picu / $\alpha$ ( $^{\circ}$ )	Tegangan Kontrol (Volt)	Tegangan Beban / $V_o$ (Volt)		Error (%)
		Pengukuran	Perhitungan	
102	6.66	86	96.95	11.3
135	7.5	60	66.06	9.17
150	8.33	32	37.3	14.2
165	9.16	17	16.96	0.2
180	10	4	0	-

Tabel 3.3 merupakan data yang diambil berdasarkan perhitungan pada variasi sudut picu dan tegangan sumber masukan dari jala-jala PLN beserta tegangan keluaran yang dihasilkan.



Gambar 3.9 Grafik hubungan tegangan kontrol dengan tegangan beban

Gambar 3.9 merupakan kurva hubungan tegangan keluaran dengan tegangan kontrol/sudut picu dimana semakin besar tegangan kontrol/sudut picu

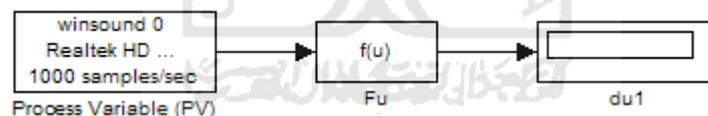
maka semakin kecil tegangan keluarannya, demikian juga sebaliknya. Perubahan yang terjadi berlangsung secara perlahan dan teratur.

Harga *error* ditentukan oleh persamaan:

$$Error = \left( \frac{Data\ perhitungan - Data\ pengukuran}{Data\ perhitungan} \right) \times 100\% \quad (3.3)$$

### 3.4 Analog Input pada Matlab

Kanal ini akan mengirimkan nilai tegangan keluaran dari rangkaian penguat sensor suhu LM35 dari rangkaian pengontrol fasa untuk diproses lebih lanjut pada Matlab.



Gambar 4.10 Blok untuk *analog input*

Agar terjadi kesamaan antara suhu yang tertampil pada *display* di simulink dengan suhu yang terukur pada termometer di *plant*, maka diperlukan adanya kalibrasi. Kalibrasi ini dilakukan dengan cara membuat perhitungan linearitas, yaitu dengan membandingkan tegangan keluaran penguat sensor LM35 yang terbaca pada *display* dengan suhu air yang terukur pada termometer.

Tabel. 3.4 Data penunjukkan termometer dan tegangan pada Matlab

Nilai pada termometer (°C)	Tegangan pada Matlab (Volt)
30	2,1
35	2,32
40	2,6
45	2,85
50	3,10
55	3,37
60	3,63
65	3,89
70	4,15
75	4,4
80	4,65

Untuk perhitungan linearitas dari data-data di atas yaitu:

misal:

$x$  = nilai yang tertampil pada matlab

$y$  = nilai yang akan dibandingkan (dari termometer)

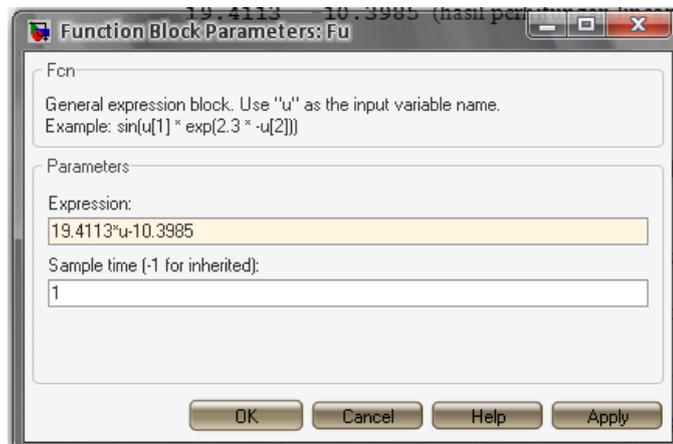
maka:

`polyfit (x,y,1)`

`ans =`

**19.4113 -10.3985** (hasil perhitungan linearitas)

(4.4)



Gambar 3.11 Blok manipulasi suhu pada Matlab

Blok manipulasi di atas dibuat pada Matlab untuk *input* tegangan suhu yang dihasilkan LM35. Nilai 19.4113 dan -10.3985 dihasilkan dari perhitungan linearitas. Tegangan suhu dari penguat sensor suhu LM35 yang masuk kemudian dikalikan dengan 19.4113 selanjutnya dikurangi dengan 10.3985, sehingga suhu yang tertampil pada Matlab akan sama dengan suhu yang terbaca pada termometer.

### 3.5 Analog Ouput pada Matlab

Pengujian ini bertujuan untuk memastikan kanal *analog output* akuisisi data NI USB-6009 dapat membangkitkan tegangan sesuai dengan perintah di Matlab. Pengujian dilakukan dengan cara Matlab mengirimkan tegangan 0 sampai 5 Volt ke kanal *analog output* NI USB-6009.

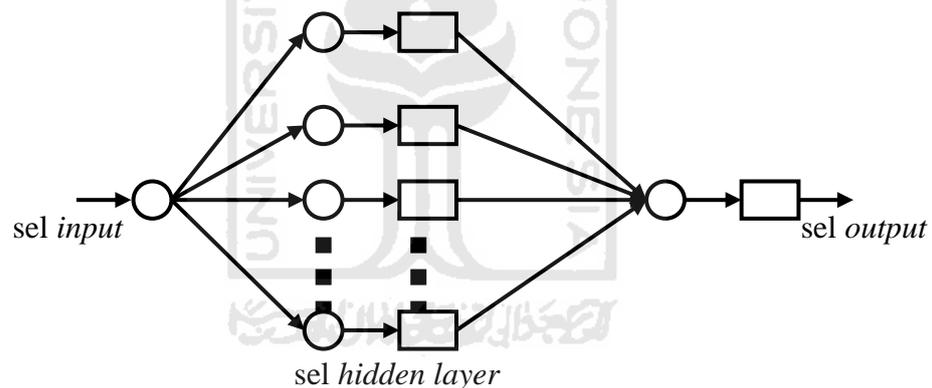


Gambar 4.12 Blok untuk *analog output*

Tegangan keluaran NI USB-6009 dapat diatur secara manual, yaitu dengan memasukkan nilai pada *constan* antara 0-5 maka tegangan keluaran NI USB-6009 akan berubah sesuai dengan perubahan nilai yang terjadi pada *constan*. Tegangan keluaran ini akan digunakan sebagai tegangan kontrol untuk mengatur tegangan yang diberikan ke pemanas.

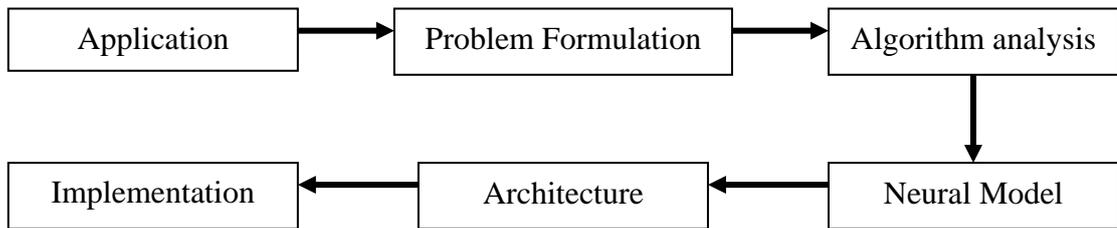
### 3.6 Perancangan Jaringan Syaraf Tiruan

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



Gambar 3.13 Arsitektur jaringan syaraf tiruan

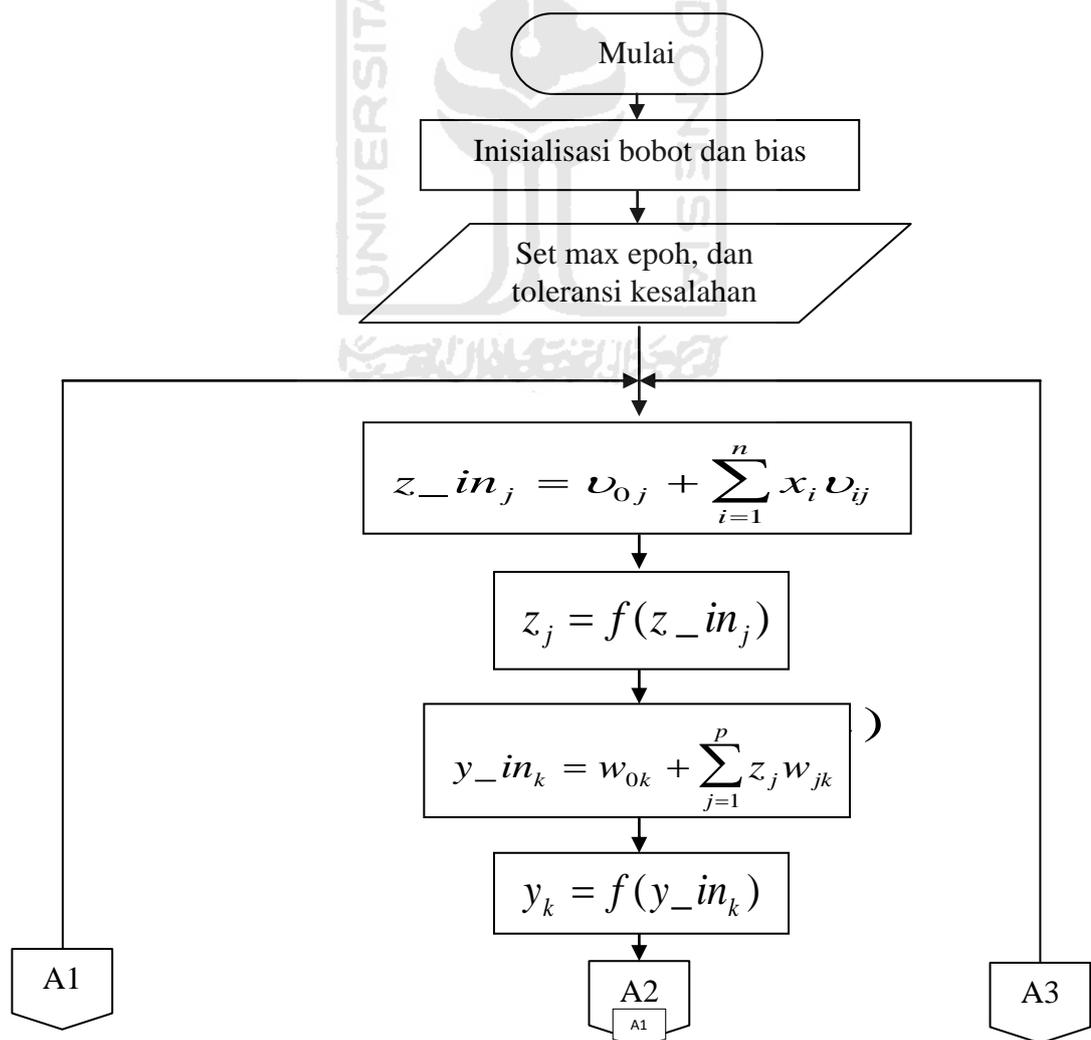
Jaringan syaraf tiruan yang digunakan sebagai pengendali suhu air ini menggunakan metode *backpropagation*. Metode *backpropagation* termasuk jenis jaringan yang *autoassociative*, yaitu *range* masukan yang di proses ke dalam jaringan sama dengan *range* hasil yang dikeluarkannya. Proses pembangunan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

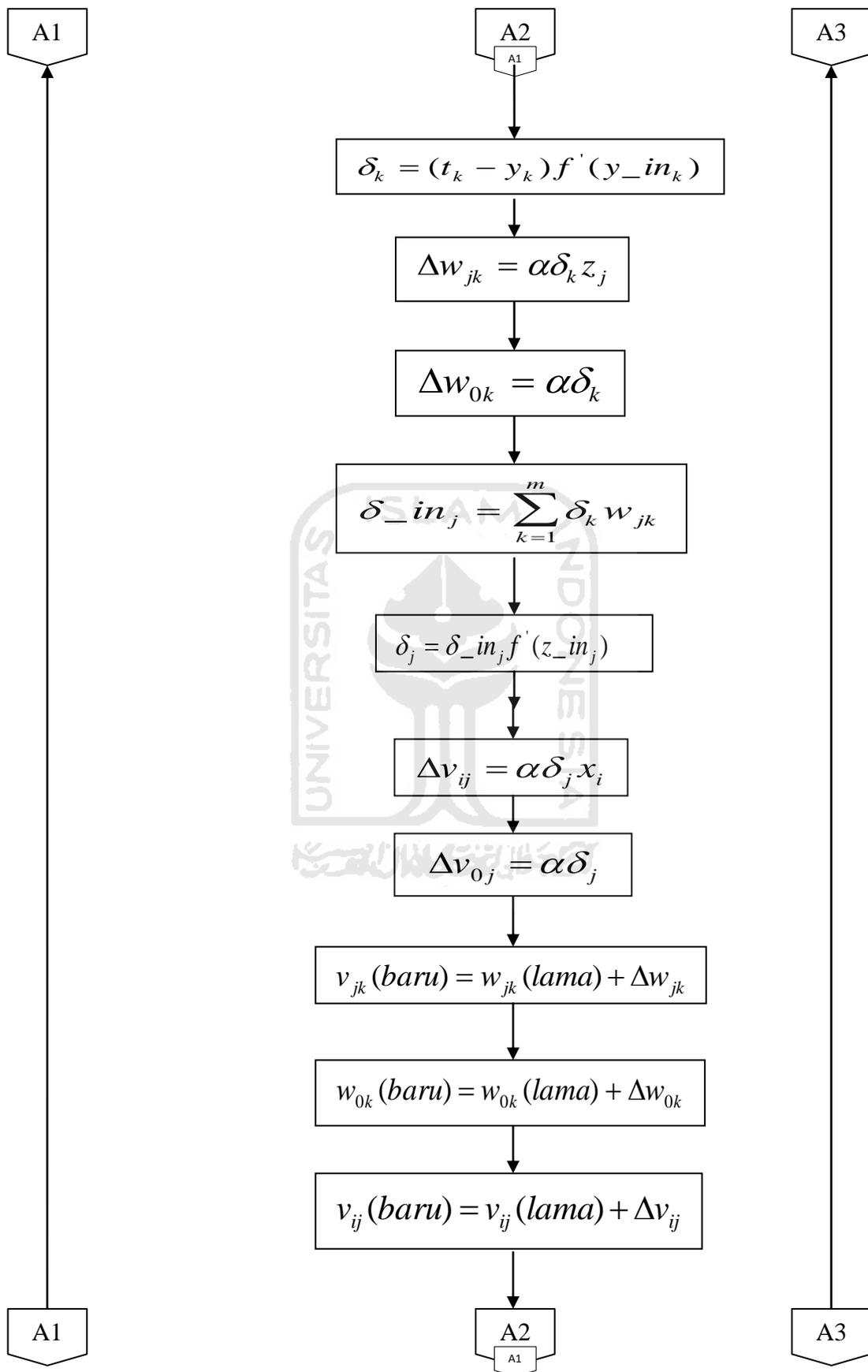


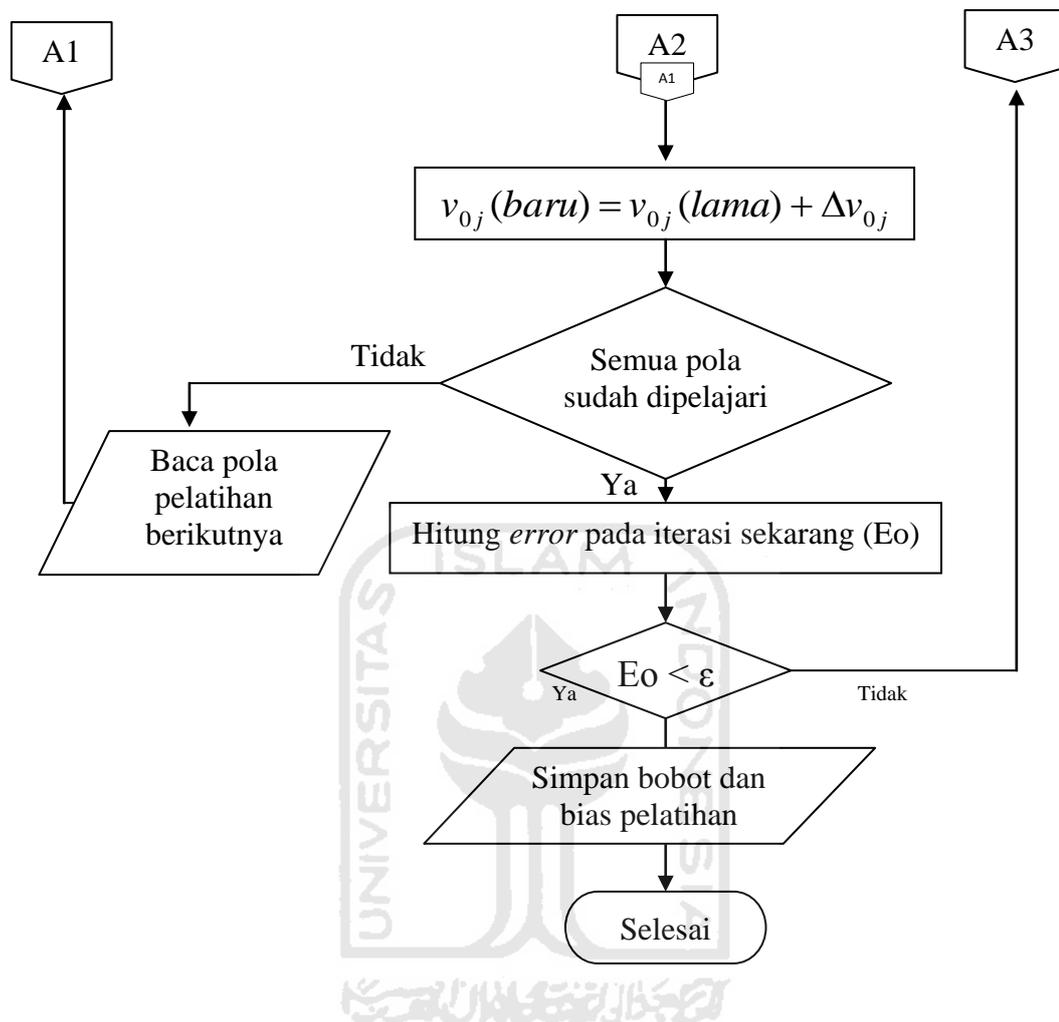
Gambar 3.14 Proses pembangunan jaringan syaraf tiruan

### 3.6.1 Diagram Alir Jaringan Syaraf Tiruan Metode *Backpropagation*

Dengan mengacu pada langkah-langkah prosedur pelatihan, maka diagram alir untuk prosedur pelatihannya adalah sebagai berikut :





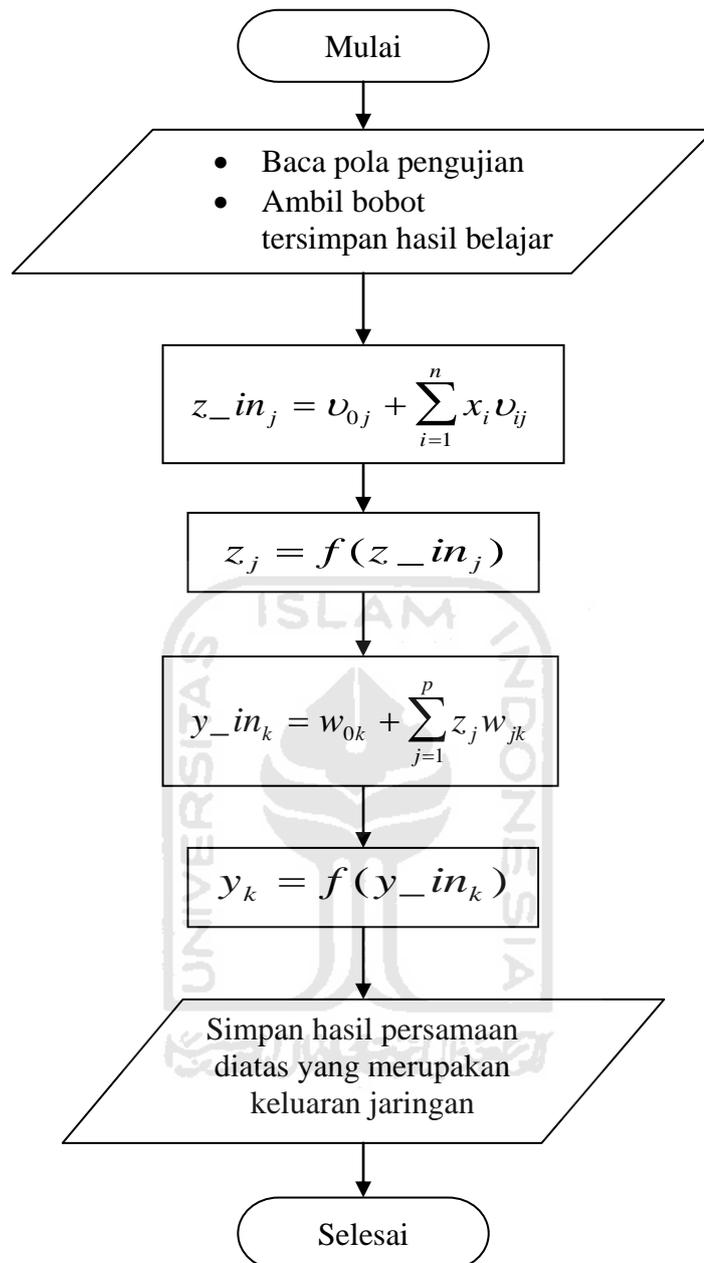


Gambar 3.15 Diagram alir/flowchart prosedur pelatihan

Setelah jaringan belajar dan menyimpan bobot-bobot dan biasnya, 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.

### 3.6.2 Diagram Alir Prosedur Pengujian

Dengan mengacu pada langkah-langkah prosedur pengujian, maka diagram alir prosedur pengujiannya adalah sebagai berikut :

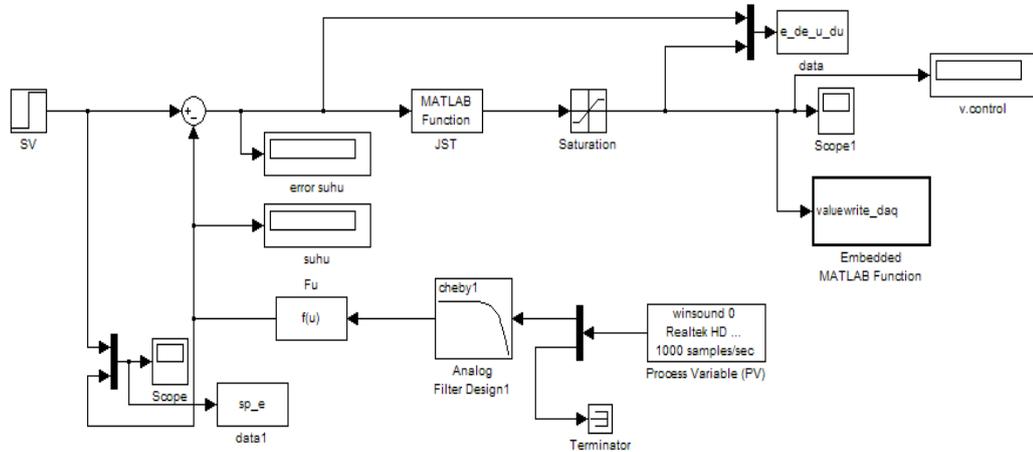


Gambar 3.16 Diagram alir/flowchart prosedur pengujian

### 3.7 Perancangan Simulink

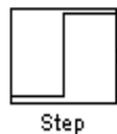
Sistem yang akan dibuat di kerjakan menggunakan *toolbox simulink*. Begitu juga dengan komponen-komponen jaringan syaraf tiruan yang dimodelkan dengan persamaan matematis ataupun yang menggunakan program.

## Blok Rangkaian Simulink



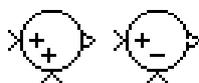
Gambar 3.17 Blok rangkaian simulink

### 1. *Step*



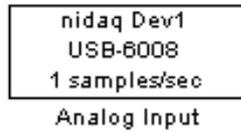
Blok ini digunakan untuk memberikan parameter nilai dimana nilai tersebut digunakan sebagai set poin dari sistem

### 2. *Sum*



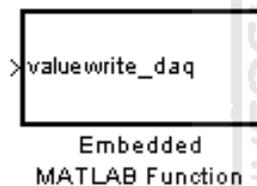
Merupakan blok penjumlahan atau pengurang input sesuai dengan tanda operasinya. Blok ini menerima input berupa skalar, vektor, matrik atau elemen dari vektor tunggal. Bila blok memiliki lebih dari satu input, kedua input tersebut harus vektor atau matrik yang memiliki dimensi sama.

### 3. *Analog input*



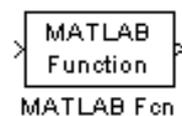
Blok untuk memperoleh data dari saluran analog perangkat akuisisi data NI USB 6009. Didalam Blok analog input kanal AI dapat dikonfigurasi dalam mode sinyal *reference single ended* (RSE) dan mode sinyal diferensial.

### 4. *Embedded matlab function/analog output*



Karena blok analog output dari data acquisition toolbox yang tersedia tidak ada internal clocking, untuk output data ke saluran analog dari perangkat akuisisi data menggunakan blok *embedded matlab function*. Dengan mengisi dengan perintah yang digunakan untuk menulis operasi ke saluran analog output.

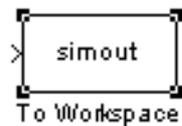
### 5. *Matlab function*





vektor. Jika semua inputnya vektor maka outputnya vektor. Jika beberapa inputnya bukan vektor, misalnya matrik. Maka output dari *mux* adalah sinyal bus.

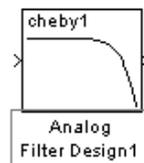
#### 9. *To Workspace*



Blok ini menyimpan data dari hasil simulasi ke *workspace* Matlab.

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

#### 10. *analog filter design*



Blok ini merupakan bentuk filter analog standar. Filter yang digunakan chebyshev tipe *lowpass*. Dengan blok filter pembacaan dari sensor oleh NI USB DAQ 6009 dapat diperhalus, sehingga data yang dibaca pada matlab dapat sesuai dengan keluaran sebenarnya yang dihasilkan sensor.

## BAB IV

### PENGUJIAN, ANALISIS DAN PEMBAHASAN

Sebelum melakukan pelatihan jaringan syaraf tiruan, terlebih dahulu dilakukan pengambilan data secara *open loop* untuk menentukan input dan target pelatihan. Pengambilan data dilakukan dengan memberi nilai tegangan acak rangkaian open loop pada blok *simulink* dengan menggunakan blok *uniform random number* yang kemudian data disimpan ke workspace. Dari pengambilan data tersebut dapat ditentukan input dan target pelatihan berupa error suhu sebagai input pelatihan dan tegangan kontrol sebagai target dari pelatihan.

#### 4.1 Pelatihan Jaringan Saraf Tiruan *Backpropagation*

Pada pelatihan jaringan saraf tiruan sebagai pengendali suhu air, menggunakan model *Levenberg-Marquardt*. Beberapa pengamatan yang dilakukan yaitu jumlah fungsi aktivasi, 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 fungsi aktivasi yang digunakan.

Penjelasan tabel hasil pelatihan dan tabel hasil pengujian :

No : Nomor.

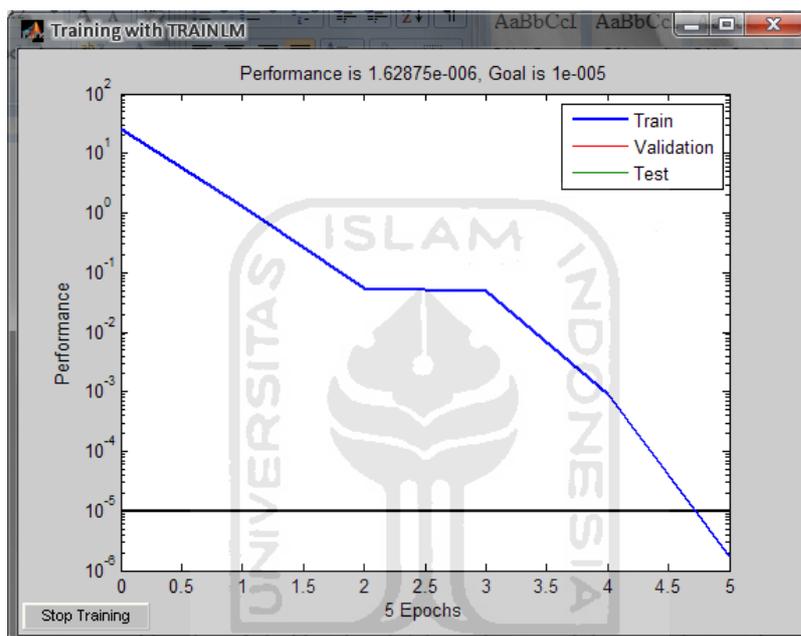
- HL : Jumlah lapisan tersembunyi pada jaringan syaraf (*hidden layer*) .
- Neuron : Angka pertama adalah jumlah neuron lapisan masukan, angka berikutnya adalah jumlah neuron lapisan tersembunyi. (angka 1 dibelakang menunjukkan jaringan terdiri dari 1 target)
- F. Aktivasi : Fungsi aktivasi pada setiap 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.

Perintah yang digunakan untuk membentuk jaringan adalah menggunakan `newff`. Pelatihan yang dilakukan matlab menggunakan tiga variasi fungsi aktivasi. Fungsi pelatihan jaringan pada sistem kontrol suhu ini menggunakan `default = trainlm` (*Levenberg-Marquardt*).

Tabel 4.1 Hasil pelatihan dengan 1 lapisan tersembunyi.

No.	HL	Neuron	F. aktivasi	Iterasi	MSE	Ket.
1	1	10 – 1	<i>logsig – logsig</i>	794	5.82706	Max
2	1	10 – 1	<i>logsig – purelin</i>	6	4.44072-007	iterasi
3	1	10 – 1	<i>logsig – tansig</i>	487	5.9347	Max
4	1	10 – 1	<i>purelin – logsig</i>	2	5.93474	gradient
5	1	10 – 1	<i>purelin – purelin</i>	5	0.0477542	gradient
6	1	10 – 1	<i>purelin – tansig</i>	39	5.82801	Max
7	1	10 – 1	<i>tansig – logsig</i>	794	5.82706	Max
<b>8</b>	<b>1</b>	<b>10 – 1</b>	<b><i>tansig – purelin</i></b>	<b>5</b>	<b>1.62875e-006</b>	<b>iterasi</b>
9	1	10 – 1	<i>tansig – tansig</i>	593	5.82706	Max

Pada pelatihan dengan 1 lapisan tersembunyi, pelatihan terbaik diperoleh oleh pelatihan no 8. Yaitu pelatihan dengan fungsi aktivasi sigmoid bipolar-fungsi identitas yang memiliki nilai MSE  $1.62875e-006$ . Pelatihan berhenti saat iterasi ke 29.

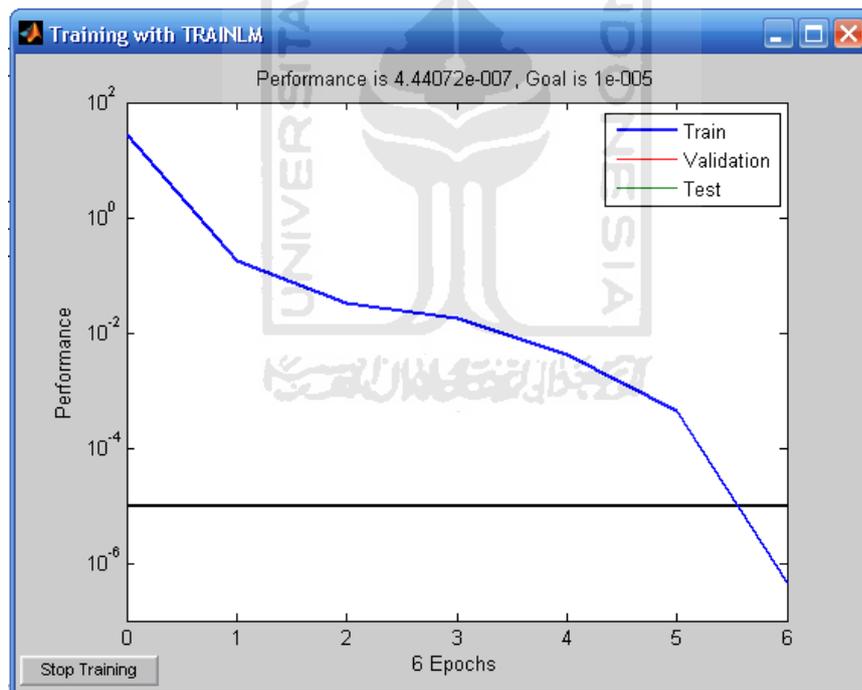


Gambar 4.1 Hasil pelatihan dengan 1 lapisan tersembunyi  
f fungsi aktivasi *sigmoid bipolar-identitas*

Sedangkan untuk pelatihan no 1,3,6,7 dan 9 iterasi dan MSE tidak mencapai target dan pelatihan terhenti karena nilai maximum mu telah tercapai/melebihi dari nilai *default*-nya sebesar  $1e10$ .

Untuk pelatihan no 4 dan 5, iterasi dan MSE tidak tercapai. Pelatihan terhenti karena nilai gradient sudah mencapai nilai target (*default* =  $1e-10$ ).

Sedangkan pelatihan no 2 dengan fungsi aktivasi sigmoid biner-fungsi aktivasi identitas target MSE tercapai sebesar  $4.44072 \times 10^{-7}$  pada saat iterasi ke 6. Dari hasil tersebut pelatihan no 2 memiliki waktu itersi yang lebih cepat jika dibandingkan dengan pelatihan no 8. Akan tetapi target yang dicapai masih lebih rendah dibanding pelatihan no 8. Sehingga hasil pelatihan no 8 masih lebih baik jika dibandingkan dengan pelatihan no 2 walaupun waktu iterasinya yang dicapai lebih cepat. Dari pelatihan dengan 1 lapisan tersembunyi pelatihan yang menggunakan fungsi identitas dapat mencapai target MSE karena target pelatih



Gambar 4.2 Hasil pelatihan dengan 1 lapisan tersembunyi  
fungsi aktivasi *sigmoid biner-identitas*

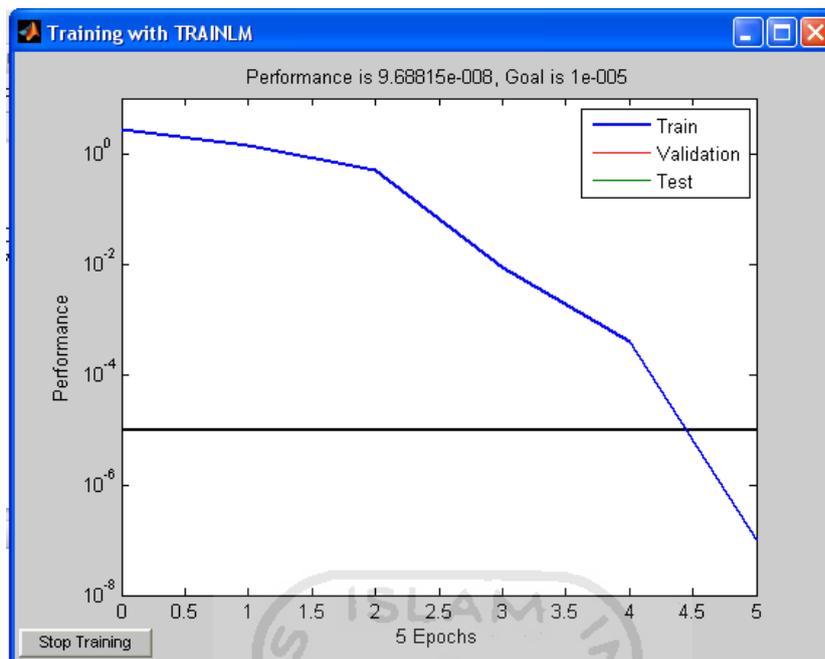
#### 4.1.2 Pelatihan Menggunakan Lebih dari 1 Lapisan Tersembunyi

Berikut ini adalah data hasil pengamatan dari pelatihan dengan menggunakan lebih dari 1 lapisan tersembunyi dan dikelompokkan kedalam tabel-tabel sesuai fungsi aktivasi yang digunakan dan jumlah lapisan tersembunyi.

Tabel 4.2 Hasil pelatihan dengan lebih dari 1 lapisan tersembunyi

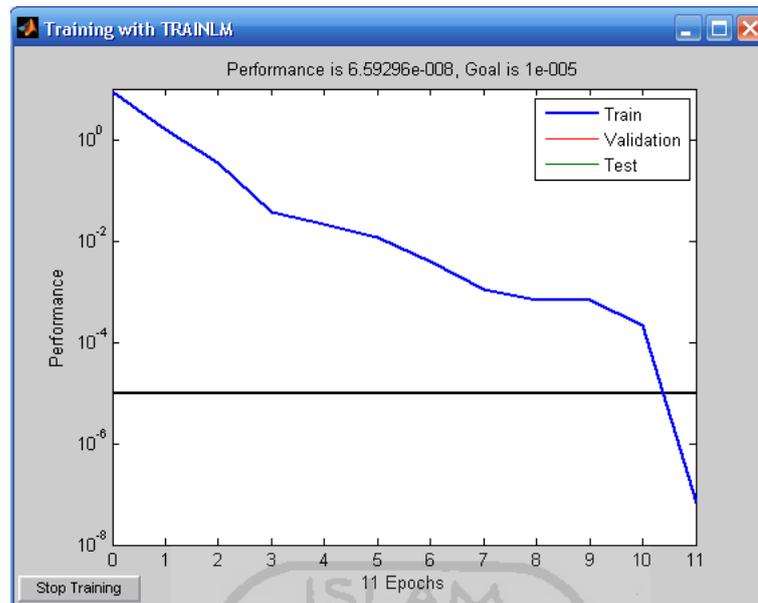
No.	HL	Neuron	F. aktivasi	Iterasi	MSE	Ket.
1	2	10-5-1	<i>logsig - logsig - logsig</i>	2	5.93474	gradient
2	2	10-5-1	<i>purelin - logsig - logsig</i>	7	5.93474	gradient
3	2	10-5-1	<i>purelin - purelin - logsig</i>	2	5.93474	gradient
<b>4</b>	<b>2</b>	<b>10-5-1</b>	<b><i>Tansig-tansig-purelin</i></b>	<b>5</b>	<b>9.68815/1e-008</b>	<b>iterasi</b>
5	2	10-5-1	<i>tansig - purelin - purelin</i>	12	3.92692e-006	iterasi
6	2	10-5-1	<i>Logsig - tansig - purelin</i>	5	6.60503e-006	iterasi
7	2	10-5-1	<i>Logsig - logsig - purelin</i>	7	2.16823e-006	iterasi
8	2	10-5-1	<i>logsig - logsig - logsig</i>	2	5.93474	gradient
9	2	10-5-1	<i>purelin - logsig - logsig</i>	7	5.93474	gradient
10	2	10-5-1	<i>Purelin-purelin-purelin</i>	4	0.0477542	gradient

Pada pelatihan dengan lebih dari 1 lapisan tersembunyi dihasilkan fungsi aktivasi sigmoid bipolar-sigmoid bipolar-fungsi identitas yang memiliki nilai MSE paling baik yaitu sebesar 9.68815/1e-008 pada iterasi ke 5.

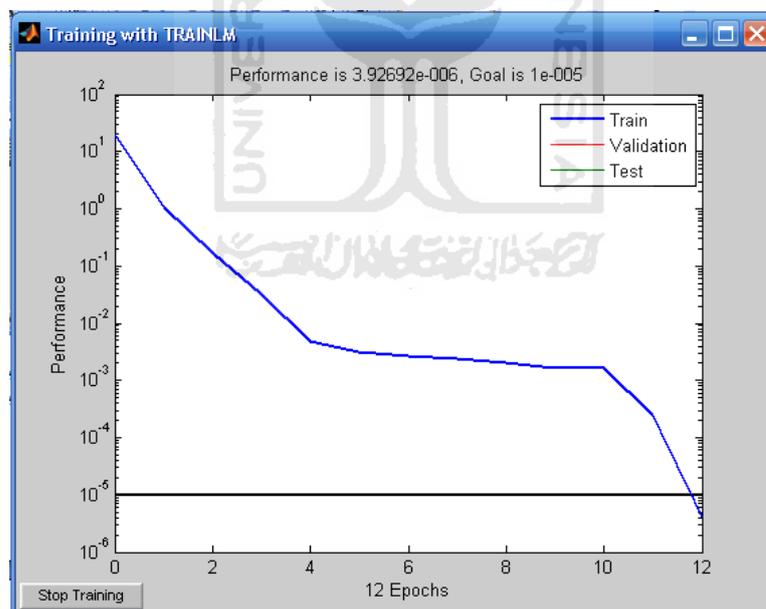


Gambar 4.3 Hasil pelatihan dengan 2 lapisan tersembunyi  
fungsi aktivasi *sigmoid bipolar-bipolar-identitas*

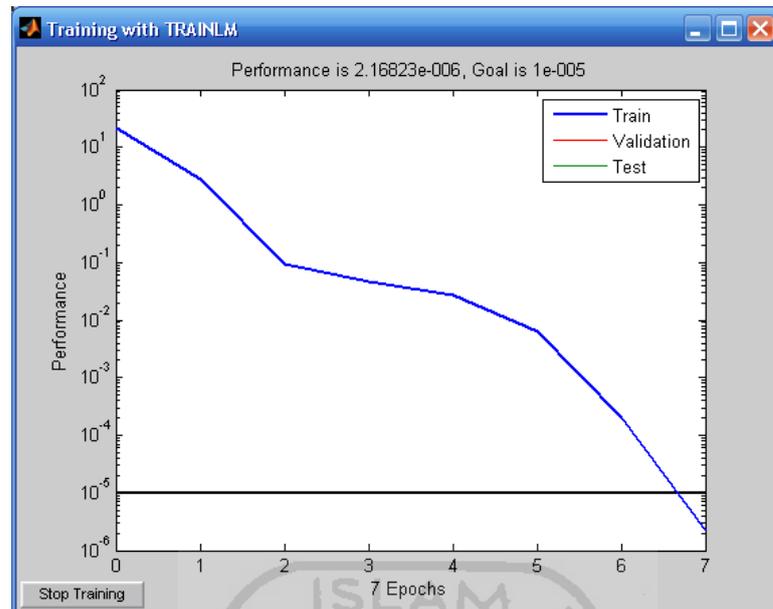
Dari tabel diatas beberapa pelatihan juga mencapai target MSE, seperti pelatihan no 5 dengan fungsi aktivasi bipolar-identitas-identitas, pelatihan no 6 dengan fungsi aktivasi bipolar-biner-identitas dan pelatihan no 7 dengan fungsi aktivasi biner-biner-identitas. Sedangkan pelatihan no 1,2,3,8,9 dan 10 tidak mencapai target MSE dan iterasi karena sudah mencapai/melebihi nilai *maximum* gradient yaitu  $1e-10$ . Dari tabel dapat dilihat bahwa pelatihan yang menggunakan target dengan fungsi aktivasi identitas dapat mencapai target MSE kecuali no 10. Hal tersebut terjadi karena data pelatihan dan fungsi aktifasi yang digunakan tidak sebanding atau sesuai.



Gambar 4.4 Hasil pelatihan dengan 2 lapisan tersembunyi fungsi aktivasi *sigmoid biner-sigmoid bipolar-identitas*



Gambar 4.5 Hasil pelatihan dengan 2 lapisan tersembunyi fungsi aktivasi *sigmoid biner-identitas-identitas*



Gambar 4.6 Hasil pelatihan dengan 2 lapisan tersembunyi fungsi aktivasi *sigmoid biner-sigmoid biner-identitas*

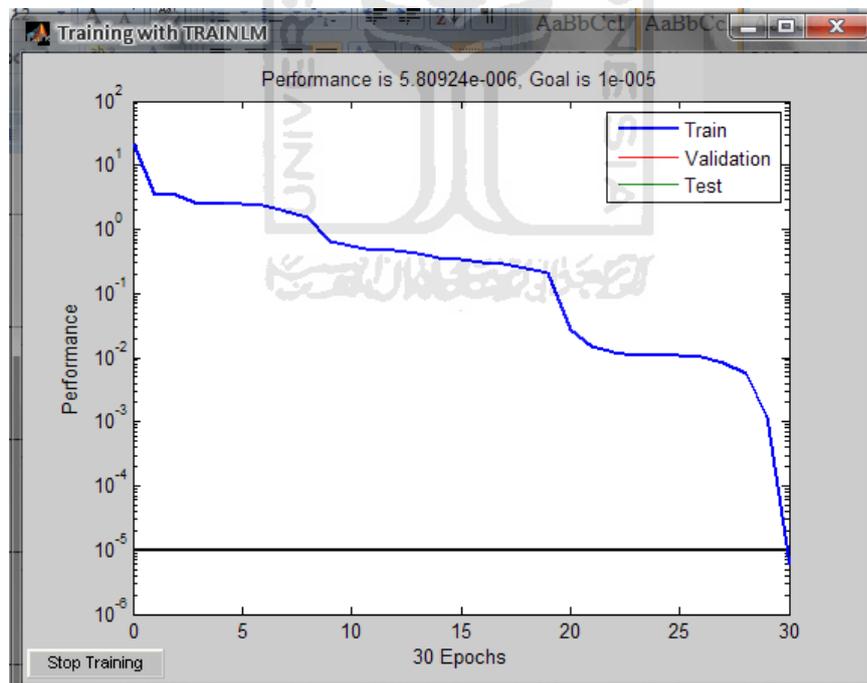
#### 4.1.3 Pelatihan Menggunakan Lebih dari 2 Lapisan Tersembunyi

Tabel 4.3 Hasil pelatihan dengan jumlah lapisan tersembunyi lebih dari 2

No.	HL	Neuron	F. aktivasi	Iterasi	MSE	Ket.
1	3	10-3-2-1	<i>Tansig-tansig-tansig-purelin</i>	126	9.9617e-008	iterasi
2	3	10-3-2-1	<i>Logsig-logsig-logsig-purelin</i>	8	2.49408	gradient
3	3	10-3-2-1	<i>Logsig-logsig-logsig-logsig</i>	2	5.93474	gradient
4	3	10-3-2-1	<i>Tansig-tansig-tansig-tansig</i>	2	5.93474	gradient
5	3	10-3-2-1	<i>Purelin-purelin-purelin-purelin</i>	7	0.0477542	gradient
6	3	10-3-2-1	<i>Logsig-tansig-logsig-tansig</i>	419	5.82706	Max mu
7	4	10-5-3-2-1	<i>Tansig-logsig-tansig-logsig-purelin</i>	30	5.80924e-006	iterasi
8	4	10-5-3-2-1	<i>Logsig-logsig-logsig-logsig-logsig</i>	7	5.93474	gradient
9	4	10-5-3-2-1	<i>Tansig-tansig-tansig-tansig</i>	1	5.93474	gradient

No.	HL	Neuron	F. aktivasi	Iterasi	MSE	Ket.
10	4	10-5-3-2-1	<i>Logsig-logsig-logsig-logsig-purelin</i>	38	6.01138e-06	iterasi
11	4	10-5-3-2-1	<i>Purelin-purelin-purelin-purelin-purelin</i>	5	0.0477542	gradient

Pada pelatihan menggunakan lebih dari 2 lapisan tersembunyi pelatihan terbaik tercapai pada saat pelatihan dengan 4 HL, yaitu pada pelatihan no 7. Pelatihan menggunakan fungsi aktivasi *sigmoid bipolar-sigmoid biner-sigmoid bipolar-sigmoid biner-identitas*. MSE yang dicapai adalah  $5.80924e-006$  dan pada saat iterasi ke 30.



Gambar 4.7 Hasil pelatihan dengan 4 lapisan tersembunyi fungsi aktivasi *sigmoid biner-sigmoid bipolar-sigmoid biner-sigmoid bipolar-identitas*

Pada pelatihan fungsi aktivasi *sigmoid bipolar-sigmoid bipolar-sigmoid bipolar-identitas* juga mencapai target MSE terbaik pada 3 HI

Tabel 4.4 Hasil pelatihan terbaik dari masing – masing pengelompokan

No.	HL	Neuron	F. aktivasi	Iterasi	MSE	Ket.
1	1	10 – 1	<i>logsig – logsig</i>	794	5.82706	Max mu
2	1	10 – 1	<i>logsig – purelin</i>	13	8.91424e-006	iterasi
3	1	10– 1	<i>logsig – tansig</i>	487	5.9347	Max mu
4	1	10 – 1	<i>purelin – logsig</i>	2	5.93474	gradient
5	1	10 – 1	<i>purelin – purelin</i>	5	0.0477542	<i>gradient</i>
6	1	10 – 1	<i>purelin – tansig</i>	39	5.82801	<i>Max mu</i>
7	1	10– 1	<i>tansig – logsig</i>	794	5.82706	Max mu
8	1	10 – 1	<i>tansig – purelin</i>	29	9.7881e-008	iterasi
9	1	10 – 1	<i>tansig – tansig</i>	593	5.82706	Max mu
10	2	10 -5 - 1	<i>purelin – purelin purelin</i>	5	0.0477542	gradien
11	2	10- 5 - 1	<i>logsig – logsig logsig</i>	2	5.93474	gradient
12	2	10 - 5- 1	<i>purelin – logsig logsig</i>	7	5.93474	gradient

No.	HL	Neuron	F. aktivasi	Iterasi	MSE	Ket.
13	2	10 - 5 - 1	<i>purelin – purelin logsig</i>	2	5.93474	gradient
<b>14</b>	<b>2</b>	<b>10 - 5 - 1</b>	<b><i>Tansig-tansig- purelin</i></b>	<b>5</b>	<b>9.68815/1e-008</b>	<b>iterasi</b>
15	2	10 - 5 - 1	<i>tansig – purelin purelin</i>	12	3.92692e-006	iterasi
16	2	10- 5 - 1	<i>Logsig-tansig- purelin</i>		6.60503e-006	iterasi
17	2	10 - 5 - 1	<i>Logsig-logsig- purelin</i>	7	2.16823e-006	iterasi
18	2	10 - 5-1	<i>Tansig-tansig- purelin</i>	5	9.83325e-006	iterasi
19	3	10-3-2-1	<i>Tansig-tansig- tansig-purelin</i>	126	9.9617e-008	iterasi
20	3	10-3-2-1	<i>Logsig-logsig- logsig-purelin</i>	8	2.49408	gradient
21	3	10-3-2-1	<i>Logsig-logsig- logsig-logsig</i>	2	5.93474	gradient
22	3	10-3-2-1	<i>Tansig-tansig- tansig-tansig</i>	2	5.93474	gradient
23	3	10-3-2-1	<i>Purelin-purelin- purelin-purelin</i>	7	0.0477542	gradient
24	3	10-3-2-1	<i>Logsig-tansig- logsig-tansig</i>	419	5.82706	Max mu
25	4	10-5-3-2-1	<i>Tansig-logsig- tansig-logsig- purelin</i>	30	5.80924e-006	iterasi
26	4	10-5-3-2-1	<i>Logsig-logsig- logsig-logsig- logsig</i>	7	5.93474	gradient

No.	HL	Neuron	F. aktivasi	Iterasi	MSE	Ket.
27	4	10-5-3-2-1	<i>Tansig-tansig-tansig-tansig-tansig</i>	1	5.93474	gradient
28	4	10-5-3-2-1	<i>Logsig-logsig-logsig-logsig-purelin</i>	38	6.01138e-06	iterasi

Dari tabel 4.4 diatas, hasil pelatihan terbaik adalah pelatihan pada tabel nomor 14. Pelatihan tersebut berhenti ketika iterasi yang ditentukan sudah tercapai dengan MSE 9.68815/1e-008. Sehingga untuk pengujian jaringan akan digunakan struktur yang sama dengan hasil pelatihan pada tabel 14 tersebut.

Pada pelatihan yang berhenti karena *gradient* sudah mencapai target, artinya MSE yang dihasilkan sudah mencapai nilai yang paling minimum untuk arsitektur jaringan syaraf. 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. Pada pelatihan yang berhenti karena *maximum mu* sudah mencapai target, artinya nilai sudah mencapai nilai yang paling maximum untuk arsitektur jaringan syaraf. Nilai *maximum mu* menggunakan nilai *default* yang di tentukan fungsi *newff* yaitu 1e10. Jika dilihat dari tabel hasil pelatihan fungsi aktivasi identitas dapat menghasilkan MSE yang yang mencapai target, hal ini disebabkan karena target dari jaringan syaraf memiliki nilai yang sebanding, hal ini sesuai dengan fungsi aktivasi identitas dimana fungsi aktivasi sama dengan keluarannya. Jumlah lapisan dan sel neuron pada masing-masing lapisan tersembunyi tidak berpengaruh besar terhadap MSE. Tetapi semua pelatihan yang menggunakan fungsi aktivasi sigmoid *bipolar* dan

sigmoid *biner* pada lapisan keluaran tidak dapat mencapai target iterasi dan MSE, disebabkan karena fungsi aktivasi *sigmoid* bipolar memiliki nilai *range* antara -1 sampai 1 fungsi aktivasi biner memiliki *range* 0 sampai 1, sehingga tidak sebanding dengan data pelatihan. Sedangkan pada pelatihan, nilai target belum di normalisasi sehingga target tidak sesuai dengan range fungsi aktivasi biner dan fungsi aktivasi bipolar. Begitu juga jika fungsi identitas jika digunakan sebagai fungsi masukan maka pelatihan akan mencapai nilai gradient.

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

Dari tabel pelatihan 4.4, maka struktur jaringan yang akan digunakan adalah struktur jaringan pada tabel nomor 14. Dimana struktur jaringan terdiri dari 1 sel neuron *input*. yaitu selisih antara *set point* dan harga *Output* pada saat *t*. Lapisan *input* (*v*) terdiri dari 10 sel neuron, sedangkan lapisan tersembunyi terdiri dari 2 lapisan. Lapisan tersembunyi pertama (*w*) terdiri dari 5 sel neuron dan sesuai dengan target sistem jaringan syaraf, maka lapisan tersembunyi kedua atau lapisan output (*w\_out*) terdiri dari 1 sel neuron. Fungsi aktivasi yang digunakan pada

lapisan masukan dan lapisan tersembunyi adalah fungsi *sigmoid bipolar*, sedangkan pada lapisan keluaran menggunakan fungsi aktivasi *identitas*.

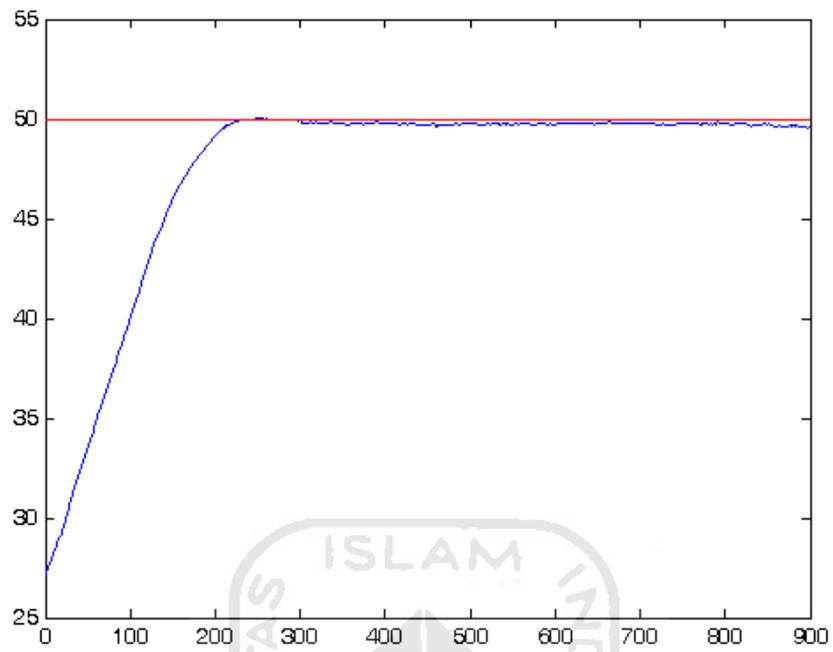
#### **4.2 Pengujian Jaringan Saraf Tiruan *Backpropagation***

Pengujian ini menggabungkan perangkat keras (*hardware*) dan perangkat lunak (*software*) yang sudah dirancang. Agar pengujian yang dilakukan valid dan terstruktur, maka dilakukan pengujian beberapa parameter sebagai berikut:

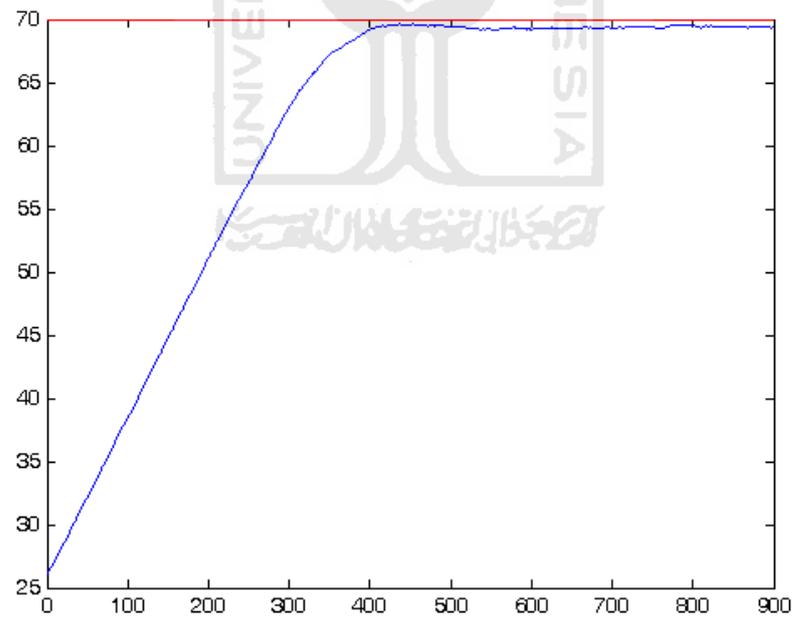
1. Pengujian sistem dengan variasi *set point*
2. Pengujian sistem pada suhu referensi naik.
3. pengujian sistem dengan pemberian gangguan

##### **4.2.1 Pengujian Sistem Dengan Variasi *Set point***

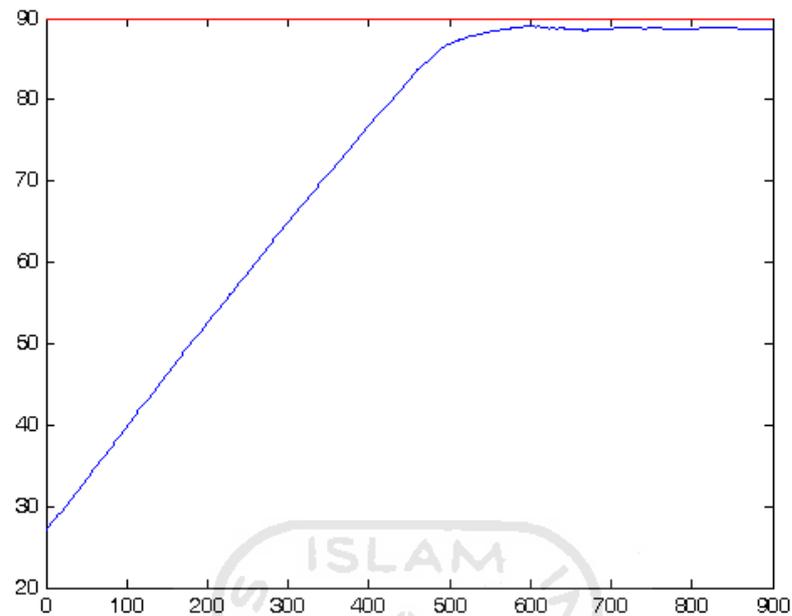
Pengujian dilakukan dengan cara memberikan nilai referensi tertentu dan mendapatkan grafik responnya. Pengujian dilakukan dengan memberikan suhu referensi sebesar 50 °C, 70 °C dan 90 °C untuk masing-masing kombinasi Respon diperlihatkan pada gambar 4.8.



Gambar 4.8 Respon sistem pada set point 50 °C



Gambar 4.9 Respon sistem pada set point 70 °C



Gambar 4.10 Respon sistem pada set point 90 °C

Tabel 4.5 Karakteristik respon sistem

Suhu Referensi	Parameter	
50°C	Rise Time (s)	152.5005
	Settling Time (s)	209.6798
	Settling Min (°C)	47.7646
	Settling Max (°C)	50.0175
	Overshoot (%)	0.0350
	Undershoot (%)	0
	Peak (°C)	50.0175
	Peak Time (s)	252
70°C	Rise Time (s)	291.1207
	Settling Time (s)	558.3409
	Settling Min (°C)	65.7025
	Settling Max (°C)	69.6357
	Overshoot (%)	0
	Undershoot (%)	0
	Peak (°C)	69.6357
	Peak Time (s)	437

Suhu Referensi	Parameter	
90°C	Rise Time (s)	416.0944
	Settling Time (s)	NaN
	Settling Min (°C)	83.6706
	Settling Max (°C)	89.7913
	Overshoot (%)	0
	Undershoot (%)	0
	Peak (°C)	89.7913
	Peak Time (s)	596

Pada saat *setting value* 50°C waktu naik (*rise time*) yang dicapai adalah 152 detik, waktu penetapan (*settling time*) adalah 209 detik, lonjakan maksimum (*overshoot*) memiliki nilai yang kecil, yaitu 0.035 % dan waktu puncak (*peak time*) terjadi pada detik ke 252. Dengan demikian pada saat *setting value* 50°C sistem dapat bekerja dengan cukup baik.

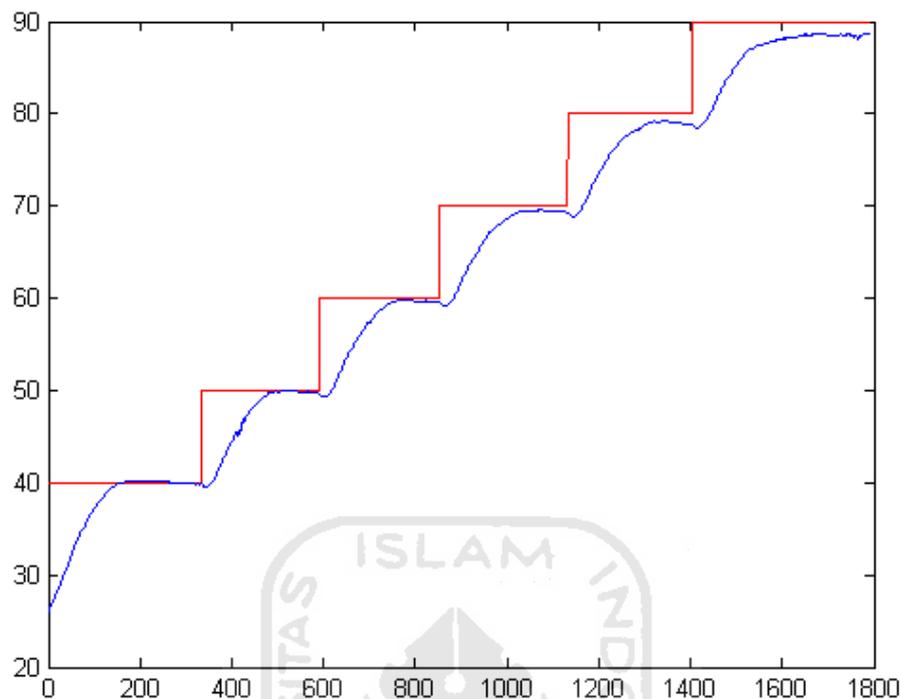
Pada saat *setting value* 70 °C, waktu naik (*rise time*) yang dicapai adalah 291 detik, waktu penetapan (*settling time*) adalah 558 detik, dari waktu *settling time* yang dihasilkan sistem membutuhkan waktu yang cukup lama untuk bisa stabil jika dilihat dari waktu naik. Lonjakan maksimum (*overshoot*) tidak terjadi, tetapi jika dilihat pada gambar respon yang dihasilkan sistem tidak dapat mencapai suhu referensi. Hal tersebut terjadi karena pada saat suhu *maximum* tercapai sebesar 69.63°C tegangan yang diberikan ke sistem dari hasil pelatihan JST tidak dapat menaikkan suhu lagi. Waktu puncak (*peak time*) terjadi pada detik ke 437. Dari step respon yang dihasilkan pada saat *setting value* 70°C sistem masih dapat bekerja dengan cukup baik walaupun terjadi selisih dengan suhu target, tetapi masih cukup kecil.

Pada saat *setting value* 90 °C, waktu naik (*rise time*) yang dicapai adalah 416 detik, waktu penetapan (*settling time*) tidak terdeteksi (NaN) dan

lonjakan maksimum (*overshoot*) tidak terjadi bahkan terjadi kesalahan keadaan tunak atau *offset*. Saat suhu mendekati *setting value*, pemanas sudah tidak dapat menaikkan suhu lagi dikarenakan tegangan yang diberikan ke pemanas semakin mendekati *setting value* semakin kecil. Semakin besar *setting value* yang diberikan *Offset* yang terjadi akan semakin besar dikarenakan tegangan yang dibutuhkan semakin besar atau tidak sama besarnya pada saat *setting value* tertentu untuk mencapai keadaan tunaknya. Waktu puncak (*peak time*) terjadi pada detik ke 596.

#### 4.2.2 Pengujian Sistem Pada Suhu Referensi Naik

Pengujian pada suhu referensi naik ini bertujuan untuk mengetahui kecepatan respon sistem pengendali Jaringan Syaraf Tiruan (JST) terhadap perubahan kenaikan suhu referensi. Hal ini penting untuk mengetahui apakah JST yang dirancang dapat bekerja baik pada daerah operasi yang lebih luas. Pengujian dilakukan dengan cara memberikan variasi suhu masukan (referensi) sebesar 40 °C, 50 °C, 60°C, 70 °C, 80 °C dan 90 °C. Perubahan suhu referensi dilakukan ketika suhu *plant* berada pada keadaan stabil pada tiap suhu referensinya. Berikut ditampilkan data grafik hasil pengujiannya.



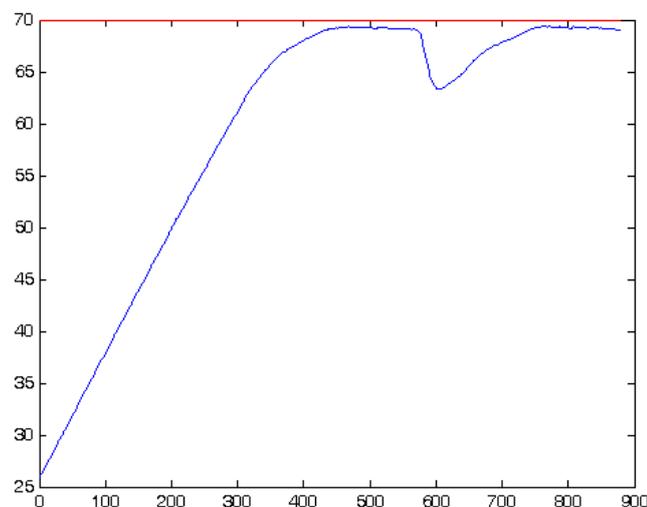
Gambar 4.11 Grafik respon sistem pada suhu referensi naik

Suhu awal *plant* adalah 27 °C, implementasinya dapat dilihat pada gambar 4.11. Mula-mula suhu referensi sebesar 40 °C, setelah 129 detik suhu *plant* mencapai 40 °C dan terus berosilasi di sekitar nilai referensinya. Perubahan nilai suhu referensi dilakukan tiap 200 detik yaitu saat suhu *plant* berada pada keadaan tunak atau stabil. Berdasarkan grafik pada gambar 4.11 terlihat suhu air pada *plant* dapat merespon perubahan referensi yang diberikan dengan bergerak mengikuti perubahan suhu referensinya. Akan tetapi tiap kenaikan suhu referensi terjadi nilai *error* yang terjadi semakin besar. Hal tersebut terjadi karena data pelatihan untuk JST tidak bisa menyesuaikan terhadap *setting value* yang diberikan. Sehingga saat suhu referensi semakin tinggi *offset* yang terjadi semakin besar. Dari pengujian ini

dapat disimpulkan JST yang dirancang mempunyai respon baik pada daerah  $40^{\circ}\text{C}$  -  $90^{\circ}\text{C}$  saja.

#### 4.2.3 Pengujian Sistem Dengan Pemberian Gangguan

Untuk mengetahui daya tahan sistem terhadap gangguan, maka dilakukan pengujian dengan memberikan gangguan berupa penambahan es batu pada sistem yang telah mencapai kestabilan untuk suatu *setting value*. Pada pengujian ini gangguan yang diberikan dengan cara memasukkan es batu ke dalam *plant* suhu yang telah mencapai suhu  $70^{\circ}\text{C}$ . Pengujian dengan gangguan panas tidak diberikan karena tugas akhir ini tidak dibuat untuk melakukan pendinginan. Tujuan pengujian dengan gangguan ini adalah mengetahui kecepatan respon sistem untuk kembali ke referensi semula setelah diberikan gangguan, waktu yang diperlukan untuk kembali ke referensi semula disebut sebagai waktu pemulihan. Grafik respon pengujian dengan pemberian gangguan diperlihatkan pada gambar 4.12.



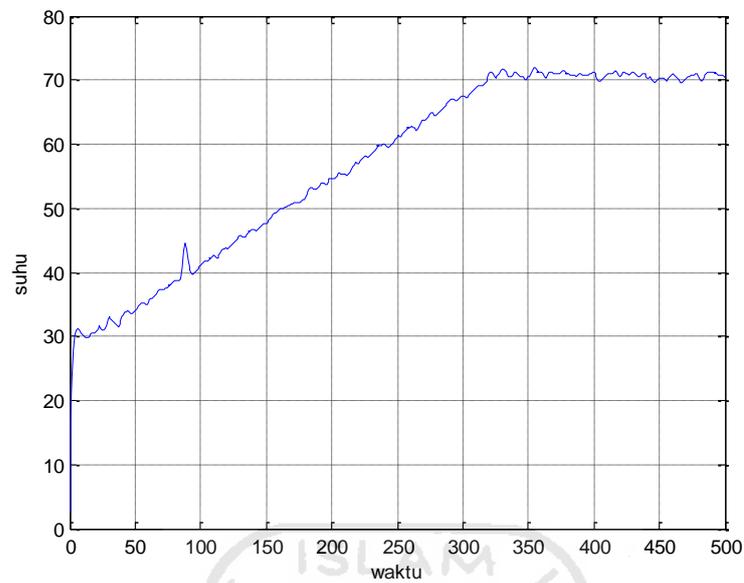
Gambar 4.12 Grafik respon sistem dengan gangguan

Pada pengujian ini mula-mula ditetapkan suhu referensi sebesar 70 °C. Nilai suhu awal adalah 27 °C. Setelah 441 detik suhu air akan mencapai 70 °C dan beresilasi pada keadaan tersebut. Setelah 566 detik dari suhu awal dan ketika suhu sudah stabil, es batu dimasukkan dalam *plant*, selanjutnya diketahui suhu air dalam *plant* turun sampai 63.2855°C pada detik ke-604, kemudian berangsur-angsur kembali ke referensi semula pada detik 745 dan kemudian beresilasi di sekitar suhu referensinya.

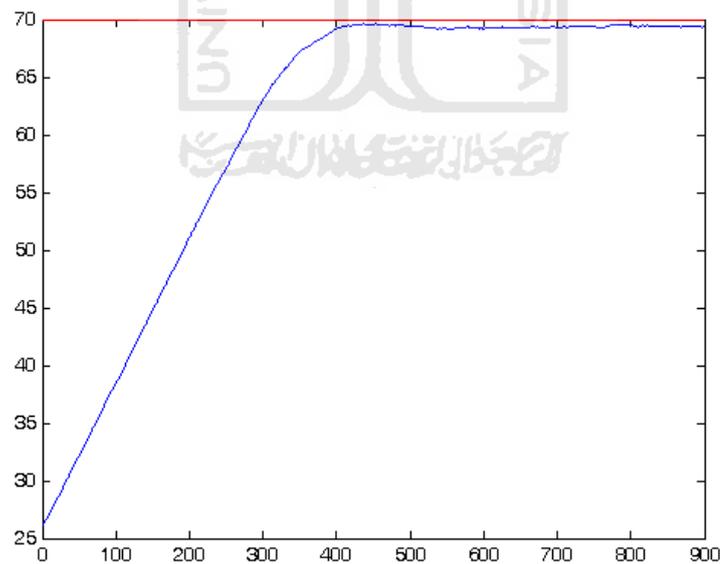
Dari pengujian dengan gangguan yang telah dilakukan dapat disimpulkan bahwa sistem pengendali JST yang dibangun dapat bertahan dan mampu mengeliminir gangguan-gangguan dari luar dengan cukup baik.

#### **4.2.4 Perbandingan Pengendali Jaringan Syaraf dengan PID Digital**

Pada perbandingan pengendali Jaringan Syaraf dengan PID Digital bertujuan untuk membandingkan respon sistem yang dihasilkan pada masing-masing pengendali. Respon PID yang digunakan adalah dari penelitian yang dilakukan oleh Roni Andika yang berjudul “*Perancangan Pengendali PID Digital pada Heater*”. Pada penelitian ini respon terbaik dihasilkan pada nilai  $K_p=1$ ,  $K_i=2$ , dan  $K_d=5$ . Respon sistem dapat dilihat pada gambar 4.13 berikut.



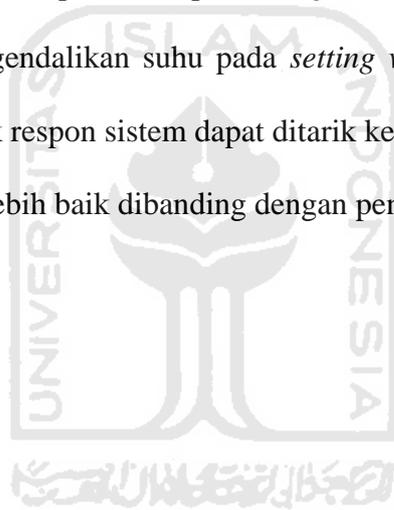
Gambar 4.13 Grafik respon sistem pengendali PID dengan nilai  $K_p=1$ ,  $K_i=2$ , dan  $K_d=5$  dan *setting value*  $70^{\circ}\text{C}$



Gambar 4.14 Respon sistem pengendali Jaringan Syaraf pada *setting value*  $70^{\circ}\text{C}$

Dari grafik respon sistem pengendali PID gambar 4.12, nilai suhu awal adalah berkisar pada suhu  $30^{\circ}\text{C}$ . Pengendali PID membutuhkan waktu 305 detik

untuk mencapai suhu  $70^{\circ}\text{C}$ . Nilai maksimum *overshootnya* kurang lebih  $2^{\circ}\text{C}$  sedangkan pada pengendali jaringan syaraf membutuhkan waktu sebesar 390 detik untuk mencapai *setting value* dan tidak mempunyai nilai *overshoot* dikarenakan terjadi kesalahan keadaan tunak atau *offset*. Respon grafik pengendali PID mempunyai waktu lebih cepat dari pengendali jaringan syaraf pada *setting value* yang sama. Respon sistem pengendali PID juga mampu mencapai *setting value* yang diberikan, sedang respon sistem pengendali jaringan syaraf mengalami *offset* atau tidak dapat mencapai *setting value* yang diberikan. Kedua respon sistem dapat mengendalikan suhu pada *setting value*  $70^{\circ}\text{C}$ . Dari analisis perbandingan kedua grafik respon sistem dapat ditarik kesimpulan pengendali PID mempunyai respon yang lebih baik dibanding dengan pengendali Jaringan syaraf.



## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Setelah pelatihan dan pengujian dilakukan, maka dapat disimpulkan bahwa :

1. Pelatihan dan pengujian akan lebih cepat jika data yang dihitung dalam jaringan menggunakan operasi matriks.
2. 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.
3. Struktur terbaik jaringan syaraf tiruan untuk kendali suhu air dari 10 sel neuron lapisan input. Lapisan tersembunyi terdiri dari 5 lapisan, lapisan tersembunyi kedua terdiri dari 1 sel neuron . MSE yang dihasilkan 9.68815/1e-008 dengan fungsi aktivasi sigmoid bipolar-sigmoid bipolar-identitas
4. Berdasarkan pengujian pada suhu naik. Pengendali dengan jaringan syaraf tiruan yang dirancang mempunyai respon yang cukup cepat dan mampu beroperasi pada 30 °C - 90°C dengan volume air 1000ml.
5. Pengendali jaringan syaraf tiruan yang dibuat mampu beradaptasi terhadap gangguan dari luar.
6. Perubahan volume dan daya pemanas akan berpengaruh terhadap sistem.

## 5.2 Saran

1. Jaringan dilakukan dengan metode yang berbeda, agar menghasilkan nilai *Mean Square Error* yang lebih kecil lagi.
2. Data pelatihan dibuat lebih baik agar sistem dapat bekerja maksimal dan pada cakupan yang lebih luas.
3. Agar kenaikan suhu semakin cepat maka perlu memperbesar daya pemanas atau menambah jumlah pemanas yang digunakan.



## DAFTAR PUSTAKA

Andika, Roni, 2010. *Perancangan Pengendalian PID Digital Pada Heater*. Skripsi, tidak diterbitkan. Yogyakarta : Fakultas Teknologi Industri Universitas Islam Indonesia.

Isnanto, 2011. *Aplikasi Logika Fuzzy Pada Pengendalian Suhu Water Heater Berbasis LabView*. Skripsi, tidak diterbitkan. Yogyakarta : Fakultas Teknologi Industri Universitas Islam Indonesia.

Priawan, Rizki, 2010. *Akuisisi Data dan Identifikasi Sistem Pemanas Air*. Skripsi, tidak diterbitkan. Yogyakarta : Fakultas Teknologi Industri Universitas Islam Indonesia.

Wiryadinata, Romi. 2005. *Simulasi jaringan syaraf tiruan berbasis metode backpropagation sebagai pengendali kecepatan motor DC*. Skripsi, tidak diterbitkan. Yogyakarta : Fakultas Teknologi Industri Universitas Islam Indonesia.

Siang, Jong Jek, 2005. *Jaringan Syaraf Tiruan dan Pemrograman Menggunakan Matlab*. Yogyakarta: Penerbit Andi.

Halvorsen, Hans-Petter. *DAQ in Matlab*. Tutorial. Finlandia:Telemark University College Department of Electrical Engineering, Information Technology and Cybernetics

KUSUMADEWI, Sri. *Sistem Cerdas : Panduan praktikum mandiri Modul II ,A Jaringa Syaraf Tiruan*. Modul Praktikum, tidak diterbitkan Yogyakarta : Fakultas Teknologi Industri Universitas Islam Indonesia.

National Instruments, 2006, “*DAQ – USB 6009 manual*”.<http://public.beuth-hochschule.de/~suchanek/EMSL/usb6009.pdf>

Wiryadinata, Romi & Ana Ratnawati, D. *Simulasi Jaringan Syaraf Tiruan Berbasis Metode Backpropagation Sebagai Pengendali Kecepatan Motor*, SNATI 2005.

