

37.609/H/10



ITS

Institut
Teknologi
Sepuluh Nopember

RSPe
620.112 23
Pur
p-1
2009

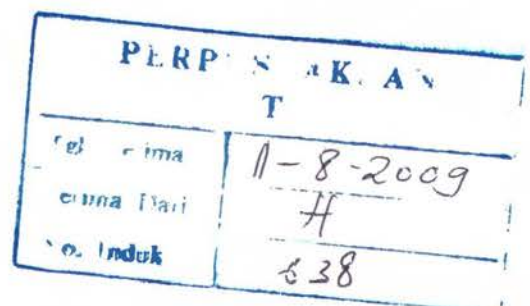
TUGAS AKHIR - LK 1347

PENGUNAAN JARINGAN SARAF TIRUAN UNTUK MENGETAHUI PENGARUH VARIASI TEMPERATUR PEMANASAN PADA PROSES PERLAKUAN PANAS PASCA PENGELASAN BAJA AISI 316 TERHADAP LAJU KOROSI

DIKA PURNAMA
N.R.P. 4104 100 066

Dosen Pembimbing
Ir. Achmad Zubaydi, M. Eng., Ph. D
Wing Hendroprasetyo A. P., S.T., M.Eng

JURUSAN TEKNIK PERKAPALAN
Fakultas Teknologi Kelautan
Institut Teknologi Sepuluh Nopember
Surabaya 2009



LEMBAR PENGESAHAN

PENGGUNAAN JARINGAN SARAF TIRUAN UNTUK MENGETAHUI PENGARUH VARIASI TEMPERATUR PEMANASAN PADA PROSES PERLAKUAN PANAS PASCA PENGELASAN BAJA AISI 316 TERHADAP LAJU KOROSI

TUGAS AKHIR

**Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Bidang Studi Konstruksi dan Kekuatan
Jurusan Teknik Perkapalan
Fakultas Teknologi Kelautan
Institut Teknologi Sepuluh Nopember**

Oleh:

**DIKA PURNAMA
N.R.P. 4104 100 066**

Disetujui oleh Dosen Pembimbing Tugas Akhir :

- 1. Ir. Achmad Zubaydi, M.Eng., Ph. D**
- 2. Wing Hendroprasetyo A, P., S.T., M..Eng**



SURABAYA, JUNI 2009

LEMBAR REVISI

PENGUNAAN JARINGAN SARAF TIRUAN UNTUK MENGETAHUI PENGARUH VARIASI TEMPERATUR PEMANASAN PADA PROSES PERLAKUAN PANAS PASCA PENGELASAN BAJA AISI 316 TERHADAP LAJU KOROSI

TUGAS AKHIR


Telah Direvisi Sesuai Dengan Hasil Sidang
Ujian Tugas Akhir
pada
Bidang Studi Konstruksi dan Kekuatan
Jurusan Teknik Perkapalan
Fakultas Teknologi Kelautan
Institut Teknologi Sepuluh Nopember

Oleh:

DIKA PURNAMA
N.R.P. 4104 100 066

Dosen Penguji


Ir. I G M Santosa


Ir. Asjhar Imron, M. Sc, M.SE, PED

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Ir. Achmad Zubaydi, M.Eng., Ph. D
2. Wing Hendropresetyo A. P., S. T., M.Eng



SURABAYA,

JULI 2009

KATA PENGANTAR

Puji syukur penulis haturkan kehadiran Allah SWT yang telah melimpahkan segala rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul “PENGUNAAN JARINGAN SARAF TIRUAN UNTUK MENGETAHUI PENGARUH VARIASI TEMPERATUR PEMANASAN PADA PROSES PERLAKUAN PANAS PASCA PENGELASAN BAJA AISI 316 TERHADAP LAJU KOROSI” dengan sebaik-baiknya. Dan tak lupa shalawat serta salam tetap tercurahkan kepada nabi Muhammad SAW, yang senantiasa memberi petunjuk arah jalan kebenaran dan kebaikan.

Laporan Tugas Akhir ini disusun untuk memenuhi persyaratan dalam menyelesaikan studi strata satu Jurusan Teknik Perkapalan Fakultas Teknologi Kelautan, Institut Teknologi Sepuluh Nopember Surabaya.

Selama proses penyusunan Laporan Tugas Akhir ini penulis telah menerima banyak bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada :

1. Bapak Ir. Achmad Zubaydi, M. Eng., Ph. D. dan Bapak Wing Hendropasetyo A.P, ST, M.Eng. Selaku dosen pembimbing, atas segala bimbingan, ilmu, waktu, dan kesabaran dalam mengarahkan dan memberi nasehat kepada penulis dalam menyelesaikan Tugas Akhir.
2. Bapak Ir. Soewefy, M.Eng. Selaku koordinator bidang konstruksi dan kekuatan.
3. Bapak Ir. Triwilaswandio, MSc. Selaku Ketua Jurusan Teknik Perkapalan – FTK ITS
4. Ibu Ir. Hesty Anita Kurniawati, M.Sc. Selaku dosen wali selama penulis menjadi mahasiswa di Jurusan Teknik Perkapalan – FTK ITS.
5. Ibunda tercinta yang telah memberikan kasih sayang yang tiada tara dan limpahan doa yang selalu menyelimuti penulis. Ayah yang selalu menjadi panutan penulis dalam bertindak, yang mengajarkan kepada penulis akan pentingnya arti sebuah kata tanggung jawab.
6. Kakakku tercinta Titik Ekowati beserta suami Mas Fuat Ratsongko, yang selalu memberi dorongan semangat dan kasih sayang serta perhatian kepada penulis. Serta kedua keponakan penulis, Wisnu dan Galih semoga kelak menjadi anak yang berbakti kepada kedua orang tua dan menjadi orang yang berguna bagi agama, keluarga, nusa dan bangsa.
7. Untuk seseorang yang selalu hadir dalam lamunan penulis “Winda Sukma Rahayu”, cinta dan kasih sayangmu yang membuat hidup ini menjadi berwarna dan lebih bermakna.
8. Seluruh Saudara penulis yang selalu menanyakan kapan penulis akan lulus. Pertanyaan itu selalu mendorong penulis untuk segera menyelesaikan masa perkuliahan ini.
9. Tri Afirianto, S. T., sahabat penulis sejak SMA yang telah banyak memberikan ilmunya tentang Jaringan Saraf Tiruan dan selalu memberi dukungan kepada penulis.
10. Teman-teman di Teknik Perkapalan, P44 - Bokeppers, penghuni Castil Dracula, penghuni Puri, serta sahabat seperjuangan TA, Febriandika Sabda Rifa, tetap semangat ya!.

11. Teman-teman kos “super-kenchot”, Ucup, bayor, Timbul, Akub, Mas Rovi, Indra, Cipox, Birul, dan Haryo, yang selalu membuat suasana menjadi lebih santai ketika pikiran mulai lelah dalam mengerjakan Tugas Akhir ini.
12. Ibu kos dan Mbok-e, yang sudah menyediakan tempat tinggal selama penulis menempuh pendidikan di kampus perjuangan ITS ini.
13. Semua pihak yang telah membantu penulis, yang tidak mungkin disebutkan satu persatu.

Penulis sadar bahwa penulisan Tugas Akhir ini masih jauh dari kesempurnaan, karena kesempurnaan hanyalah milik Allah. Maka saran dan kritik yang bersifat membangun sangat diharapkan. Akhir kata semoga tulisan ini dapat bermanfaat bagi banyak pihak.

Surabaya, Juni 2009

Penulis

ABSTRAK

Baja tahan karat AISI 316 yang telah dikenai proses pengelasan, struktur mikronya akan berubah dan cenderung menyebabkan korosi. Dari penelitian sebelumnya diketahui pengaruh perlakuan panas pasca pengelasan terhadap timbulnya korosi. Variasi suhu perlakuan panas adalah 850°C, 900°C, 950°C, dan 1000°C, sedangkan suhu yang mempengaruhi perubahan struktur mikro baja tahan karat AISI 316 adalah sampai 1120°C.

Pada penelitian ini akan dikembangkan perangkat lunak berbasis jaringan saraf tiruan (JST) dengan arsitektur *backpropagation* untuk memprediksi laju korosi yang terjadi jika suhu pemanasan pasca pengelasan dilakukan sampai 1120°C. Data-data untuk pelatihan jaringan didapatkan dari data pengujian terdahulu yang terlebih dahulu diperluas dengan menggunakan metode regresi polynomial dan interpolasi.

Berdasarkan hasil perhitungan dengan menggunakan jaringan saraf tiruan diperoleh nilai laju korosi untuk temperatur pemanasan 850°C, 900°C, 950°C, 1000°C, 1050°C, 1100°C, dan 1120°C secara berturut-turut sebesar : 0.345 mmpy, 0.332 mmpy, 0.327 mmpy, 0.225 mmpy, 0.187 mmpy 0.189 mmpy dan 0.175 mmpy. Dengan melihat besarnya laju korosi yang dihasilkan oleh masing-masing suhu pemanasan dapat disimpulkan bahwa laju korosi menjadi berkurang ketika suhu pemanasan bertambah besar.

Kata kunci : baja tahan karat AISI 316, *backpropagation*, laju korosi, jaringan saraf tiruan

ABSTRACT

After welding process, the microstructure of stainless steel AISI 316 will be altered, which may cause corrosion problem. From the previous research, effect of post weld heat treatment to corrosion problem has been known. Temperature variation for post weld heat treatment are 850°C, 900°C, 950°C, and 1000°C.

This research will develop a software, based on artificial neural network (ANN) with backpropagation algorithm to predict corrosion rate which will arise if the heating temperature is increased until 1120°C. This ANN models will then be taught and tested by using the data that obtained from polynomial and interpolation regressions.

The result indicated that for heating temperature of 850°C, 900°C, 950°C, 1000°C, 1050°C, 1100°C, and 1120°C the corrosion rates are 0.345 mmpy, 0.332 mmpy, 0.327 mmpy, 0.225 mmpy, 0.187 mmpy, 0.189 mmpy and 0.175 mmpy respectively. Based on the test result, the higher temperatures of heat treatment produce the lower corrosion rates.

Key words : stainless steel AISI 316, backpropagation, corrosion rates, artificial neural network

DAFTAR ISI

LEMBAR PENGESAHAN.....	ii
LEMBAR REVISI	iii
KATA PENGANTAR.....	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
Bab 1. Pendahuluan	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah.....	2
1.3. Tujuan.....	2
1.4. Manfaat.....	2
1.5. Batasan Masalah.....	2
1.6. Hipotesis Awal	2
1.7. Metodologi	3
1.8. Sistematika Penulisan.....	5
Bab 2. Tinjauan Pustaka	7
2.1. Jaringan Saraf Tiruan (<i>Artificial Neural Network</i>).....	7
2.1.1. Mengenal Jaringan Saraf Tiruan	7
2.1.2. Sistem Jaringan Secara Biologis	7
2.1.3. Struktur Jaringan Saraf Tiruan	8
2.1.4. Pembelajaran dalam Jaringan Saraf Tiruan.....	9
2.1.5. Pola Koneksi Jaringan Saraf Tiruan.....	12
2.1.6. Struktur <i>feedforward</i>	12
2.2. Backpropagation Neural Network.....	13
2.2.1. Metode <i>Backpropagation</i>	13
2.2.2. Fungsi Aktivasi.....	14
2.2.3. Learning Rate dan Momentum.....	15
2.2.4. Algoritma <i>Backpropagation</i>	16
2.2.5. Bias.....	19
2.2.6. Inisialisasi Jaringan	20
2.3. Korosi	21
2.3.1. Jenis-jenis Korosi	21
2.4. Baja Tahan Karat.....	24
2.4.1. Klasifikasi Baja Tahan Karat	24
2.4.2. Pengelasan Baja Tahan Karat.....	26
2.4.3. Perlakuan Panas Baja Tahan Karat Austenitik.....	27
2.4.4. Presipitasi Karbida Krom	28
2.5. Metode Numerik Penambahan Data.....	28
2.5.1. Regresi Polynomial	29
2.5.2. Interpolasi.....	30

Bab 3.	Metodologi Penelitian	31
3.1.	Sumber Data dan Variabel	31
3.2.	Metode Penelitian.....	31
3.2.1.	Identifikasi Permasalahan.....	31
3.2.2.	Penentuan Tujuan Penelitian.....	31
3.2.3.	Studi literatur.....	32
3.2.4.	Identifikasi Input dan Output Jaringan.....	32
3.2.5.	Pengumpulan Data	32
3.2.6.	Penentuan Vektor Input Jaringan	33
3.2.7.	Pendefinisian Output Jaringan.....	34
3.2.8.	Pendesainan Jaringan Saraf Tiruan	34
3.3.	Diagram Alir Prosedur Jaringan saraf Tiruan	36
Bab 4.	Analisa dan Pembahasan.....	37
4.1.	Analisa Pemilihan Data Pelatihan	37
4.1.1.	Regresi Polynomial	37
4.1.2.	Interpolasi.....	39
4.2.	Analisa Pengaruh Perubahan Laju Pelatihan (α) terhadap Perilaku Jaringan	41
4.2.1.	Pelatihan jaringan menggunakan laju pelatihan 0,1 dan momentum 0,1.....	41
4.2.2.	Pelatihan jaringan menggunakan laju pelatihan 0,2 dan momentum 0,1.....	42
4.2.3.	Pelatihan jaringan menggunakan laju pelatihan 0,3 dan momentum 0,1.....	43
4.2.4.	Pelatihan jaringan menggunakan laju pelatihan 0,4 dan momentum 0,1.....	44
4.2.5.	Pelatihan jaringan menggunakan laju pelatihan 0,5 dan momentum 0,1.....	45
4.3.	Analisa Pengaruh Perubahan Monentum (η) terhadap Perilaku Jaringan.....	47
4.3.1.	Pelatihan jaringan menggunakan laju pelatihan 0,1 dan momentum 0,1.....	48
4.3.2.	Pelatihan jaringan menggunakan laju pelatihan 0,1 dan momentum 0,2.....	49
4.3.3.	Pelatihan jaringan menggunakan laju pelatihan 0,1 dan momentum 0,3.....	50
4.3.4.	Pelatihan jaringan menggunakan laju pelatihan 0,1 dan momentum 0,4.....	51
4.3.5.	Pelatihan jaringan menggunakan laju pelatihan 0,1 dan momentum 0,5.....	52
4.4.	Analisa Laju Korosi dengan Jaringan Saraf Tiruan	54
Bab 5.	Kesimpulan dan Saran.....	57
5.1.	Kesimpulan.....	57
5.2.	Saran.....	58
DAFTAR PUSTAKA.....		59

DAFTAR GAMBAR

Gambar 1.1.	Flow chart metodologi.....	5
Gambar 2.1.	Diagram sederhana dari neuron.....	8
Gambar 2.2.	Struktur artificial neural networks.....	9
Gambar 2.3.	Jaringan saraf tiruan <i>feedforward</i>	12
Gambar 2.4.	Arsitektur <i>backpropagation</i>	13
Gambar 2.5.	Fungsi aktivasi sigmoid-biner.....	14
Gambar 2.6.	Neuron dengan sebuah bias.....	20
Gambar 2.7.	Grafik heat treatment.....	27
Gambar 2.8.	Pola grafik yang dibentuk oleh data sekunder.....	28
Gambar 3.1.	Jaringan saraf tiruan yang direncanakan.....	33
Gambar 3.2.	Start up program delphi 7.....	34
Gambar 3.3.	Prosedur pelatihan jaringan saraf tiruan.....	36
Gambar 4.1.	Grafik hubungan antara temperatur pemanasan dan laju korosi dengan regresi polynomial orde 2, orde 3 dan orde 4.....	38
Gambar 4.2.	Grafik hubungan antara temperatur pemanasan dan laju korosi dengan regresi polynomial dan interpolasi.....	40
Gambar 4.3.	Grafik hubungan antara data asli dan data hasil perhitungan jaringan dengan laju pelatihan 0,1.....	42
Gambar 4.4.	Grafik hubungan antara data asli dan data hasil perhitungan jaringan dengan laju pelatihan 0,2.....	43
Gambar 4.5.	Grafik hubungan antara data asli dan data hasil perhitungan jaringan dengan laju pelatihan 0,3.....	44
Gambar 4.6.	Grafik hubungan antara data asli dan data hasil perhitungan jaringan dengan laju pelatihan 0,4.....	45
Gambar 4.7.	Grafik hubungan antara data asli dan data hasil perhitungan jaringan dengan laju pelatihan 0,5.....	46
Gambar 4.8.	Grafik pengaruh variasi learning rate terhadap perilaku masing - masing model jaringan saraf tiruan.....	46
Gambar 4.9.	Grafik hubungan antara data asli dan data hasil perhitungan jaringan dengan momentum 0,1.....	48
Gambar 4.10.	Grafik hubungan antara data asli dan data hasil perhitungan jaringan dengan momentum 0,2.....	49
Gambar 4.11.	Grafik hubungan antara data asli dan data hasil perhitungan jaringan dengan momentum 0,3.....	50
Gambar 4.12.	Grafik hubungan antara data asli dan data hasil perhitungan jaringan dengan momentum 0,4.....	51
Gambar 4.13.	Grafik hubungan antara data asli dan data hasil perhitungan jaringan dengan momentum 0,5.....	52
Gambar 4.14.	Grafik pengaruh variasi momentum terhadap perilaku jaringan saraf tiruan.....	53
Gambar 4.15.	Grafik hubungan antara temperatur pemanasan dan laju korosi yang dihitung dengan menggunakan JST.....	54

DAFTAR TABEL

Tabel 2.1.	Hubungan antara biologi dan jaringan saraf tiruan.....	8
Tabel 2.2.	Paradigma, aturan pembelajaran, arsitektur, algoritma dan aplikasi jaringan saraf tiruan yang terawasi.....	10
Tabel 2.3.	Paradigma, aturan pembelajaran, arsitektur, algoritma dan aplikasi jaringan saraf tiruan yang tidak terawasi.....	11
Tabel 2.4.	Klasifikasi ketahanan korosi berdasarkan nilai laju korosi.....	24
Tabel 2.5.	AISI Martensitic Steel.....	24
Tabel 2.6.	AISI Ferritic steel.....	25
Tabel 2.7.	AISI Austenitic steel.....	26
Tabel 3.1.	Data sekunder.....	31
Tabel 3.2.	Data in-sample.....	32
Tabel 3.3.	Data out-sample.....	33
Tabel 4.1.	Laju korosi dari perhitungan dengan regresi polynomial.....	38
Tabel 4.2.	Laju korosi yang didapatkan dari perhitungan interpolasi.....	39
Tabel 4.3.	Perbandingan laju korosi dari perhitungan interpolasi dan regresi...	40
Tabel 4.4.	Jaringan dilatih dengan laju pelatihan 0,1 dan momentum 0,1.....	41
Tabel 4.5.	Jaringan dilatih dengan laju pelatihan 0,2 dan momentum 0,1.....	42
Tabel 4.6.	Jaringan dilatih dengan laju pelatihan 0,3 dan momentum 0,1.....	43
Tabel 4.7.	Jaringan dilatih dengan laju pelatihan 0,4 dan momentum 0,1.....	44
Tabel 4.8.	Jaringan dilatih dengan laju pelatihan 0,5 dan momentum 0,1.....	45
Tabel 4.9.	Rekapitulasi error masing-masing model jaringan.....	47
Tabel 4.10.	Jaringan dilatih dengan laju pelatihan 0,1 dan momentum 0,1.....	48
Tabel 4.11.	Jaringan dilatih dengan laju pelatihan 0,1 dan momentum 0,2.....	49
Tabel 4.12.	Jaringan dilatih dengan laju pelatihan 0,1 dan momentum 0,3.....	50
Tabel 4.13.	Jaringan dilatih dengan laju pelatihan 0,1 dan momentum 0,4.....	51
Tabel 4.14.	Jaringan dilatih dengan laju pelatihan 0,1 dan momentum 0,5.....	52
Tabel 4.15.	Rekapitulasi error masing-masing model jaringan.....	53
Tabel 4.16.	Laju korosi yang didapatkan dari JST.....	54

Bab 1. Pendahuluan

1.1. Latar Belakang

Baja tahan karat merupakan material yang dirancang khusus untuk menahan laju korosi. Namun demikian setiap material yang dikenai perlakuan pengelasan struktur mikronya akan mengalami perubahan. Perubahan ini akan mempengaruhi timbulnya korosi dari material tersebut. Sehingga untuk mengembalikan ketahanan korosi pada stainless steel tersebut perlu dilakukan adanya suatu perlakuan panas dan kemudian didinginkan secara cepat agar laju korosi tersebut dalam diperkecil. Dari percobaan telah diketahui berapa besarnya suhu yang paling baik untuk melakukan perlakuan panas pasca pengelasan baja tahan karat AISI 316. Variasi temperatur yang digunakan pada percobaan berkisar antara 850°C – 1000°C , yang mana suhu pemanasan yang digunakan ini adalah di atas suhu sensitisasi yaitu di atas 800°C [Sulistianto, 2007]. Pada dasarnya suhu pemanasan yang mempengaruhi timbulnya laju korosi ini adalah sampai suhu 1120°C , namun karena memperhitungkan biaya pengujian yang mahal dan terbatasnya peralatan maka percobaan hanya dilakukan sampai suhu 1000°C saja.

Dalam tugas akhir ini akan dikembangkan penganalisaan mengenai hasil pengujian tersebut dengan menggunakan jaringan saraf tiruan. Dengan menggunakan data dari pengujian, selanjutnya akan diperkirakan bagaimana hasil yang akan didapatkan jika suhunya dinaikkan sampai suhu 1120°C . Ini dilakukan karena pada dasarnya suhu pemanasan pasca pengelasan yang mempengaruhi berkisar antara 850 sampai 1120°C . [Fontana, 1987]

Jaringan saraf tiruan (*Artificial Neural Network*) adalah suatu metode pemrosesan informasi berbasis komputer yang mensimulasikan dan memodelkan suatu permasalahan seperti layaknya saraf biologis manusia, yaitu memproses informasi-informasi yang datang dari luar kemudian memberi respon atas masukan yang datang dari luar tersebut dan akhirnya membuat suatu keputusan. Begitu pula pada penelitian kali ini, data-data yang telah didapatkan akan difungsikan sebagai masukan yang datang dari luar untuk selanjutnya diproses ke dalam jaringan saraf tiruan. Model diberi pembelajaran sampai akhirnya menghasilkan suatu pola jaringan yang mampu menganalisa dan membuat suatu pola tertentu. Jaringan ini selanjutnya dapat digunakan sebagai “tool” untuk menganalisa pengaruh suhu-suhu selain yang telah diuji coba.

Penganalisaan masalah akan menggunakan jaringan saraf tiruan yang dibentuk dengan proses pembelajaran terawasi, tipe *feedforward network* serta algoritma pembelajaran yang digunakan adalah *error backpropagation*.

1.2. Perumusan Masalah

Pada penelitian ini, metode *Artificial Neural Network (ANN)* akan diterapkan pada penentuan besarnya laju korosi di atas suhu 1000°C. Perhitungan menggunakan prosedur *backpropagation*. Data *input* didapatkan dari data percobaan, kemudian dari data-data tersebut dilakukan perluasan data dan *training* sampai dengan beberapa iterasi tertentu sehingga akhirnya akan menemukan suatu pola jaringan saraf tiruan yang sesuai dengan apa yang diharapkan. Masalah utama dalam penelitian ini adalah bagaimana mendapatkan model *neural network* yang paling sesuai untuk dapat belajar dan menganalisa semua kondisi suhu yang diinputkan.

1.3. Tujuan

Tujuan dari penelitian ini adalah menemukan suatu model jaringan saraf tiruan yang dapat digunakan untuk :

- 1). Mengetahui pengaruh temperatur pada proses perlakuan panas pasca pengelasan terhadap laju korosi.
- 2). Mengetahui temperatur dengan laju korosi minimal pada baja AISI 316.
- 3). Mengetahui laju korosi pada temperatur pemanasan pasca pengelasan di atas suhu 1000°C.

1.4. Manfaat

Dengan dilakukannya percobaan mengenai pengaruh suhu pemanasan terhadap hasil lasan dengan menggunakan *artificial neural network* (jaringan saraf tiruan) ini maka nantinya akan didapatkan suatu model jaringan yang dapat digunakan langsung secara dini untuk menganalisa berapa suhu pasca pengelasan (PWHT) yang paling bagus untuk memperkecil kemungkinan timbulnya korosi dari suatu hasil lasan.

1.5. Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah :

1. Data sekunder dari hasil percobaan material baja tahan karat AISI 316 yang telah diteliti sebelumnya.
2. Pengelasan menggunakan SMAW.
3. Variasi suhu data sekunder adalah 850°C, 900°C, 950°C, dan 1000°C.
4. Analisa data menggunakan *artificial neural network* yang dilatih dengan *backpropagation algorithm*.

1.6. Hipotesis Awal

Perkiraan awal dari tugas akhir ini adalah dengan semakin naiknya suhu pemanasan maka akan memperkecil presipitasi karbida krom pada daerah batas butir sehingga laju korosi akan semakin kecil/ berkurang.

1.7. Metodologi

Metode dan langkah-langkah yang digunakan dalam penyusunan tugas akhir ini meliputi :

1. Identifikasi masalah

Tahap ini merupakan tahap awal dari pengerjaan tugas akhir, yaitu dengan menetapkan tujuan penelitian serta perumusan masalah dari apa yang akan dikerjakan.

2. Studi literatur.

Studi kepustakaan dilakukan dengan mencari dan mempelajari literatur-literatur yang sesuai, sehingga dapat mempermudah dalam proses penelitian dan analisa data penelitian.

3. Pengolahan data analisis

Untuk melaksanakan pengolahan data dalam penelitian ini, ada beberapa tahap yang dilakukan, tahap-tahap tersebut adalah :

- ❖ Mengumpulkan data-data sekunder untuk kemudian diolah ke dalam program pelatihan jaringan.
- ❖ Melakukan penambahan data dari data sekunder di atas.
Penambahan data dilakukan dengan menggunakan regresi polynomial dan interpolasi.
- ❖ Memodelkan data-data sekunder ke dalam program Delphi.
- ❖ Menyusun hasil dari pemrograman tiap data sekunder untuk diambil sebuah kesimpulan dari material yang mengalami perlakuan panas pasca pengelasan dengan suhu tertentu.

4. Analisa hasil pengujian dan pembahasan

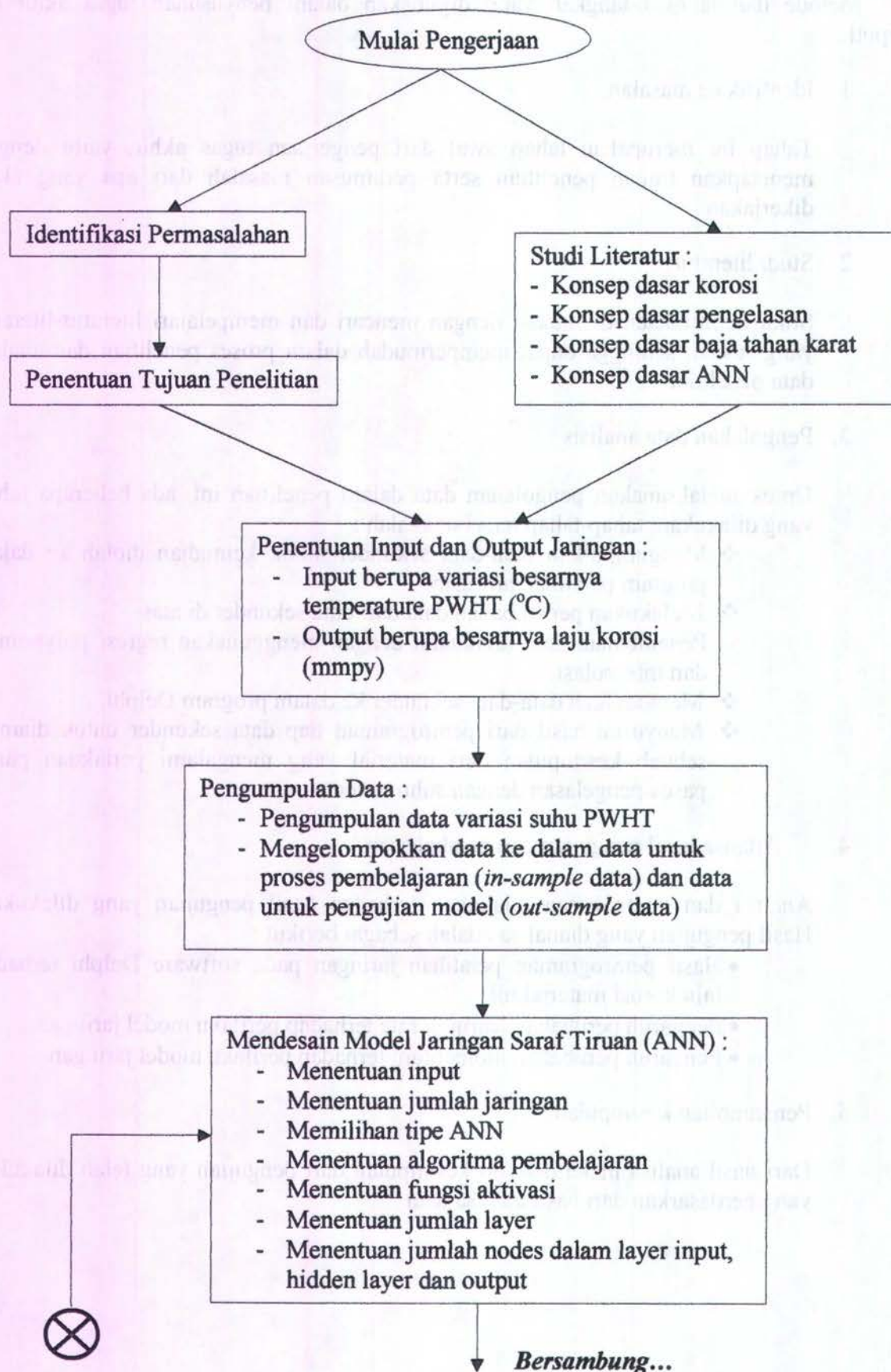
Analisa dan pembahasan dilakukan terhadap hasil pengujian yang dilakukan. Hasil pengujian yang dianalisa adalah sebagai berikut :

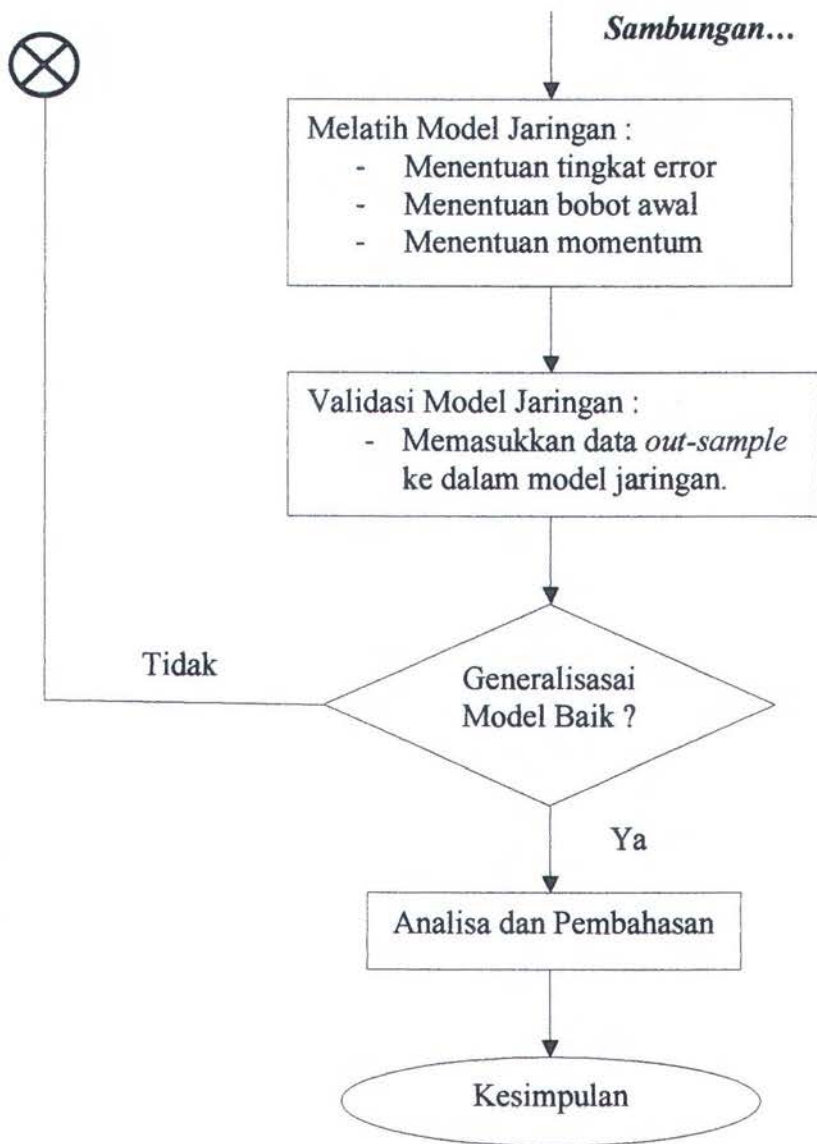
- Hasil pemrograman pelatihan jaringan pada software Delphi terhadap laju korosi material uji.
- Pengaruh perubahan learning rate terhadap perilaku model jaringan.
- Pengaruh perubahan momentum terhadap perilaku model jaringan.

5. Pengambilan kesimpulan

Dari hasil analisa maka diambil kesimpulan dari pengujian yang telah dilakukan yang berdasarkan dari hasil analisa data.

Metode dan langkah-langkah di atas dapat ditampilkan dalam bentuk diagram, seperti yang tampak pada gambar di bawah ini :





Gambar 1.1. Flow chart metodologi.

1.8. Sistematika Penulisan

Untuk memperoleh hasil laporan tugas akhir yang sistematis dan tidak keluar dari pokok permasalahan maka dibuat sistematika sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini akan dibahas mengenai latar belakang masalah, perumusan masalah, tujuan, dan manfaat penulisan tugas akhir.

BAB II TINJAUAN PUSTAKA

Berisikan teori yang memberikan penjelasan mengenai teori yang digunakan dalam penganalisaan masalah yang diteliti.

BAB III METODOLOGI PENELITIAN

Pada bab ini akan dibahas tentang kegiatan yang dilakukan selama proses pembuatan model pada software yang digunakan.

BAB IV ANALISA DAN PEMBAHASAN

Pada bab ini akan berisi analisa beserta pembahasannya yang dilakukan terhadap hasil perhitungan model dan error yang dihasilkan.

BAB V KESIMPULAN DAN SARAN

Pada bab ini akan berisi kesimpulan dari tugas akhir dan saran untuk penelitian selanjutnya.

DAFTAR PUSTAKA

LAMPIRAN

Bab 2. Tinjauan Pustaka

2.1. Jaringan Saraf Tiruan (*Artificial Neural Network*)

2.1.1. Menenal Jaringan Saraf Tiruan

Artificial Neural Networks adalah suatu teknologi komputasi yang berbasis hanya pada model saraf biologis dan mencoba mensimulasikan tingkah laku dan kerja model saraf terhadap berbagai macam masukan. [Pandjaitan, 2007]

Hal yang sama diutarakan oleh Simon Haykin, yang menyatakan bahwa JST (Jaringan Saraf Tiruan) adalah sebuah mesin yang dirancang untuk memodelkan cara otak manusia mengerjakan fungsi atau tugas-tugas tertentu. Mesin ini memiliki kemampuan menyimpan pengetahuan berdasarkan pengalaman dan menjadikan simpanan pengetahuan yang dimiliki menjadi bermanfaat.

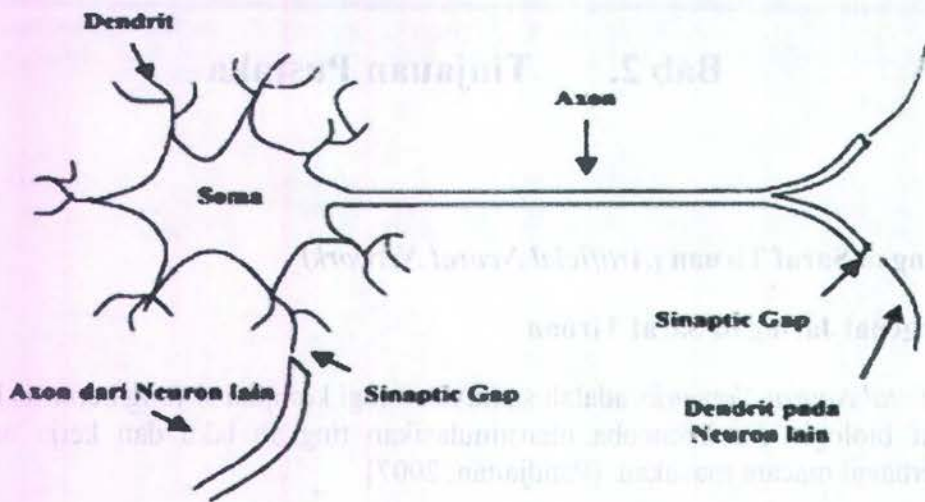
2.1.2. Sistem Jaringan Secara Biologis

Fausett (1994) mengemukakan bahwa jaringan saraf tiruan (JST) merupakan suatu sistem pemrosesan informasi yang memiliki karakteristik-karakteristik menyerupai jaringan saraf biologi, sama seperti otak yang memproses suatu informasi. Elemen mendasar dari paradigma tersebut adalah struktur yang baru dari sistem pemrosesan informasi. Jaringan saraf tiruan, seperti manusia, belajar dari suatu contoh. Contoh-contoh yang ada berupa data-data yang dimasukkan kedalam model jaringan.

Otak manusia terdiri dari lebih dari kurang lebih 10.000.000.000 sel saraf yang saling berhubungan. Sel saraf mempunyai cabang struktur input (dendrites), sebuah inti sel dan percabangan struktur output (axon). Axon dari sebuah sel terhubung dengan dendrites yang lain melalui sebuah synapse. Ketika sebuah sel saraf aktif, kemudian menimbulkan suatu signal electrochemical pada axon. Signal ini melewati synapses menuju ke sel saraf yang lain. Sebuah sel saraf lain akan mendapatkan signal jika memenuhi batasan tertentu yang sering disebut dengan nilai ambang atau (threshold).

Jaringan saraf tiruan disusun dengan asumsi yang sama seperti jaringan saraf biologis manusia (neuron), yaitu:

1. Pengolahan informasi terjadi pada elemen-elemen pemrosesan (neuron-neuron).
2. Sinyal antara dua buah neuron diteruskan melalui link-link koneksi.
3. Setiap link koneksi memiliki bobot terasosiasi.
4. Setiap neuron menerapkan sebuah fungsi aktivasi terhadap input jaringan (jumlah sinyal input berbobot). Tujuannya adalah untuk menentukan sinyal output. Fungsi aktiasi yang digunakan biasanya fungsi non-linier.



Gambar 2.1. Diagram sederhana dari neuron. [Fausett, 1994]

Berikut ini adalah hubungan antara konsep biologi dengan jaringan saraf tiruan menurut Medsker dan Liebowitz (1994) dalam Turban (2001) yang digambarkan dalam tabel di bawah ini :

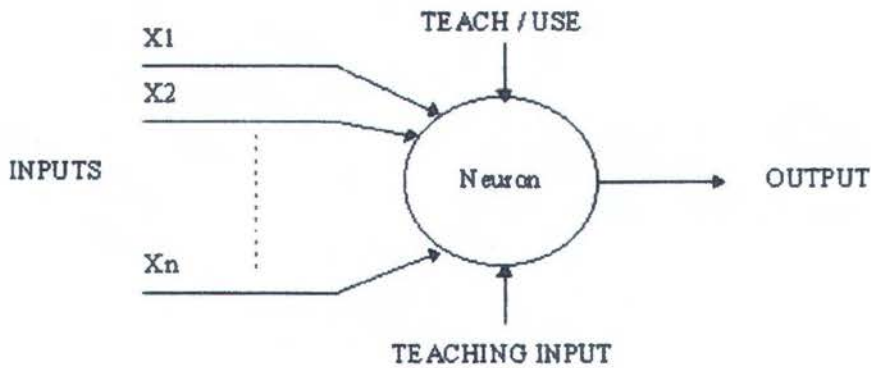
Tabel 2.1. Hubungan antara biologi dan jaringan saraf tiruan.

Biologi	Jaringan saraf tiruan
Soma	Node (simpul)
Dendrite	Input
Axon	Output
Synapse	Weight (bobot)
Slow speed	Fast speed
Terdiri dari banyak akson (10^9)	Beberapa neuron

2.1.3. Struktur Jaringan Saraf Tiruan

Pada dasarnya, struktur jaringan saraf tiruan terdiri dari *input*, proses/hiding, dan *output*. *Input* merupakan masukan yang digunakan baik saat pembelajaran maupun dalam mengenali suatu objek, input ini dapat lebih dari satu buah data sebagai inputnya. *Weight* adalah beban yang selalu berubah setiap kali diberikan *input* sebagai proses pembelajaran. Setiap sel saraf mempunyai sebuah nilai ambang. Jumlah bobot dari input dan dikurangi dengan nilai ambang kemudian akan mendapatkan suatu aktivasi dari sel saraf.

Signal aktivasi kemudian menjadi fungsi aktivasi/fungsi transfer untuk menghasilkan output dari sel saraf. Teaching Unit (neuron) merupakan tempat berlangsungnya proses pengenalan suatu objek berdasarkan pembebanan yang diberikan. Output yang dihasilkan merupakan hasil prediksi dari suatu proses yang dilakukan terhadap input yang diberikan.



Gambar 2.2. Struktur artificial neural networks. [http://trirezqiariantoro.files.wordpress.com/2007/05/jaringan_syaraf_tiruan.pdf]

Sebuah pemodelan jaringan memiliki masukan input X_n sebanyak n , yang berasal dari inputan luar (bukan dari neuron/jaringan tersebut). Selanjutnya setiap inputan X_n akan dikalikan dengan pembobot W_n yang berkesesuaian. Untuk semua hasil perkalian akan dijumlahkan sebagaimana seperti pada persamaan di bawah ini : [Fausett, 1994]

$$y_{in} = \sum_{j=1}^n w_j x_j \quad (2.1)$$

atau jika dijabarkan akan menjadi :

$$y_{in} = w_1 x_1 + w_2 x_2 + w_3 x_3 + \dots + w_n x_n \quad (2.2)$$

dan hasil persamaan tersebut akan menjadi masukan bagi fungsi *aktivasi* untuk mendapatkan tingkat derajat sinyal keluaran pada neuron, dimana terdapat bermacam-macam jenis fungsi *aktivasi*.

2.1.4. Pembelajaran dalam Jaringan Saraf Tiruan

Secara umum, dalam penggunaan jaringan saraf tiruan, hubungan antara input dan output harus sudah diketahui secara pasti. Ini ditujukan agar permasalahan yang akan dipecahkan dapat dibuat ke dalam suatu model. Agar suatu input dapat menghasilkan output yang sesuai dengan harapan, maka hal terpenting dalam pembuatan jaringan adalah memikirkan bagaimana proses menghubungkan antara input dan output tersebut. Dan tentunya proses belajar hubungan antara input dan output ini dapat dilakukan dengan cara melakukan pembelajaran/melatih jaringan.

Hal yang ingin dicapai dengan melatih/mengajari jaringan saraf tiruan adalah untuk mencapai keseimbangan antara kemampuan *memorisasi* dan *generalisasi*. Yang dimaksud dengan kemampuan *memorisasi* adalah kemampuan jaringan saraf tiruan untuk memanggil kembali secara sempurna sebuah pola yang telah dipelajari. Kemampuan *generalisasi* adalah kemampuan jaringan saraf tiruan untuk menghasilkan respon yang dapat diterima terhadap pola-pola input yang serupa (namun tidak identik) dengan pola yang sebelumnya telah dipelajari. Hal ini sangat bermanfaat bila pada suatu saat ke dalam jaringan saraf tiruan itu diinputkan informasi baru yang belum pernah dipelajari, maka jaringan saraf tiruan itu masih akan tetap dapat memberikan tanggapan yang baik, memberikan keluaran yang paling mendekati. [Puspitaningrum, 2006]

Ada beberapa aturan pembelajaran yang biasa digunakan dalam JST (Jaringan Saraf Tiruan), diantaranya adalah seperti yang tampak pada table 2.2 dan table 2.3 di bawah ini :

Table 2. 2. Paradigma, aturan pembelajaran, arsitektur, algoritma, dan aplikasi jaringan saraf tiruan yang terawasi. [Puspitaningrum, 2006]

Paradigma	Aturan Pembelajaran	Arsitektur	Algoritma Pembelajaran	Bidang Kerja
Terawasi (supervised)	Pengkoreksian eror	Perceptron lapisan tunggal atau multi lapis	<ul style="list-style-type: none"> • Perceptron • Propagasi balik • Adaline dan Madaline 	<ul style="list-style-type: none"> ♦ Klasifikasi pola ♦ Aproksimasi fungsi ♦ Prediksi, control
	Boltzmann	Recurrent (berulang)	Boltzmann	Klasifikasi pola
	Hebbian	Umpan maju multilapisan	Analisis deskriminan linear	<ul style="list-style-type: none"> ♦ Analisa data ♦ Klasifikasi pola
	Kompetitif	Kompetitif	LVQ (Learning Vector quantization)	<ul style="list-style-type: none"> ♦ Kategori kelas ♦ Kompresi data
		Jaringan ART	ARTMap	<ul style="list-style-type: none"> ♦ Klasifikasi pola ♦ Kategori kelas

Pada pembelajaran yang terawasi, metode ini digunakan jika ouput yang diharapkan sudah diketahui sebelumnya. Biasanya pembelajaran dilakukan dengan menggunakan data-data yang telah ada. Perbedaan antara output-output aktual dengan output-output yang diinginkan digunakan untuk mengoreksi bobot jaringan saraf tiruan agar jaringan dapat menghasilkan jawaban yang sedekat mungkin dengan jawaban yang benar yang telah diketahui oleh jaringan saraf tiruan.

Table 2. 3. Paradigma, aturan pembelajaran, arsitektur, algoritma, dan aplikasi jaringan saraf tiruan yang tidak terawasi. [Puspitaningrum, 2006]

Paradigma	Aturan Pembelajaran	Arsitektur	Algoritma Pembelajaran	Bidang Kerja
Tidak terawasi	Pengkoreksian Error	Multilayer feed forward	Proyeksi Sammon	Analisa data
	Hebbian	Umpan maju atau kompetitif	Analisis komponen principal	<ul style="list-style-type: none"> ◆ Analisa data ◆ Kompresi data
		Hopfield net	Aturan memori asosiatif	Memori aosiatif
	Kompetitif	Kompetitif	Kuantisasi vector	<ul style="list-style-type: none"> ◆ Kategorisasi ◆ Kompresi data
		Kohonen SOM	Kohonen SOM	<ul style="list-style-type: none"> ◆ Kategorisasi ◆ Analisa data
		Jaringan ART	ART1, ART2	Kategorisasi
Hibrida (hybrid)	Pengkoreksian error dan kompetitif	Jaringan RBF	Algoritma RBF	<ul style="list-style-type: none"> ◆ Klasifikasi pola ◆ Aproksimasi fungsi ◆ Prediksi, kontrol

Pada metode pembelajaran yang tidak terawasi, tidak memerlukan target output. Metode ini tidak dapat ditentukan hasil seperti apa yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu range tertentu tergantung pada nilai input yang diberikan. Struktur menggunakan dasar data atau korelasi antara pola-pola data yang dieksploitasi. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran seperti ini biasanya sangat cocok untuk pengelompokkan (klasifikasi) pola.

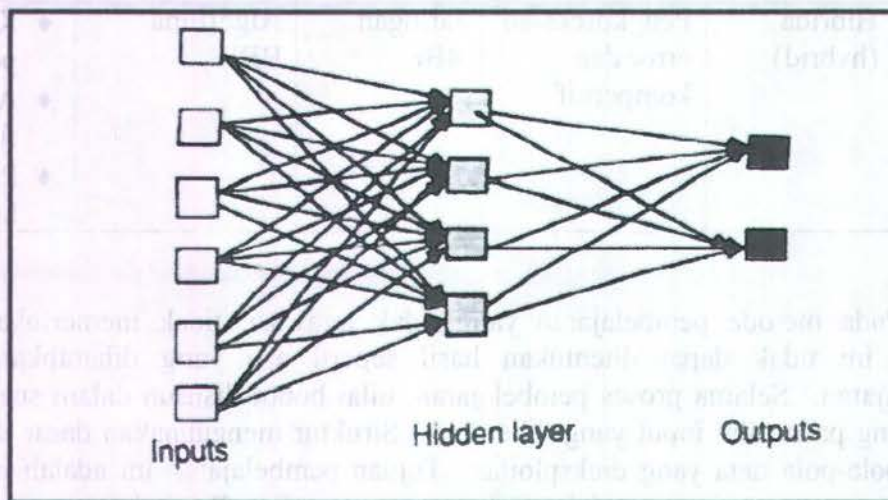
2.1.5. Pola Koneksi Jaringan Saraf Tiruan

Jaringan saraf tiruan dibentuk atas node-node yang saling berhubungan antar satu dengan yang lainnya. Sehingga keseluruhan node dalam jaringan ini akan memiliki hubungan yang tidak dapat dipisahkan. Sebagai contohnya organ pada manusia, ketika kita mengambil sebuah batu, maka terlebih dahulu sensor akan bekerja pada saraf sensorik dan motorik kemudian signal datang dari mata dan diteruskan ke tangan untuk kemudian mengambil batu tersebut. Hal ini berlaku pula pada pembuatan sebuah jaringan saraf tiruan. Berdasarkan dari arsitektur pola koneksinya, jaringan saraf tiruan dapat dibagi menjadi dua kategori, yaitu struktur *feedforward* dan struktur *feedback*. Namun untuk JST yang paradigmanya terawasi umumnya menggunakan struktur *feedforward*, karena memang sejauh ini struktur inilah yang sangat berguna untuk memecahkan masalah.

2.1.6. Struktur *feedforward*

Sebuah jaringan yang sederhana mempunyai struktur *feedforward* dimana signal bergerak dari input kemudian melewati lapisan tersembunyi dan akhirnya mencapai unit output (mempunyai struktur perilaku yang stabil). Tipe jaringan *feedforward* mempunyai sel saraf yang tersusun dari beberapa lapisan. Lapisan input bukan merupakan sel saraf. Lapisan ini hanya memberi pelayanan dengan mengenalkan suatu nilai dari suatu variabel. Lapisan tersembunyi dan lapisan output sel saraf terhubung satu sama lain dengan lapisan sebelumnya. Kemungkinan yang timbul adalah adanya hubungan dengan beberapa unit dari lapisan sebelumnya atau terhubung semuanya (lebih baik).

Struktur JST yang termasuk kedalam kategori ini antara lain : single layer perceptron, multilayer perceptron, radial-basis function networks, polynomial learning networks, dan lain lain.

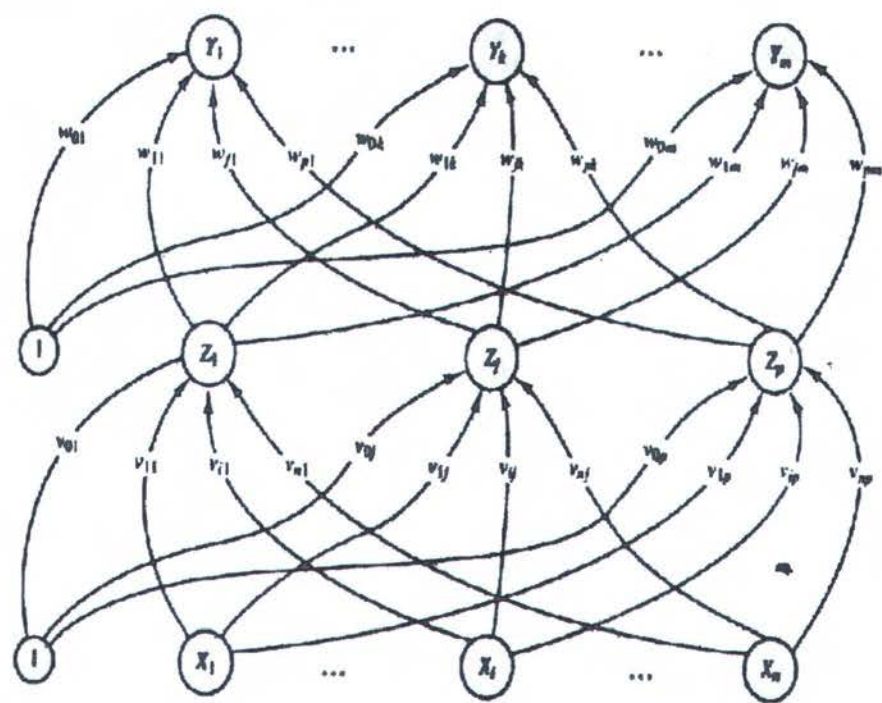


Gambar 2.3. Jaringan saraf tiruan *feedforward*. [http://trirezqiariantoro.files.wordpress.com/2007/05/jaringan_syaraf_tiruan.pdf]

2.2. Backpropagation Neural Network

2.2.1. Metode *Backpropagation*

Desiani (2006) dalam bukunya yang membahas tentang konsep kecerdasan buatan menyatakan bahwa metode *backpropagation* (propagasi balik) adalah merupakan metode pembelajaran lanjut yang dikembangkan dari aturan perceptron. Hal yang ditiru dari perceptron adalah tahapan dalam algoritma jaringan.



Gambar 2.4. Arsitektur *backpropagation*. [Fausett, 1994]

Didalam jaringan propagasi balik, setiap unit yang berada di lapisan input terhubung dengan setiap unit yang ada di lapisan tersembunyi. Hal serupa berlaku pula pada lapisan tersembunyi. Setiap unit yang ada di lapisan tersembunyi terhubung dengan setiap unit yang ada di lapisan output.

Jumlah lapisan tersembunyi pada *backpropagation* ditentukan dengan percobaan. Semakin banyak jumlah lapisan tersembunyi diharapkan jaringan akan memberikan hasil yang lebih akurat, tetapi proses pelatihannya lebih rumit dan butuh waktu lama. Penentuan jumlah neuron pada lapisan tersembunyi disesuaikan dengan jumlah neuron pada lapisan masukan.

Fase perhitungan backpropagation secara sederhana dapat dijelaskan sebagai berikut :

a. *Forward propagation*

Menyalurkan input ke dalam jaringan dan tiap layer akan mengeluarkan output. Output dari satu layer akan menjadi input untuk layer berikutnya.

b. *Back propagation*

Menghitung nilai error yang timbul akibat perbedaan antara output actual dengan output target dan digunakan untuk meng-update bobot-bobot yang relevan dengan jalan mempropagasikan kembali error. Setiap perubahan bobot yang terjadi diharapkan dapat mengurangi besarnya error. Iterasi (siklus setiap pola pelatihan) seperti ini dilakukan pada semua set pelatihan sampai unjuk kerja jaringan mencapai tingkat yang diinginkan atau sampai kondisi berhenti disini. Yang dimaksud dengan istilah berhenti disini adalah seumpama pelatihan akan dihentikan setelah iterasi mencapai 500 kali atau mungkin pelatihan telah mencapai 1000 kali, tergantung pada pembuatannya sampai berapa kali epoch (iterasi).

c. *Weight update*

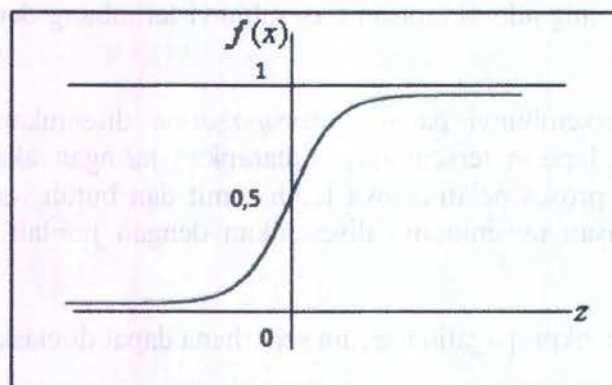
Menyesuaikan nilai parameter bobot (W) dan bias (b) dengan menggunakan pendekatan steepest descent.

2.2.2. Fungsi Aktivasi

Kusumadewi (2004) menyatakan bahwa prosedur (algoritma) *backpropagation* menggunakan *error* output untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan error ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan fungsi aktivasi sigmoid biner.

$$f(x) = \frac{1}{1 + \exp(-\alpha x)} \quad (2.3)$$

Gambar fungsi aktivasi sigmoid-biner dapat dilihat pada gambar 2.6 di bawah ini.



Gambar 2.5. Fungsi aktivasi sigmoid-biner.

Perlu diingat, bahwa fungsi aktivasi pada metode back propagation tidak hanya menggunakan sebuah fungsi aktivasi, melainkan juga turunan dari fungsi tersebut.

$$f'(x) = \sigma f(x)[1 - f(x)] \quad (2.4)$$

Fungsi aktivasi sigmoid biner ini tergantung pada steepness parameter (σ). Agar fungsi ini menghasilkan nilai yang dibatasi oleh jaringan biner (0 sampai 1) maka σ haruslah sama dengan 1 dan akan menghasilkan grafik kontinu yang tidak linier.

Selain fungsi aktivasi sigmoid biner, masih ada beberapa fungsi aktivasi yang lain, yaitu : [Fausett,1994]

1. Fungsi sigmoid bipolar.

Sama halnya dengan fungsi sigmoid biner, fungsi ini juga tergantung pada steepness parameter (σ). Fungsi ini merupakan perluasan dari fungsi sigmoid biner. Fungsi sigmoid biner diperluas hingga mencapai nilai negative melalui sumbu x. Dengan demikian untuk $\sigma = 1$, fungsi ini akan menghasilkan nilai keluaran antara -1 sampai 1. Fungsinya dapat ditulis sebagai berikut :

$$g(x) = 2f(x) - 1 = \frac{1}{1 + \exp(-\sigma x)} - 1 = \frac{1 - \exp(-\sigma x)}{1 + \exp(-\sigma x)} \quad (2.5)$$

2. Fungsi identitas.

Fungsi ini biasanya digunakan pada jaringan lapis tunggal. Fungsi ini akan menghasilkan nilai yang sama dengan nilai masukannya, grafiknya akan membentuk garis lurus atau linier dan fungsinya dapat ditulis sebagai berikut :

$$f(x) = x \quad \text{untuk semua } x \quad (2.6)$$

3. Fungsi tangga biner

Fungsi ini merupakan fungsi identitas dengan pembulatan yang bergantung pada parameter pembulatan θ . Untuk $\theta = 1$, fungsi ini akan menghasilkan nilai 1 atau 0. Dengan demikian jika grafiknya digambarkan maka akan terlihat seperti tangga pada interval antara 0 dan 1. Fungsinya adalah sebagai berikut :

$$f(x) = \begin{cases} 1, & \text{jika } x \geq 0 \\ 0, & \text{jika } x < 0 \end{cases} \quad (2.7)$$

2.2.3. Learning Rate dan Momentum

Learning rate (laju pelatihan) dan momentum merupakan parameter-parameter yang utama dalam membangun jaringan saraf tiruan. Learning rate merupakan parameter yang menentukan seberapa cepat suatu jaringan dapat mencapai nilai konvergenya. Namun ada hal yang harus diperhatikan dalam menentukan learning rate ini. Apabila learning rate nya kecil, maka pencapaian nilai konvergenya lambat, sedangkan bila learning rate nya besar, maka pencapaian konvergensinya cepat namun ada bahaya osilasi yang dapat mengakibatkan nilai minimum global tidak tercapai. Pemberian learning rate diambil secara acak dengan nilai berkisar antara 0,1 sampai dengan 0,9.

Metode sederhana untuk meningkatkan kecepatan proses pembelajaran dan menghindari terjadinya ketidakstabilan jaringan adalah dengan memodifikasi fungsi aturan delta dengan memasukkan momentum di dalamnya. Penambahan momentum bertujuan untuk menghaluskan osilasi yang terjadi. Filter momentum ini akan ditambahkan pada persamaan weight (bobot) matrix dan bias, dimana pembobotan saat ini adalah merupakan fungsi dari perubahan pembobotan sebelumnya.

Rummelhart, Hinton dan Williams (1986) menambahkan momentum pada fungsi pembaharuan pembobotan dengan menggunakan fungsi sebagai berikut ini :

$$w_y(n+1) = w_y(n) + [\Delta w_y(n) + \eta(\Delta w_y(n-1))] \quad (2.8)$$

dimana :

$$\Delta w_y(n) = \alpha \nabla_y(n)$$

Sejnowsky dan Rosenberg (1987) mendeskripsikan metode lain yang akan memberikan hasil yang sama dengan menggunakan eksponensial smoothing, sebagai berikut :

$$w_y(n+1) = w_y(n) + (1-\eta)[\Delta w_y(n)] + \eta[\Delta w_y(n-1)] \quad (2.9)$$

dimana :

α = learning rate

η = momentum

Seperti pada learning rate, tidak banyak literatur/konsep yang mendasari dalam penentuan nilai momentum ini.

2.2.4. Algoritma *Backpropagation*

Algoritma pelatihan back propagation terdiri dari dua proses, *feed forward* dan *backpropagation* dari galatnya. Selama *feed forward* masing-masing unit masukkan menerima (X) atau sinyal masukkan dari luar, kemudian sinyal tersebut disebarkan masing-masing unit pada *hidden layer* (Z), masing-masing *hidden* unit menghitung sesuai dengan fungsi aktivasinya. Dan kemudian mengirim sinyal itu kemasing-masing unit pada *output layer* (Y) akan menghitung sesuai dengan fungsi aktifasinya juga, yang akan menghasilkan sinyal keluaran sebagai respon jaringan dengan adanya pemberian pola *input* tersebut.

Pada propagasi baliknya, masing-masing output unit dibandingkan dengan hasil perhitungan aktivasi Y dengan nilai target t untuk mendapatkan *error*, berdasarkan *error* inilah akan dihitung nilai δk , selanjutnya harga *error* pada *output* unit akan disebarkan mundur ke masing-masing unit pada *hidden layer*. Selanjutnya *error* tersebut digunakan untuk memperbaiki faktor pembobot antara unit *output* dengan unit *hidden* demikian selanjutnya dicari *error* dari keluaran *hidden* untuk memperbaiki faktor pembobot antara unit *input*.

Untuk jelasnya urutan pengerjaan algoritma backpropagation dapat dijelaskan rinciannya sebagai berikut : [Fausett, 1994]

Langkah 0 :

Pemberian inisialisasi penimbang (diberi nilai kecil secara acak).

Prosedur pemberian inisialisasi bias maupun bobot, baik dari unit input ke unit tersembunyi maupun dari unit tersembunyi ke unit output yang umum dilakukan adalah menginisialisasi ke dalam sebuah interval tertentu ($-Y$ dan Y), semisal antara -0,4 sampai 0,4, -0,5 sampai 0,5 dan -1 sampai 1.

Langkah 1 :

Ulangi langkah 2 hingga 9 sampai kondisi akhir iterasi dipenuhi.

Langkah 2 :

Untuk masing-masing pasangan data pelatihan (training data) lakukan langkah 3 hingga 8.

Propagasi maju (Feedforward)

Langkah 3 :

Masing-masing unit masukan ($X_i, i = 1, \dots, n$) menerima sinyal masukan X_i dan sinyal tersebut disebarkan ke unit-unit bagian berikutnya (unit-unit lapisan tersembunyi).

Langkah 4 :

Masing-masing unit dilapisan tersembunyi dikalikan dengan faktor penimbang dan dijumlahkan serta ditambah dengan biasnya:

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.10)$$

Kemudian menghitung sesuai dengan fungsi aktivasi yang digunakan :

$$z_j = f(z_in_j) \quad (2.11)$$

Bila yang digunakan adalah fungsi sigmoid maka bentuk fungsi tersebut adalah :

$$z_j = \frac{1}{1 + \exp(-z_in_j)} \quad (2.12)$$

Kemudian mengirim sinyal tersebut ke semua unit di lapisan keluaran (unit keluaran).



Langkah 5 :

Masing-masing unit keluaran (Y_k , $K=1,2,3,...,m$) dikalikan dengan faktor penimbang dan dijumlahkan :

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (2.13)$$

Menghitung kembali sesuai dengan fungsi aktifasi :

$$y_k = f(y_in_k) \quad (2.14)$$

Back Propagation dan Galatnya

Langkah 6 :

Masing-masing unit keluaran (Y_k , $k=1,...,m$) menerima pola target sesuai dengan pola masukan saat pelatihan/training dan dihitung galatnya:

$$\delta_k = (t_k - y_k) f'(y_in_k) \quad (2.15)$$

karena $f'(y_in_k) = y_k$ menggunakan fungsi sigmoid, maka :

$$\begin{aligned} f'(y_in_k) &= f(y_in_k)(1 - f(y_in_k)) \\ &= y_k(1 - y_k) \end{aligned} \quad (2.16)$$

Menghitung perbaikan faktor penimbang (kemudian untuk memperbaiki w_{jk}).

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (2.17)$$

Menghitung perbaikan koreksi :

$$\Delta w_{0k} = \alpha \delta_k \quad (2.18)$$

dan menggunakan nilai δ_k pada semua unit lapisan sebelumnya.

Langkah 7 :

Masing-masing penimbang yang menghubungkan unit-unit lapisan keluaran dengan unit-unit pada lapisan tersembunyi (Z_j , $j=1,...,p$) dikalikan delta dan dijumlahkan sebagai masukan ke unit-unit lapisan berikutnya :

$$\delta_in_j = \sum_{k=1}^m \delta_k w_{jk} \quad (2.19)$$

Selanjutnya dikalikan dengan turunan fungsi aktifasinya untuk menghitung galat.

$$\delta = \delta_in_j f'(z_in_j) \quad (2.20)$$

Kemudian menghitung perbaikan penimbang (digunakan untuk memperbaiki v_{ij}).

$$\Delta v_{ij} = \alpha \delta_j X_i \quad (2.21)$$

Kemudian menghitung perbaikan bias (untuk memperbaiki v_{0j}).

$$\Delta v_{0j} = \alpha \delta_j \quad (2.22)$$

Memperbaiki penimbang dan bias

Langkah 8 :

Masing-masing keluaran unit (Y_k , $k=1, \dots, m$) diperbaiki bias dan penimbangnya ($j=0, \dots, p$),

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.23)$$

masing-masing unit tersembunyi (Z_j , $j=1, \dots, p$) diperbaiki bias dan penimbangnya ($j=0, \dots, n$).

$$v_{jk}(\text{baru}) = v_{jk}(\text{lama}) + \Delta v_{jk} \quad (2.24)$$

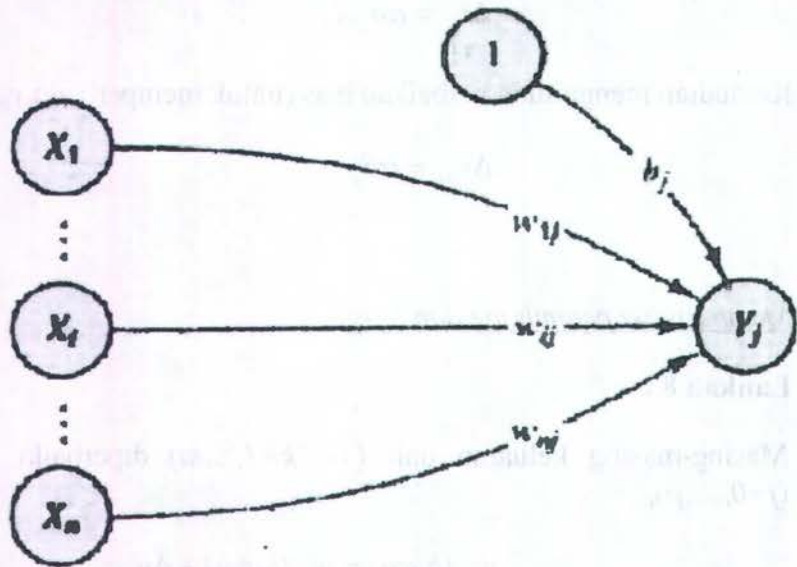
Langkah 9 :

Uji kondisi pemberhentian (akhir iterasi).

2.2.5. Bias

Nilai bias merupakan suatu konstanta input yang digunakan bersamaan dengan bobot tertentu. Bias dapat ditambahkan dengan cara menambah sebuah komponen $x_0 = 1$ kedalam vector input x , contohnya seperti $x = (1, x_1, \dots, x_2, \dots, x_n)$. Perlakuan bias dalam sebuah jaringan sama seperti perlakuan terhadap bobot, atau dengan kata lain bobot dan bias dalam sebuah jaringan adalah mempunyai peranan yang sama ($w_{0j} = b_j$). Dalam bentuk fungsi dapat ditulis sebagai berikut :

$$\begin{aligned} y_{in_j} &= xw_j \\ &= \sum_{i=0}^n x_i w_{ij} \\ &= w_{0j} + \sum_{i=1}^n x_i w_{ij} \\ &= b_j + \sum_{i=1}^n x_i w_{ij} \end{aligned} \quad (2.25)$$



Gambar 2.6. Neuron dengan sebuah bias. [Fausett, 1994]

2.2.6. Inisialisasi Jaringan

Pada proses pembelajaran jaringan propagasi balik, langkah awal yang dilakukan adalah pemberian inisialisasi bobot (dan bias) kepada model jaringan. Baik buruknya suatu model jaringan akan sangat dipengaruhi oleh pemilihan inisialisasi ini. Pilihan yang baik dari nilai awal parameter bebas, yaitu bobot yang dapat diatur pada jaringan akan sangat membantu dalam pendesainan model jaringan. Pemberian nilai awal pada parameter bebas dapat dilakukan secara random dengan distribusi seragam yang memiliki range sempit. Range untuk pemberian inisial ini bias diambil nilai acak antara -0,5 sampai +0,5 atau dapat juga nilai yang range-nya antara -1 sampai 1. Jika kita salah dalam menentukan inisialisasi bobot awal, maka akan tercipta suatu fenomena yang dikenal dengan *kejenuhan dini*, yaitu suatu keadaan dimana error kuadrat akan tetap konstan untuk beberapa saat selama proses pembelajaran berlangsung, atau sering juga disebut dengan local minima. Namun setelah beberapa saat kuadrat error akan turun kembali setelah saat-saat ini selesai dilewati.

Selain penginisialisasian secara random di atas, ada satu metode inisialisasi yang lain, namun secara prinsip hampir sama dengan inisialisasi secara random tersebut, yaitu yang dikenal dengan inisialisasi *Nguyen-Widrow*. Faktor skala inisialisasi Nguyen-Widrow (β) didefinisikan sebagai : [Fausett, 1994]

$$\beta = 0,7(p)^{1/n} \quad (2.26)$$

dimana :

n = banyak unit input

p = banyak unit tersembunyi

β = faktor skala

Faktor skala di atas digunakan untuk bias dan bobot dari unit-unit input ke unit-unit tersembunyi. Penggunaan skala ini bertujuan agar bias dan bobot dapat jatuh pada range tertentu. Dengan penskalaan ini diharapkan kemampuan belajar dari unit-unit tersembunyi dapat meningkat. Pada inisialisasi Nguyen-widrow, inisialisasi acak tetap terpakai namun digunakan untuk menginisialisasi bias dan bobot dari unit tersembunyi ke unit output saja, dan nilai acak yang sering kali digunakan pada proses pembelajaran backpropagation adalah berkisar antara $-0,5$ sampai $+0,5$.

2.3. Korosi

Korosi adalah proses perubahan struktur logam yang menimbulkan kerapuhan dan yang disebabkan karena reaksi sebagai akibat dari adanya sel konsentrasi dari ion logam atau adanya proses galvanik. Proses korosi secara kasar dapat dibagi dalam dua kelompok, yaitu korosi permukaan secara merata dan korosi setempat. Dua jenis korosi yang paling berbahaya dalam konstruksi baja adalah korosi antar butir (intergranular corrosion) dan korosi tegangan yang dapat menimbulkan retak. [Wiryosumarto, 2000].

Suatu reaksi korosi dapat berlangsung bila ada bagian yang berfungsi sebagai anoda (yang terkorosi) dan ada bagian lain yang berfungsi sebagai katoda, yang berhubungan satu sama lain. Logam dengan elektrode potensial yang lebih negatif berarti lebih mudah terkorosi, dan logam yang lebih mulia tidak terkorosi. Bila elektron yang terkumpul pada potongan logam tadi dapat mengalir ke suatu tempat lain maka keseimbangan akan terganggu, dan reaksi akan berlanjut yaitu makin banyak atom logam yang larut menjadi ion logam dan makin banyak elektron yang disalurkan ke tempat lain itu. Dalam hal ini logam tempat terjadinya reaksi oksidasi diatas akan berfungsi sebagai anoda, reaksi yang terjadi pada anoda dinamakan reaksi anodik. Elektron yang dihasilkan di anoda dialirkan ke tempat lain yaitu menuju katoda. Katoda ini dapat berupa logam lain yang dihubungkan dengan potongan logam anoda, atau bagian lain dari potongan logam yang dicelupkan ke dalam elektrolit. [Suherman, 1988].

2.3.1. Jenis-jenis Korosi

Korosi dapat diklasifikasikan dari cara karat tersebut terbentuk untuk selanjutnya pengklasifikasian korosi ditinjau dari bentuk korosi tersebut. Jenis-jenis korosi menurut bentuknya adalah :

1. Uniform attack

Uniform attack adalah bentuk korosi yang paling banyak terjadi. Korosi jenis ini dapat terjadi dengan reaksi kimia maupun elektrokimia. Proses yang terjadi seragam pada permukaan logam dan memiliki daerah yang luas. Logam akan semakin tipis dan pada akhirnya akan tidak layak dipakai pada sebuah konstruksi. Uniform attack adalah jenis korosi yang paling banyak menghilangkan logam tetapi tidak terlalu berbahaya jika ditinjau dari segi teknik, karena *life time* dari sebuah konstruksi yang mengalami korosi jenis ini dapat diperkirakan secara tepat hanya dengan percobaan yang sederhana. Untuk mencegah atau mengurangi terjadinya korosi jenis ini dapat dilakukan cara-cara seperti pengecatan, inhibitors, dan perlindungan katodik. [Fontana, 1987]

2. Galvanic corrosion

Galvanic corrosion sering disebut dengan korosi logam tak sejenis. *Galvanic corrosion* terjadi jika ada dua logam tak sejenis berada dalam satu elektrolit. Ketika dua logam yang tak sejenis berada dalam satu elektrolit maka akan terjadi beda potensial yang tetap, kemudian akan terjadi aliran electron antara kedua jenis logam tersebut. Logam dengan daya tahan korosi lebih baik akan terlindung dari korosi dan logam dengan daya tahan korosi lebih buruk akan mengalami korosi., bahkan korosi yang terjadi akan lebih hebat dibandingkan jika logam tersebut tidak bersama logam yang memiliki daya tahan korosi yang lebih baik tersebut. Logam dengan daya tahan korosi yang lebih baik dinamakan katoda, dan logam yang lebih buruk daya tahannya terhadap korosi dinamakan anoda. [Fontana, 1987]

3. Crevice corrosion

Crevice corrosion adalah jenis korosi yang sering terjadi di dalam celah dan pada daerah permukaan logam yang terlindungi. Korosi jenis ini sering terjadi pada volume kecil yang tetap pada daerah permukaan logam, biasanya disebabkan oleh lubang, permukaan basket, dan celah di bawah paku keeling dan baut. Oleh karena itu maka korosi jenis ini sering disebut dengan crevice corrosion (korosi celah). [Fontana, 1987]

4. Pitting corrosion

Pitting corrosion adalah korosi lokal yang menyerang permukaan logam. Hasil dari korosi jenis ini adalah lubang di permukaan logam. Lubang yang terjadi memiliki diameter yang berbeda mulai dari kecil sampai besar, tetapi pada kebanyakan kasus diameter yang terjadi akibat pitting corrosion relatif kecil. [Fontana, 1987]

Pitting corrosion menyerang bagian permukaan logam yang selaput pelindungnya tergores atau retak akibat perlakuan mekanik, mempunyai tonjolan akibat dislokasi atau slip yang disebabkan oleh tegangan tarik yang dialami atau tersisa., mempunyai komposisi heterogen akibat adanya inklusi, segregasi atau presipitasi. [Trethwey, et, al, 1991]

5. Intergranular corrosion

Intergranular corrosion adalah korosi yang terjadi pada batas butir. Batas butir sering kali merupakan tempat mengumpulnya impurity atau suatu peripital, juga merupakan daerah yang lebih tegang, oleh karena itu maka tidak tertutup kemungkinan untuk terjadinya korosi korosi pada batas butir. Korosi jenis ini merupakan salah satu jenis korosi yang sangat berbahaya, karena intergranular corrosion akan sangat menurunkan kekuatan atau ketangguhan. Intergranular corrosion juga sulit untuk dideteksi, sehingga kerusakan pada suatu alat maupun konstruksi dapat terjadi tanpa diketahui tanda-tanda penyebabnya. [Suherman, 1988]

Intergranular corrosion sering terjadi pada logam yang dikenal tahan terhadap korosi, seperti stainless steel, korosi jenis ini dapat terjadi jika stainless steel mengalami pemanasan dan pendinginan lambat. Pada temperature 950° F sampai 1450° F. Cr_{23}C_6 (dan karbon) tidak dapat larut dan akan terjadi presipitasi diluar

larutan jika kadar karbon adalah 0,02% atau lebih. Krom akan keluar dari larutan, dan daerah yang berdekatan dengan batas butir akan kekurangan krom. Karbida krom pada daerah yang berdekatan dengan batas butir tidak mengalami korosi. Korosi akan menyerang pada daerah yang dekat dengan batas butir, karena pada daerah ini telah kehilangan krom sehingga daya tahannya terhadap korosi menurun. [Fontana, 1987]

6. Selective leaching

Selective leaching adalah pelepasan netto sebuah unsur dari paduan, dan karena itu sering disebut *dealloying* atau *demetalification*. Korosi jenis ini menyerang seluruh permukaan yang terbuka, sehingga bentuk keseluruhan tidak berubah, namun demikian hilangnya sebuah unsur paduan dalam jumlah besar menjadikan logam berpori-pori dan hampir tanpa kekuatan mekanik lagi. Efek ini terkadang bersifat sangat lokal sehingga yang terbentuk bukan pori melainkan lubang. Penyebab utama dari selective leaching adalah efek galvanik antara unsur-unsur berlainan yang membentuk paduan, walaupun faktor-faktor lain seperti kandungan udara dan temperatur yang berbeda juga penting. [Fontana, 1987]

7. Erosion corrosion

Erosion corrosion adalah percepatan atau penambahan korosi dari logam akibat gerak relatif antara cairan korosif dengan permukaan logam. Biasanya gerakan relatif yang terjadi termasuk gerakan cepat sehingga efek yang terjadi adalah abrasi. Permukaan logam akan hilang, sehingga hasil yang diperoleh berupa suatu bentuk aliran pada permukaan logam. Terkadang pergerakan yang terjadi di lingkungan dapat mengurangi korosi yang terjadi, kondisi tersebut terjadi jika dalam keadaan berhenti, kejadian tersebut tidak termasuk dalam erosion corrosion karena keadaan yang terjadi tidak bertambah buruk. [Fontana, 1987]

8. Stress corrosion

Stress corrosion adalah korosi yang terjadi akibat bekerjanya tegangan dan media yang korosif. Akibat dari korosi jenis ini adalah menyebabkan terjadinya keretakan pada logam. Jenis tegangan yang terjadi pada stress corrosion berupa tegangan tarik, tegangan tarik tersebut dapat berupa tegangan sisa. Yang termasuk dalam stress corrosion adalah *season cracking* pada kuningan dan *fatigue corrosion*, yang dapat menyebabkan turunnya *fatigue strength* karena adanya media yang korosif. [Suherman, 1988]

Berikut ini tabel yang dapat menunjukkan klasifikasi daya tahan korosi berdasarkan nilai dari perhitungan laju korosi.

Tabel 2.4. Klasifikasi ketahanan korosi berdasarkan nilai laju korosi [Fontana, 1987].

Relative corrosion resistance	Approximate metric equivalent				
	mpy	mm/yr	$\mu\text{m/yr}$	nm/yr	pm/sec
Outstanding	< 1	< 0.02	< 25	< 2	< 1
Excellent	1 - 5	0.02 - 0.1	25 - 100	2 - 10	1 - 5
Good	5 - 20	0.1 - 0.5	100 - 500	10 - 50	5 - 20
Fair	20 - 50	0.5 - 1	500 - 1000	50 - 150	20 - 50
Poor	20 - 200	1 - 5	1000 - 5000	150 - 500	50 - 200
Unacceptable	> 200	> 5	> 5000	> 500	> 200

2.4. Baja Tahan Karat

Baja tahan karat termasuk dalam baja paduan tinggi yang tahan terhadap korosi, suhu tinggi dan suhu rendah. Disamping itu baja tahan karat juga mempunyai ketangguhan dan sifat mampu potong yang cukup. Baja tahan karat mempunyai sifat yang berbeda baik dengan baja karbon maupun dengan baja paduan rendah, sehingga dapat mempengaruhi sifat mampulasnya. Paduan utama dari baja tahan karat adalah Cr atau Cr dan Ni dengan sedikit tambahan unsur lain seperti Mo, Cu dan Mn. [Wiryosumarto, et, al, 2000].

2.4.1. Klasifikasi Baja Tahan Karat

Berdasarkan dari komposisi, microstructural dan faktor crystallographic baja tahan karat terbagi dalam tiga jenis, yaitu : martensitic steel, ferritic steel dan austenitic steel. Berikut ini penjelasan tentang ketiga jenis baja tahan karat. [Peckner, et, al, 1977].

1. Martensitic steel

Martensitic steel merupakan salah satu tipe baja tahan karat yang dikembangkan untuk memiliki ketahanan terhadap korosi dan dapat mengalami penerasan dengan cara perlakuan panas. Pada baja tahan karat tipe ini dilakukan penambahan karbon pada campuran iron-chromium agar memperoleh paduan yang dapat mengalami perubahan dengan pemanasan dan pendinginan cepat. Untuk mengetahui jenis dan campuran dari baja tahan karat tipe martensitic dapat dilihat pada tabel di bawah ini.

Tabel 2.5. AISI Martensitic Steel. [Peckner, et, al, 1977]

AISI Tipe	(kandungan dalam %)				
	C	Mn	Cr	Ni	Lainnya
403	0.15 Max	1.0	11.5 - 13		
410	0.15 Max	1.0	11.5 - 14		
416	0.15 Max	1.2	12 - 14		0.15 S Min
420	0.15 Max	1.0	12 - 14		
431	0.20 Max	1.0	15 - 17	1.2 - 2.5	
440A	0.60 - 0.75	1.0	16 - 18		0.75 Mo Max
440B	0.75 - 0.95	1.0	17 - 18		0.75 Mo max
440C	0.95 - 1.20	1.0	18 - 18		0.75 Mo Max

Baja tahan karat tipe martensitic memiliki ketahanan korosi yang berbeda jika dibandingkan dengan baja tahan karat tipe ferritic dan austenitic. Martensitic steel memiliki kekuatan sampai dengan 270000 psi (1862 MN/m²). Pemilihan baja tahan karat tipe martensitic atau ferritic dapat dilakukan dengan dasar kebutuhan ketahanan korosi berbanding dengan kekerasan, kekuatan dan ketahanan korosi.

2. Ferritic steel

Baja tahan karat tipe ferritic berbeda dengan tipe martensitic. Baja tahan karat tipe ferritic tidak dapat mengalami penambahan kekuatan dengan cara perlakuan panas. Penambahan kekuatan dapat dilakukan dengan cara pengerjaan dingin dan anneal. Kondisi tersebut terjadi karena komposisi dari baja tahan karat tipe ferritic dibuat agar memiliki ketahanan yang baik terhadap korosi dan oksidasi serta diperuntukkan pada penggunaan temperature tinggi.

Ferritic steel memiliki ketahanan korosi yang lebih baik jika dibandingkan dengan martensitic steel, tetapi masih lebih rendah jika dibandingkan dengan austenitic steel. Untuk mengetahui jenis dan campuran dari baja tahan karat tipe ferritic dapat dilihat pada table di bawah ini.

Table 2.6. AISI Ferritic Steel. [Peckner, et, al, 1977]

AISI Type	(kandungan dalam %)			
	C	Mn	Cr	Lainnya
430	0.08 Max	1.00	16 – 18	
430F	0.12 Max	1.25	16 - 18	0.6 Mo Max
430F Se	0.12 Max	1.25	16 - 18	0.18 Se Min
446	0.20 Max	1.50	23 -27	0.25 N Max

3. Austenitic steel

Baja tahan karat tipe ini adalah jenis baja tahan karat yang paling banyak digunakan, penggunaan baja tahan karat tipe ini antara 65% sampai 70%. Seri AISI 300 adalah jenis baja tahan karat austenitic yang paling sering digunakan. Hal tersebut terjadi karena baja tahan karat tipe ini memiliki level yang tinggi pada proses fabrikasi, memiliki ketahanan korosi yang lebih baik jika dibandingkan dengan tipe lain dan memiliki variasi komponen yang banyak. Baja tahan karat tipe ini memiliki perbedaan untuk setiap tipe walau terdapat dalam satu grup. Untuk mengetahui jenis dan campuran baja tahan karat tipe austenitic dapat dilihat pada table di bawah ini.

Table 2.7. AISI Austenitic Steel. [Peckner, et, al, 1977]

AISI Tipe	(kandungan dalam %)				
	C	Mn	Cr	Ni	Lainnya
201	0.15 Max	7.5	16 - 18	3.5 - 5.5	0.25 N Max
202	0.15 Max	10.0	17- 19	4.0 - 6.0	0.25 N Max
301	0.15 Max	2.0	16 - 18	6.0 - 8.0	
302	0.15 Max	2.0	17-19	8.0 - 10	
304	0.08 Max	2.0	18 - 20	8.0 - 12	
304L	0.03 Max	2.0	18 - 20	8.0 - 12	
309	0.20 Max	2.0	22 - 24	12 - 15	
310	0.25 Max	2.0	24 - 26	19 - 22	
316	0.08 Max	2.0	16 - 18	10 - 14	2 - 3 Mo
316L	0.03 Max	2.0	16 - 18	10-14	2 - 3 Mo
321	0.08 Max	2.0	17 - 19	9 - 12	(5 x %C) Ti Min
347	0.08 Max	2.0	17 - 19	9 - 13	(10 x %C) Cb-Ta Min

Tipe 301 (17Cr-7Ni) adalah tipe yang dapat mengalami penambahan kekuatan dengan pengerjaan dingin. Baja tahan karat tipe ini digunakan pada kondisi yang menuntut adanya pembentukan, terutama jika terdapat peregangan. Tipe 301 sangat baik jika digunakan pada suhu lingkungan. Tipe 302 dan 304 memiliki keseimbangan yang lebih besar dan ketahanan korosi yang lebih baik. Tipe 304 adalah tipe yang paling banyak diproduksi dan tipe ini dapat digunakan pada suhu yang lebih tinggi. Tipe 305 adalah tipe baja tahan karat yang terdapat Nikel dalam paduannya, baja tahan karat tipe ini digunakan jika menuntut keseimbangan yang lebih tinggi. Tipe 316 dan 317 adalah tipe yang dibuat berdasar paduan dari tipe 304 tetapi dilakukan penambahan paduan molybdenum. Tipe 316 dan 317 memiliki ketahanan terhadap korosi yang lebih tinggi dan peningkatan kekuatan pada temperatur yang lebih tinggi. Tipe 309, 310 dan 314 adalah tipe dengan peningkatan paduan logam. Tipe ini dibuat untuk penggunaan pada temperatur yang lebih tinggi.

2.4.2. Pengelasan Baja Tahan Karat

Baja tahan karat adalah baja paduan tinggi, sehingga kualitas sambungan las sangat dipengaruhi oleh atmosfer pengelasan dan dapat menjadi getas akibat panas dari pengelasan. Setiap jenis baja tahan karat memiliki sifat mampu las yang berbeda. Berikut ini sifat mampu las untuk setiap jenis baja tahan karat: [Wiryosumarto, 2000].

1. Baja tahan karat jenis Martensit

Baja ini dalam siklus pemanasan dan pendinginan selama proses pengelasan akan membentuk martensit yang keras dan getas sehingga sifat mampu lasnya kurang baik. Dalam mengelas baja tahan karat jenis ini harus diperhatikan dua hal yaitu, pertama harus diberikan pemanasan mula sampai suhu antara 200 sampai 400° C dan suhu antara pengelasan lapisan harus ditahan jangan terlalu dingin dan kedua segera setelah selesai pengelasan suhunya harus ditahan antara 700 sampai 800° C untuk beberapa waktu.

2. Baja tahan karat jenis Ferit

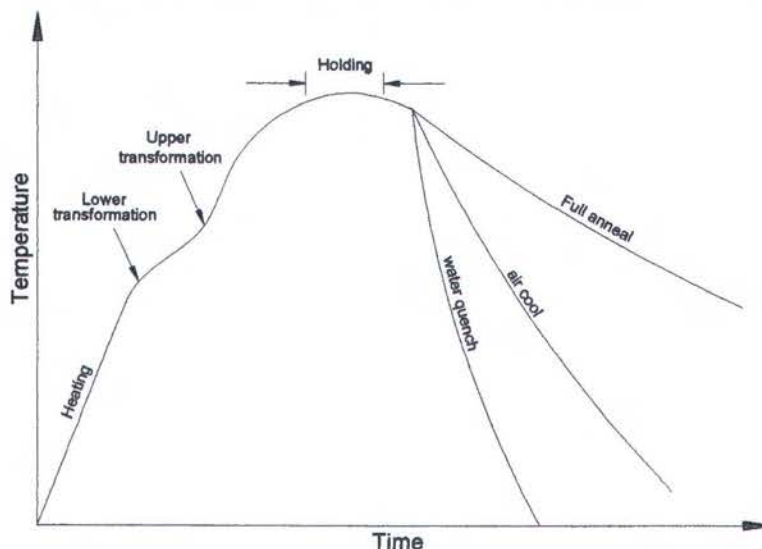
Baja tahan karat jenis ferit sangat sukar mengeras, tetapi butirnya mudah menjadi kasar yang menyebabkan ketangguhan dan keuletannya menurun. Penggetasan biasanya terjadi pada pendinginan lambat dari suhu 600° ke 400° C. Karena sifatnya ini maka pada pengelasan baja ini harus dilakukan pemanasan mula antara 70° sampai 100° C untuk menghindari retak dingin dan pendinginan dari 600° ke 400° C harus terjadi dengan cepat untuk menghindari penggetasan seperti diterangkan di atas.

3. Baja tahan karat jenis Austenit

Baja tahan karat jenis ini mempunyai sifat mampu las yang lebih baik bila dibandingkan dengan kedua jenis yang lainnya. Tetapi walaupun demikian pada pendinginan lambat dari 680° ke 480° C akan terbentuk karbida krom yang mengendap diantara butir material. Endapan ini terjadi pada suhu sekitar 650° C dan menyebabkan penurunan sifat tahan karat dan sifat mekaniknya.

2.4.3. Perlakuan Panas Baja Tahan Karat Austenitik

Pada baja tahan karat proses perlakuan panas yang sering dilakukan adalah proses perlakuan panas dengan pendinginan cepat (Quenching). Proses pemanasan dilakukan sampai semua karbida larut kembali dan austenit menjadi homogen. Temperatur pemanas untuk baja tahan karat AISI 316 yang digunakan adalah di atas suhu sensitis yaitu diatas 800° C. Temperatur tertinggi yang digunakan adalah sampai 1120° C [Fontana, 1987]. Setelah dipanaskan kemudian material didiamkan beberapa saat. Penahanan temperatur bertujuan untuk memberi kesempatan agar karbida dapat larut dan austenit lebih homogen. Langkah terakhir adalah dengan melakukan pendinginan cepat. Pendinginan cepat bertujuan agar menahan karbon dalam larutan dan mencegah presipitasi karbida krom. [Suherman, 1988].



Gambar 2.7. Grafik heat treatment [Totten, et, al, 1991].

Berbeda dengan Fontana, Jones (1996) dalam bukunya yang berjudul "*Principles and Prevention of Corrosion*" menyebutkan bahwa besarnya suhu pemanasan pada proses perlakuan panas pasca pengelasan ini bisa mencapai suhu 1230°C. Suhu pemanasan sudah tidak bisa lagi dinaikkan karena suhu tersebut sudah mendekati titik leleh dari material *austenite stainless steel*.

2.4.4. Presipitasi Karbida Krom

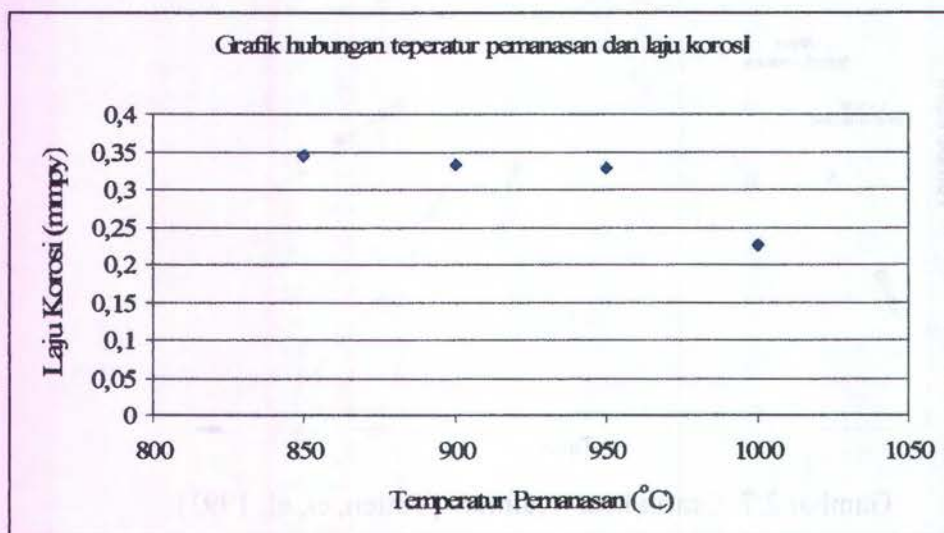
Pada pengelasan baja tahan karat terdapat masalah yang sering kali di jumpai yaitu presipitasi karbida krom. Tipe baja tahan karat yang tidak stabil (seperti tipe 302, 304, 316 dan 317) adalah tipe baja tahan karat yang dapat mengalami presipitasi karbida krom jika mengalami pemanasan pada suhu 800°F sampai 1600°F (426°C sampai 871°C) dan terjadi pendinginan lambat. Presipitasi karbida krom adalah terjadinya pengendapan karbida krom pada batas butir. Kondisi yang menyebabkan terjadinya presipitasi karbida krom terjadi selama proses pengelasan, kondisi tersebut terjadi pada Heat Affected Zone (HAZ). [Peckner, et, al, 1977].

2.5. Metode Numerik Penambahan Data

Salah satu sifat yang dimiliki oleh jaringan saraf tiruan adalah kemampuannya dalam mempelajari sesuatu. Kemampuan belajar dari model jaringan saraf tiruan ditentukan oleh seberapa banyak jumlah data yang digunakan pada proses pembelajarannya, semakin banyak data yang digunakan maka model jaringan yang akan dihasilkan akan semakin baik pula.

Data sekunder yang didapatkan dari penelitian sebelumnya berjumlah empat data yang merupakan data perbandingan antara temperatur pemanasan pada proses PWHT dan besar laju korosi yang dihasilkannya. Oleh karena itu agar model jaringan yang dibentuk dapat melalui proses belajarnya dengan baik maka selanjutnya akan dilakukan perluasan data. Perluasan data menggunakan metode regresi dan interpolasi.

Dalam menentukan metode penambahan data ini terlebih dahulu dilihat bagaimana pola grafik yang dibentuk oleh data sekunder. Pola yang dibentuk dari data sekunder lebih cenderung ke bentuk persamaan polynomial, karena grafik yang dibentuk adalah parabola.



Gambar 2.8. Pola grafik yang dibentuk oleh data sekunder.

2.5.1. Regresi Polynomial

Bentuk polynomial adalah merupakan persamaan matematika dengan variabel orde n yang dapat dirumuskan sebagai berikut :

$$y = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (2.27)$$

dimana :

y = fungsi obyektif $y(x)$

a = konstanta

jumlah kuadrat sisa yang dihasilkan dari persamaan 2.28 di atas dapat dihitung dengan persamaan :

$$s_r = \sum_{i=1}^n (y_i - a_0 - a_1x_i - a_2x_i^2 - \dots - a_nx_i^n) \quad (2.28)$$

Sedangkan untuk mencari besarnya nilai konstanta a , maka dari dapat digunakan persamaan-persamaan berikut ini.[Chapra dan Canale, 1985]

$$a_0n + a_1 \sum x_i + a_2 \sum x_i^2 + \dots + a_n \sum x_i^n = \sum y_i \quad (2.29)$$

$$a_0 \sum x_i + a_1 \sum x_i^2 + a_2 \sum x_i^3 + \dots + a_n \sum x_i^{n+1} = \sum x_i y_i \quad (2.30)$$

$$a_0 \sum x_i^2 + a_1 \sum x_i^3 + a_2 \sum x_i^4 + \dots + a_n \sum x_i^{n+2} = \sum x_i^2 y_i \quad (2.31)$$

atau jika ditulis dalam persaaan umum adalah sebagai berikut :

$$a_0 \sum x_i^n + a_1 \sum x_i^{n+1} + a_2 \sum x_i^{n+2} + \dots + a_n \sum x_i^{2n} = \sum x_i^n y_i \quad (2.32)$$

Setelah nilai dari konstanta a sudah diketahui maka akan ditemukan suatu persamaan polynomial orde n . Untuk mencari besarnya koefisien ketepatan perhitungan (R^2) yang dihasilkan dari regresi polynomial ini dapat menggunakan persamaan 2.33 berikut ini.

$$R^2 = \frac{s_t - s_r}{s_t} \quad (2.33)$$

Jika ada data yang variabelnya x dan y , maka harga S_t adalah jumlah kuadrat dari data y yang dikurangi dengan rata-ratanya. Koefisien ketepatan yang baik adalah koefisien R^2 yang nilainya sama dengan 1.

2.5.2. Interpolasi

Pada dasarnya interpolasi digunakan untuk data-data yang polanya sudah presisi/baik. Jenis-jenis interpolasi terbagi menjadi tiga, yaitu interpolasi linear, interpolasi orde dua (kuadratik), dan interpolasi orde tiga.

Interpolasi linear digunakan jika data yang ada berupa dua buah titik, sehingga persamaan yang dihasilkan merupakan persamaan garis lurus/linear, interpolasi orde dua digunakan jika data yang digunakan berjumlah tiga buah titik, sedangkan interpolasi orde tiga digunakan jika data yang digunakan berjumlah empat titik, yang selanjutnya interpolasi yang memiliki orde ini disebut interpolasi polynomial orde- n .

Persamaan umum untuk interpolasi polynomial adalah:[Chapra dan Canale, 1985]

$$f_3(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) + b_3(x - x_0)(x - x_1)(x - x_2) \quad (2.34)$$

dengan :

$$b_0 = f(x_0)$$

$$b_1 = f[x_1, x_0]$$

$$b_2 = f[x_2, x_1, x_0]$$

.

.

$$b_n = f[x_n, x_{n-1}, \dots, x_1, x_0]$$

dimana :

$$f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j} \quad (2.35)$$

$$f[x_i, x_j, x_k] = \frac{f[x_i, x_j] - f[x_j, x_k]}{x_i - x_k} \quad (2.36)$$

dan untuk orde ke- n , mengikuti persamaan :

$$f[x_n, x_{n-1}, \dots, x_1, x_0] = \frac{f[x_n, x_{n-1}, \dots, x_1] - f[x_{n-1}, x_{n-2}, \dots, x_0]}{x_n - x_0} \quad (2.37)$$

Bab 3. Metodologi Penelitian

3.1. Sumber Data dan Variabel

Data yang digunakan dalam penelitian ini merupakan data sekunder, yaitu data yang diperoleh dari penelitian sebelumnya yang telah dilakukan oleh Sulistianto, 2007.

Tabel 3.1. Data sekunder.

Temperatur PWHT	Laju korosi (mmpy)
850° C	0,345
900° C	0,332
950° C	0,327
1000° C	0,225

Data di atas kemudian diolah sedemikian rupa, sehingga dari mula-mula data yang didapatkan hanya berupa empat variasi suhu ini nantinya dapat diperluas, ini bertujuan karena program jaringan saraf tiruan mempunyai sifat layaknya saraf yang ada pada otak manusia, yaitu semakin banyak data yang digunakan untuk proses pelatihan, maka program akan lebih banyak belajar sehingga setelah ditraining program akan memberikan suatu keputusan yang akan mendekati kondisi kenyataannya.

Perluasan data dilakukan dengan dua cara, yaitu dengan menggunakan regresi *polynomial* dan *interpolasi*. Sedangkan variabel yang digunakan pada penelitian ini adalah besarnya variasi temperatur pemanasan yang berkisar antara 850° C sampai dengan 1000° C.

3.2. Metode Penelitian

Pada penelitian yang dilakukan dalam tugas akhir ini ada beberapa tahapan yang akan dilalui dalam kaitannya untuk mendesain model jaringan saraf tiruan seperti yang diharapkan.

Tahapan-tahapan tersebut adalah sebagai berikut :

3.2.1. Identifikasi Permasalahan

Permasalahan dalam penelitian ini dapat dirumuskan sebagai berikut : “Bagaimana cara menerapkan metode jaringan saraf tiruan (ANN) untuk memprediksi besarnya laju korosi pada baja tahan karat AISI 316 yang dikenai perlakuan panas setelah pengelasan dengan melakukan training/pelatihan dari data-data yang didapatkan dari penelitian sebelumnya?”.

3.2.2. Penentuan Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk menerapkan model jaringan saraf tiruan untuk memprediksi besarnya laju korosi serta untuk merancang model jaringan yang memiliki tingkat kesalahan minimal untuk selanjutnya dapat digunakan sebagai tool (alat) untuk memprediksi besarnya laju korosi (dalam *mmpy*) secara dini.

3.2.3. Studi literatur

Pada penelitian ini studi literatur dilakukan untuk memperoleh informasi selengkap-lengkapny tentang konsep penyusunan suatu model jaringan saraf tiruan dan konsep dasar mengenai pengelasan serta korosi yang terjadi pada baja tahan karat AISI 316 setelah dikenai perlakuan panas pasca pengelasan serta konsep dasar tentang baja tahan karat. Konsep dasar artificial neural network (ANN) meliputi beberapa hal, diantaranya konsep kerja, penentuan input-output, struktur jaringan, fungsi aktivasi, model-model jaringan dan juga mengenai prosedur dalam pendesainan sebuah jaringan saraf tiruan.

Konsep pengelasan meliputi seputar pengertian pengelasan, sedangkan untuk konsep korosi meliputi pengertian korosi dan berbagai jenis korosi yang dapat timbul pada suatu material. Konsep baja tahan karat meliputi pengklasifikasian baja tahan karat menurut unsur utama yang terkandung di dalamnya.

3.2.4. Identifikasi Input dan Output Jaringan

Penentuan input dan output dalam membangun sebuah jaringan sangatlah penting. Karena sesuai dengan sifatnya, jaringan yang akan dibangun disini merupakan tipe jaringan yang terawasi, sehingga untuk membangun model jaringan ini maka input dan output haruslah sudah diketahui terlebih dahulu. Pada penelitian ini input ditentukan berupa temperatur pemanasan pasca pengelasan sedangkan outputnya berupa besarnya laju korosi yang timbul akibat adanya perlakuan panas pasca pengelasan tersebut.

3.2.5. Pengumpulan Data

Pengumpulan data pada penelitian ini meliputi tiga tahapan, yaitu sebagai berikut :

- Pengumpulan data PWHT (Post Weld heat Treatment)
- Pengelompokan data menjadi data yang digunakan sebagai melatih jaringan, atau yang disebut data *in-sample*.

Data-data yang digunakan untuk melatih jaringan dipilih secara random dari data yang ada. Data-data yang terpilih sebagai data *in-sample* antara lain :

Tabel 3.2. Data *in-sample*.

Temperatur (°C)	Laju Korosi (mmpy)
850	0,345
860	0,337
880	0,330
900	0,332
910	0,334
930	0,335
950	0,327
980	0,284
990	0,258
1000	0,225

- Pengelompokan data yang digunakan untuk memvalidasi jaringan hasil pelatihan (data *out-sample*).

Model jaringan yang telah melakukan proses pembelajaran dengan menggunakan data in-sample di atas selanjutnya perlu dilakukan pengujian terhadap validitas model jaringan. Data-data yang digunakan pada tahap pengujian jaringan ini adalah data yang tidak digunakan selama proses pembelajaran. Dari uji validitas ini dilihat apakah jaringan sudah mampu melakukan generalisasi pada pola yang terbentuk dalam proses pembelajaran atau belum.

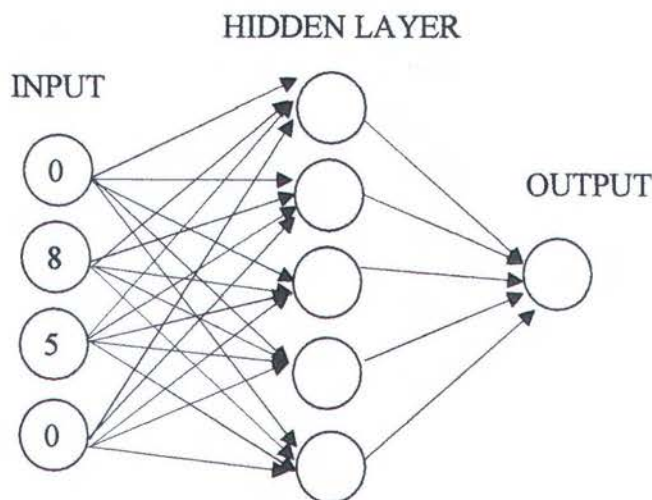
Tabel 3.3. Data *out-sample*.

Temperatur (°C)	Laju Korosi (mmpy)
870	0,332
890	0,331
920	0,335
940	0,332
960	0,318
970	0,304

3.2.6. Penentuan Vektor Input Jaringan

Pemilihan vektor input akan mempengaruhi performansi pelatihan jaringan yang akan dibentuk, vektor ini akan memberikan suatu karakter pada jaringan yang akan membantu proses perhitungan selanjutnya, dalam hal ini untuk menentukan output yang sesuai dengan apa yang kita harapkan. Pada penelitian ini menggunakan satu vektor input yang berupa variabel suhu pemanasan pasca pengelasan pada baja tahan karat AISI 316.

Agar jaringan dapat membaca vektor input dengan mudah, maka input yang pada mulanya terdiri dari tiga atau empat digit angka dipecah menjadi satuan, semisal untuk angka 1000 di pecah menjadi 1 (ribuan) 0 (ratusan) 0 (puluhan) 0 (satuan). Sedangkan untuk angka-angka yang hanya terdiri dari tiga digit, maka untuk ribumannya berupa angka 0.



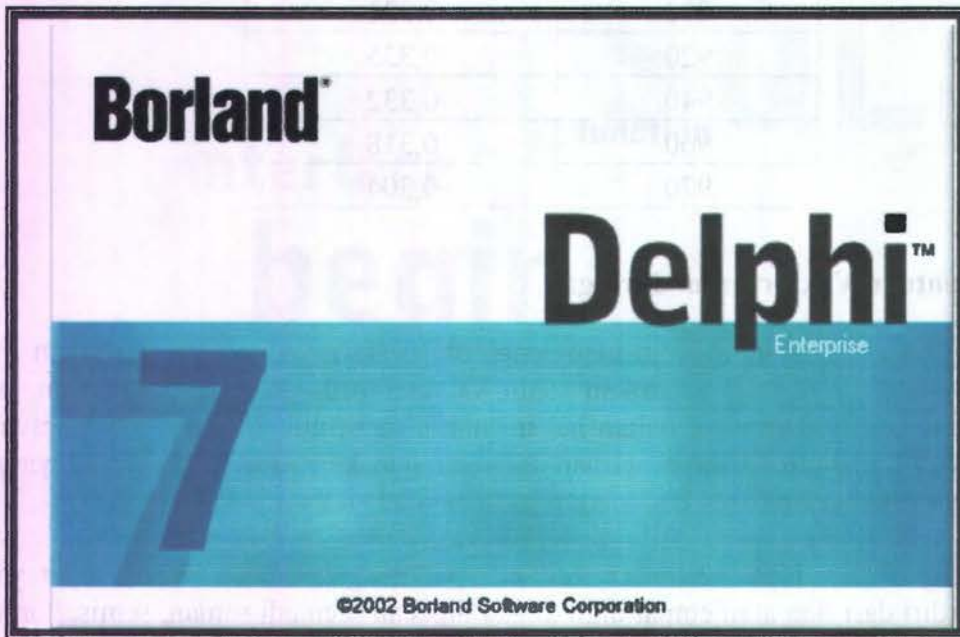
Gambar 3.1. Jaringan saraf tiruan yang direncanakan.

3.2.7. Pendefinisian Output Jaringan

Output jaringan yang diharapkan berupa angka yang menunjukkan besarnya laju korosi dalam satuan mmpy (*milimeter per year*).

3.2.8. Pendesainan Jaringan Saraf Tiruan

Pendesainan model jaringan saraf tiruan dilakukan dengan memanfaatkan software aplikasi Delphi 7 dan menggunakan jenis jaringan *feedforward* dengan algoritma pembelajaran yang dipilih adalah *error backpropagation*. Delphi 7 merupakan salah satu software yang berbasis pada bahasa pemrograman komputer. Langkah pembentukan jaringan ke dalam software ini adalah pertama-tama merepresentasikan rumus-rumus yang ada pada algoritma *backpropagation* kedalam bahasa delphi, kemudian menentukan parameter-parameter utama, seperti learning rate, momentum, dan maksimum iterasi/epoch pelatihan.



Gambar 3.2. Start up program Delphi 7.

Selain membuat bahasa pemrograman kedalam delphi, untuk mendesain sebuah jaringan saraf tiruan itu sendiri ada beberapa tahapan yang harus dilalui, diantaranya :

1. Penentuan jumlah jaringan

Penentuan jumlah jaringan pada penelitian ini adalah satu jenis jaringan yang selanjutnya akan berfungsi sebagai alat/tool untuk memprediksi besarnya laju korosi.

2. Pemilihan tipe jaringan

Tipe jaringan saraf tiruan yang digunakan adalah *feed forward network* yang mana memiliki kemampuan baik dalam pengenalan pola data dimana sinyal input akan bergerak ke depan yang akan menghasilkan sinyal output yang akan terus menyesuaikan dengan target, dengan cara melakukan pembaruan/perubahan pembobotan yang digunakan dalam tiap layer pada jaringan saraf tiruan tersebut

3. Penentuan algoritma pembelajaran

Algoritma pembelajaran yang dipilih adalah algoritma *back propagation*. Back propagation merupakan algoritma yang saat ini paling banyak digunakan dalam proses membangun jaringan saraf tiruan. Istilah back propagation sesuai dengan karakteristik algoritma ini sendiri, yang mana dalam proses pembelajarannya algoritma ini bergerak maju sampai menghasilkan suatu output, ketika hasil output masih belum mencapai target maka proses pembelajaran akan dikembalikan lagi ketiap-tiap layer yaitu dengan cara memperbaiki error yang terjadi. Algoritma backpropagation secara lengkap sudah dibahas pada bab sebelumnya.

4. Penentuan fungsi aktivasi

Fungsi aktivasi sangat menentukan dalam proses pembelajaran. Fungsi aktivasi ini memiliki beberapa macam, diantaranya fungsi linier, fungsi sigmoid, fungsi biner atau bipolar dan lain-lain. Pada penelitian ini fungsi aktivasi yang digunakan adalah fungsi sigmoid.

5. Penentuan jumlah layer

Banyaknya jumlah layer didapatkan dari eksperimental, yaitu dengan mencoba-coba. Akan tetapi pada dasarnya jaringan saraf tiruan terdiri dari 3 layer, yaitu input layer, hidden layer dan output layer dan di tiap-tiap layer memiliki jumlah node yang berbeda-beda.

6. Penentuan jumlah nodes

Jumlah unit disesuaikan dengan kebutuhan. Penelitian ini pada input layernya menggunakan 4 node, pada hidden layernya menggunakan 5 node dan pada output layernya menggunakan 1 node.

7. Pelatihan model jaringan

Menentukan parameter-parameter yang digunakan untuk proses pembelajaran adalah hal yang harus dilakukan pertama kali sebelum proses pelatihan dimulai. Inisial bobot, berat/weight dan besarnya error merupakan parameter-parameter utama dalam proses pembelajaran jaringan saraf tiruan.

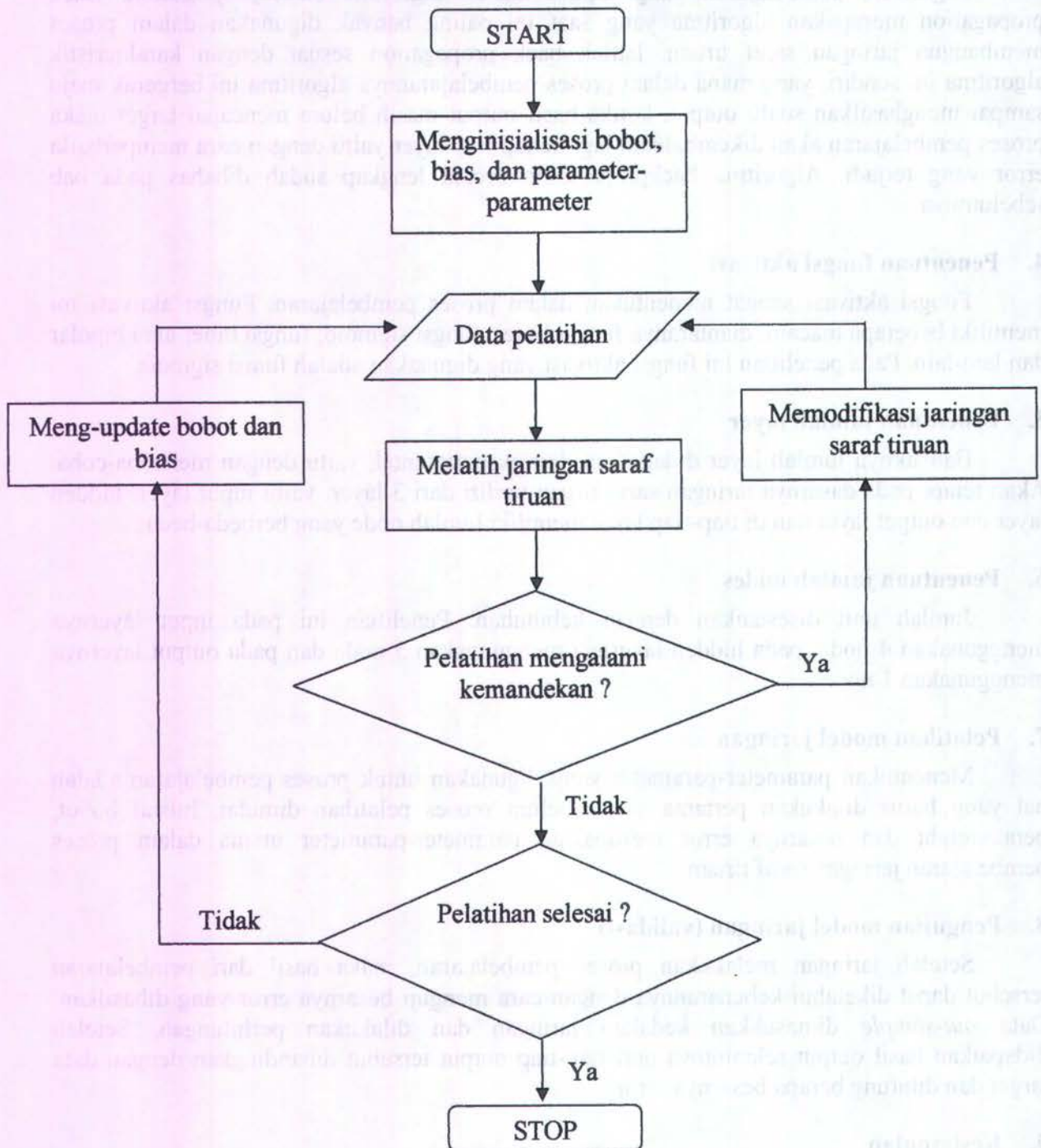
8. Pengujian model jaringan (validasi)

Setelah jaringan melakukan proses pembelajaran, maka hasil dari pembelajaran tersebut dapat diketahui kebenarannya dengan cara menguji besarnya error yang dihasilkan. Data *out-sample* dimasukkan kedalam jaringan dan dilakukan perhitungan. Setelah didapatkan hasil output selanjutnya dari tiap-tiap output tersebut dibandingkan dengan data target dan dihitung berapa besarnya error.

9. Kesimpulan

Dari proses pengujian akan menghasilkan suatu error, yaitu perbandingan antara hasil output perhitungan jaringan dengan target. Jika error yang dihasilkan sudah sesuai dengan harapan, maka jaringan ini sudah dapat digunakan untuk menganalisa penelitian, akan tetapi jika error yang dihasilkan masih jauh dari harapan, maka perlu diadakan proses perbaikan jaringan.

3.3. Diagram Alir Prosedur Jaringan saraf Tiruan



Gambar 3.3. Prosedur pelatihan jaringan saraf tiruan.

Bab 4. Analisa dan Pembahasan

4.1. Analisa Pemilihan Data Pelatihan

Data sekunder yang akan digunakan dalam pembelajaran jaringan saraf tiruan terlebih dahulu diperluas, yaitu dengan mencari nilai-nilai yang terletak diantara suhu pemanasan yang telah dilakukan pada percobaan. Hal ini dilakukan karena program jaringan saraf tiruan mempunyai sifat layaknya saraf yang ada pada otak manusia, yaitu semakin banyak data yang diinputkan, maka program akan lebih banyak mengenali pola data yang dimasukkan. Sehingga setelah ditraining program akan memberikan suatu keputusan yang akan mendekati kondisi kenyataannya. Perluasan data dilakukan dengan dua metode, yaitu dengan menggunakan regresi dan *interpolasi*.

4.1.1. Regresi Polynomial

Regresi polynomial digunakan untuk menentukan fungsi polynomial yang paling sesuai dengan kumpulan titik data (x_n, y_n) yang diketahui. Dalam tugas akhir ini titik x merupakan besarnya temperatur pemanasan yang digunakan pada proses perlakuan panas pasca pengelasan dan titik y merupakan besarnya laju korosi yang dihasilkan dari adanya proses perlakuan panas tersebut.

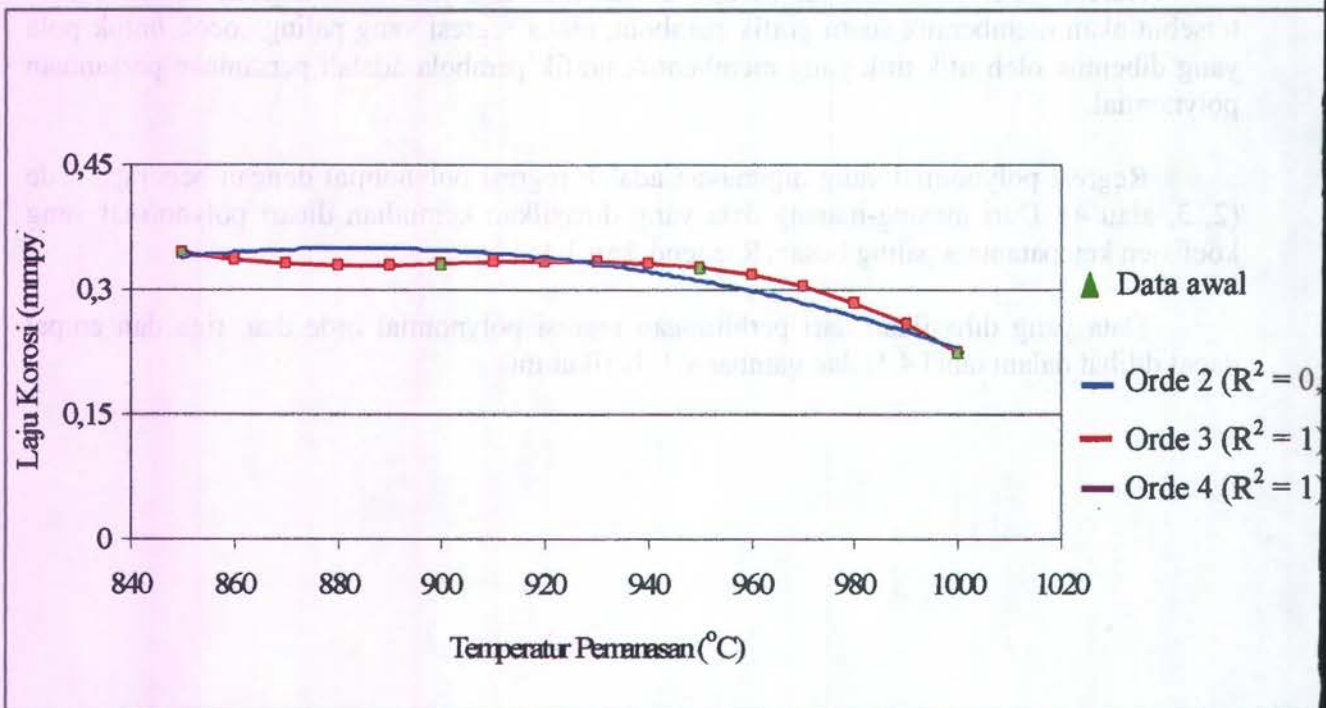
Karena data yang didapat berupa empat titik dan jika dihubungkan keempat titik tersebut akan membentuk suatu grafik parabola, maka regresi yang paling cocok untuk pola yang dibentuk oleh titik-titik yang membentuk grafik parabola adalah persamaan-persamaan polynomial.

Regresi polynomial yang digunakan adalah regresi polynomial dengan beberapa orde (2, 3, atau 4). Dari masing-masing data yang dihasilkan kemudian dicari polynomial yang koefisien ketepatannya paling besar (R^2 mendekati 1).

Data yang dihasilkan dari perhitungan regresi polynomial orde dua, tiga dan empat dapat dilihat dalam tabel 4.1. dan gambar 4.1. berikut ini :

Tabel 4.1. Laju korosi dari perhitungan dengan regresi polynomial.

SUHU (°C)	LAJU KOROSI (mmpy)		
	ORDE 2	ORDE 3	ORDE 4
850	0,340	0,345	0,345
860	0,345	0,337	0,337
870	0,348	0,332	0,332
880	0,350	0,330	0,330
890	0,350	0,331	0,331
900	0,348	0,332	0,332
910	0,344	0,334	0,334
920	0,338	0,335	0,335
930	0,331	0,335	0,335
940	0,322	0,332	0,332
950	0,311	0,327	0,327
960	0,299	0,318	0,318
970	0,284	0,304	0,304
980	0,268	0,284	0,284
990	0,250	0,258	0,258
1000	0,230	0,225	0,225



Gambar 4.1. Grafik hubungan antara temperatur pemanasan dan laju korosi dengan regresi polynomial orde 2, orde 3 dan orde 4.

Dari tabel 4.1. dan gambar 4.1 di atas terlihat bahwa grafik yang menghasilkan error paling kecil ($R^2 = 1$) adalah grafik dari regresi polynomial orde tiga dan empat. Selain itu jika dilihat besarnya laju korosi yang dihasilkan dari perhitungan polynomial orde tiga, untuk suhu-suhu 850° C, 900° C, 950° C dan 1000° C menghasilkan laju korosi yang sama dengan laju korosi yang berasal dari data awal.

Sehingga dengan demikian data yang dipilih sebagai data pada tugas akhir ini adalah data-data yang dihasilkan dari perhitungan dengan menggunakan regresi polynomial orde tiga.

4.1.2. Interpolasi

Interpolasi yang digunakan pada penelitian ini adalah interpolasi orde tiga, ini dikarenakan data awal yang diketahui berjumlah empat data. Pada dasarnya interpolasi digunakan untuk data-data yang sifatnya sudah presisi / tepat. Namun pada tugas akhir ini interpolasi dilakukan untuk membandingkan hasil data besarnya laju korosi yang akan diperoleh dari perhitungan interpolasi ini apakah lebih baik (mendekati data awal) jika dibandingkan dengan hasil yang diperoleh dari perhitungan regresi polynomial orde tiga sebelumnya.

Dengan menggunakan persamaan-persamaan yang telah dijelaskan pada Bab.2 maka selanjutnya diperoleh besarnya laju korosi untuk masing-masing suhu pemanasan.

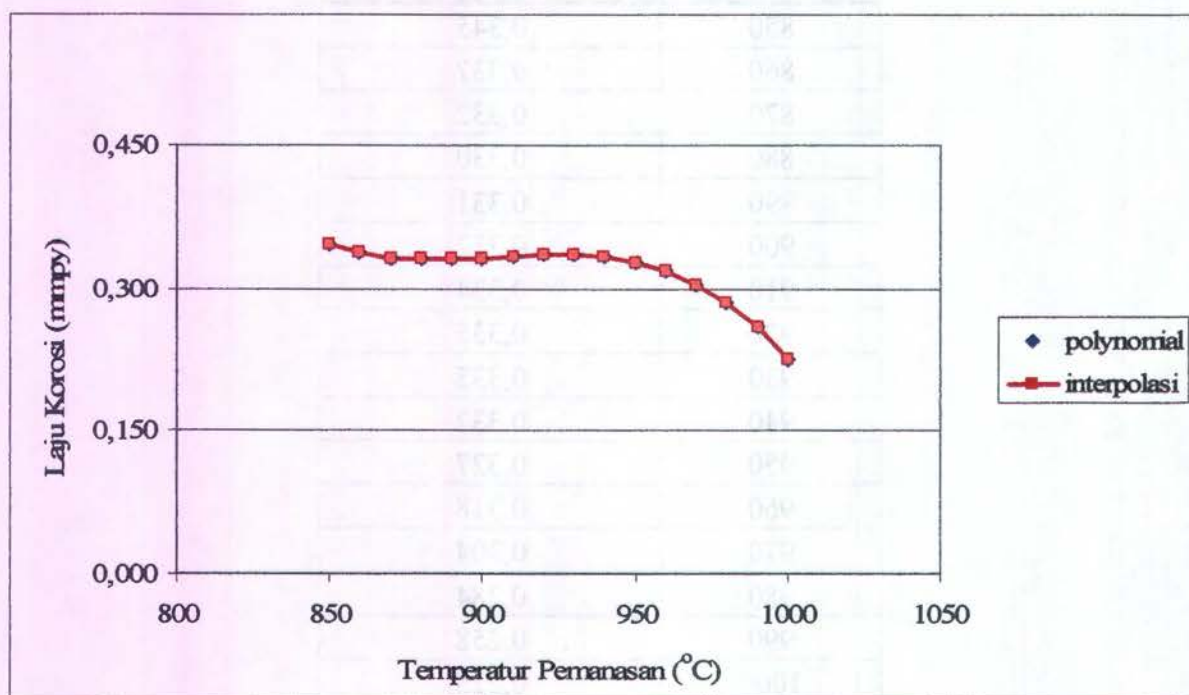
Tabel 4.2. Laju korosi yang didapatkan dari perhitungan interpolasi.

SUHU (°C)	LAJU KOROSI
	(mmpy)
850	0,345
860	0,337
870	0,332
880	0,330
890	0,331
900	0,332
910	0,334
920	0,335
930	0,335
940	0,332
950	0,327
960	0,318
970	0,304
980	0,284
990	0,258
1000	0,225

Dari tabel 4.2. di atas kemudian dibandingkan dengan hasil yang didapatkan dari perhitungan polynomial, akan didapat tabel perbandingan seperti di bawah ini :

Tabel 4.3. Perbandingan laju korosi dari perhitungan interpolasi dan polynomial.

SUHU (°C)	LAJU KOROSI (mmpy)	
	POLYNOMIAL	INTERPOLASI
850	0,345	0,345
860	0,337	0,337
870	0,332	0,332
880	0,330	0,330
890	0,331	0,331
900	0,332	0,332
910	0,334	0,334
920	0,335	0,335
930	0,335	0,335
940	0,332	0,332
950	0,327	0,327
960	0,318	0,318
970	0,304	0,304
980	0,284	0,284
990	0,258	0,258
1000	0,225	0,225



Gambar 4.2. Grafik hubungan antara temperatur pemanasan dan laju korosi dengan regresi polynomial dan interpolasi.

Jika diperhatikan pada gambar 4.2. di atas, hasil perhitungan antara regresi polynomial dan interpolasi ini menghasilkan data yang sama, hal ini bisa terjadi karena data-data yang didapatkan dari percobaan bukan merupakan data presisi. Data-data tersebut merupakan nilai rata-rata dari percobaan yang dilakukan sebanyak tiga kali untuk setiap suhu pemanasan.

Sehingga data-data seperti yang tercantum pada tabel 4.3 di atas selanjutnya akan disebut sebagai *data utama* dalam penelitian ini.

4.2. Analisa Pengaruh Perubahan Laju Pelatihan (α) terhadap Perilaku Jaringan

Learning rate merupakan suatu nilai yang berfungsi untuk menentukan seberapa cepat proses pelatihan pada suatu jaringan berlangsung. Berikut ini akan dianalisa mengenai pengaruh perubahan learning rate terhadap perilaku jaringan. Model jaringan akan dikenai perlakuan yang sama, baik dalam hal penggunaan data pelatihan maupun besarnya momentum yang digunakan. Hanya saja besar dari nilai learning rate akan divariasasi sampai nilai yang terbesar adalah 0.5, ini bertujuan agar masing-masing model jaringan menunjukkan hasil perbedaan dalam pembelajarannya.

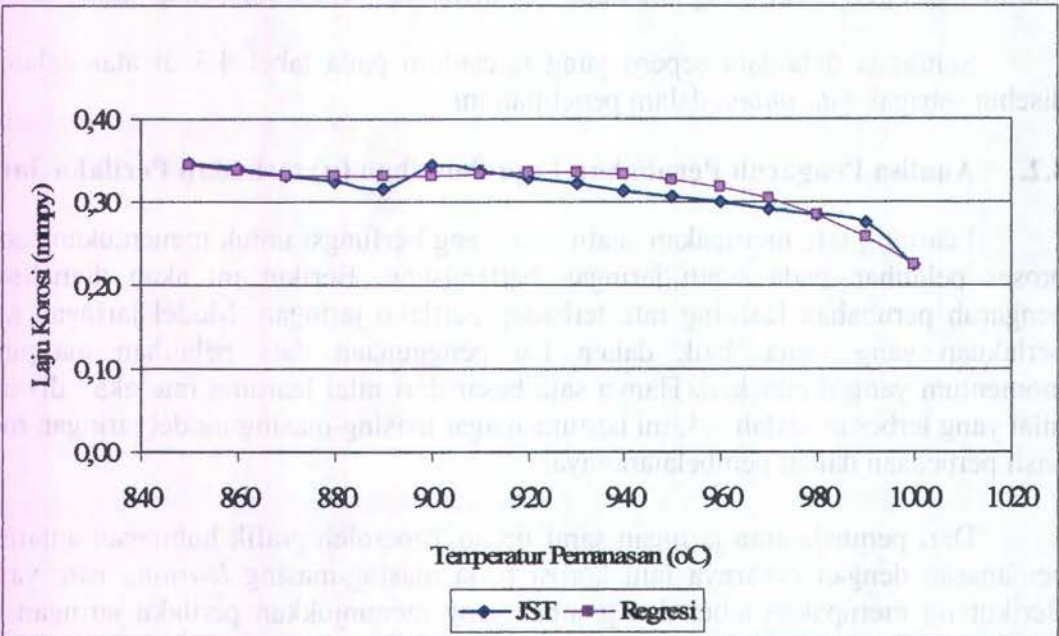
Dari pembelajaran jaringan saraf tiruan, diperoleh grafik hubungan antara temperatur pemanasan dengan besarnya laju korosi pada masing-masing *learning rate* yang berbeda. Berikut ini merupakan tabel dan gambar yang menunjukkan perilaku jaringan saraf tiruan setelah dilakukan proses pembelajaran dengan menggunakan nilai laju pelatihan antara 0,1 sampai dengan 0,5.

4.2.1. Pelatihan jaringan menggunakan laju pelatihan 0,1 dan momentum 0,1.

Tabel 4.4. Jaringan dilatih dengan laju pelatihan 0,1 dan momentum 0,1.

Suhu (°C)	Laju Korosi (mmpy)		Error	(%)
	Regresi	JST		
850	0,345	0,347757	-0,002757	99,207331
860	0,337	0,339912	-0,003192	99,061051
870	0,332	0,332041	0,000079	99,976249
880	0,330	0,324152	0,006208	98,120934
890	0,331	0,316253	0,014347	95,660217
900	0,332	0,345117	-0,013117	96,199257
910	0,334	0,337464	-0,003744	98,890518
920	0,335	0,329784	0,005136	98,466508
930	0,335	0,322084	0,012676	96,213259
940	0,332	0,314370	0,018030	94,575667
950	0,327	0,306649	0,020351	93,776545
960	0,318	0,298930	0,018790	94,086089
970	0,304	0,291220	0,012500	95,884442
980	0,284	0,283527	0,000633	99,777183
990	0,258	0,275858	-0,017658	93,598838
1000	0,225	0,224998	0,000002	99,999260

Untuk melihat lebih jelas dari hasil perhitungan jaringan di atas, dari tabel 4.4 di atas maka dibuat grafik seperti yang terlihat pada gambar berikut ini.

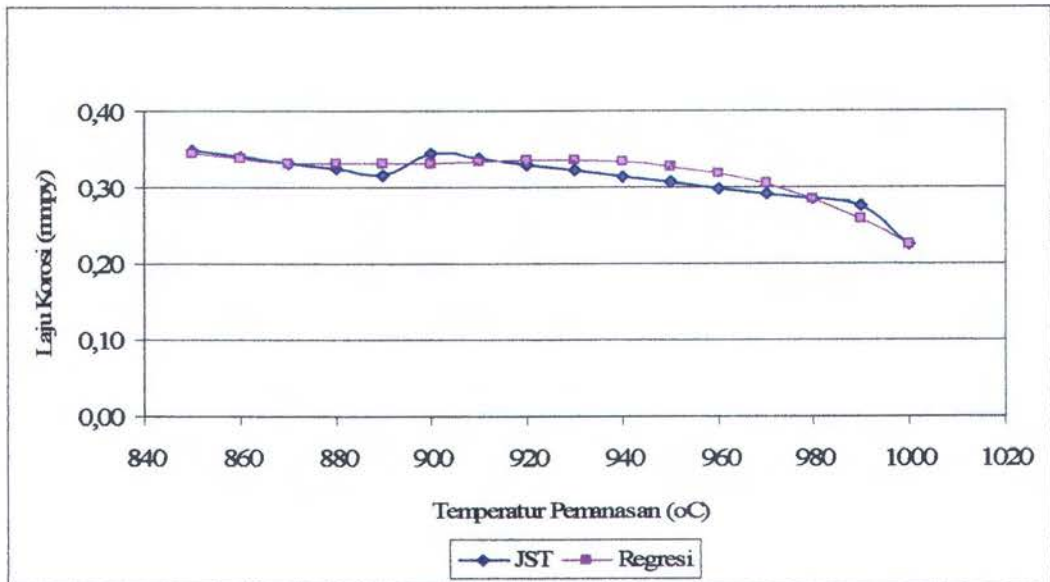


Gambar 4.3. Grafik perbandingan antara data asli dan data hasil perhitungan jaringan dengan laju pelatihan 0,1.

4.2.2. Pelatihan jaringan menggunakan laju pelatihan 0,2 dan momentum 0,1.

Tabel 4.5. Jaringan dilatih dengan laju pelatihan 0,2 dan momentum 0,1.

Suhu (°C)	Laju Korosi (mmpy)		Error	(%)
	Regresi	JST		
850	0,345	0,347843	-0,002843	99,182808
860	0,337	0,339964	-0,003244	99,045673
870	0,332	0,332059	0,000061	99,981679
880	0,330	0,324134	0,006226	98,115469
890	0,331	0,316197	0,014403	95,643486
900	0,332	0,345339	-0,013339	96,137404
910	0,334	0,337641	-0,003921	98,838652
920	0,335	0,329915	0,005005	98,505520
930	0,335	0,322166	0,012594	96,238042
940	0,332	0,314404	0,017996	94,585962
950	0,327	0,306634	0,020366	93,771850
960	0,318	0,298865	0,018855	94,065467
970	0,304	0,291104	0,012616	95,846235
980	0,284	0,283360	0,000800	99,718530
990	0,258	0,275641	-0,017441	93,672576
1000	0,225	0,224991	0,000009	99,996019

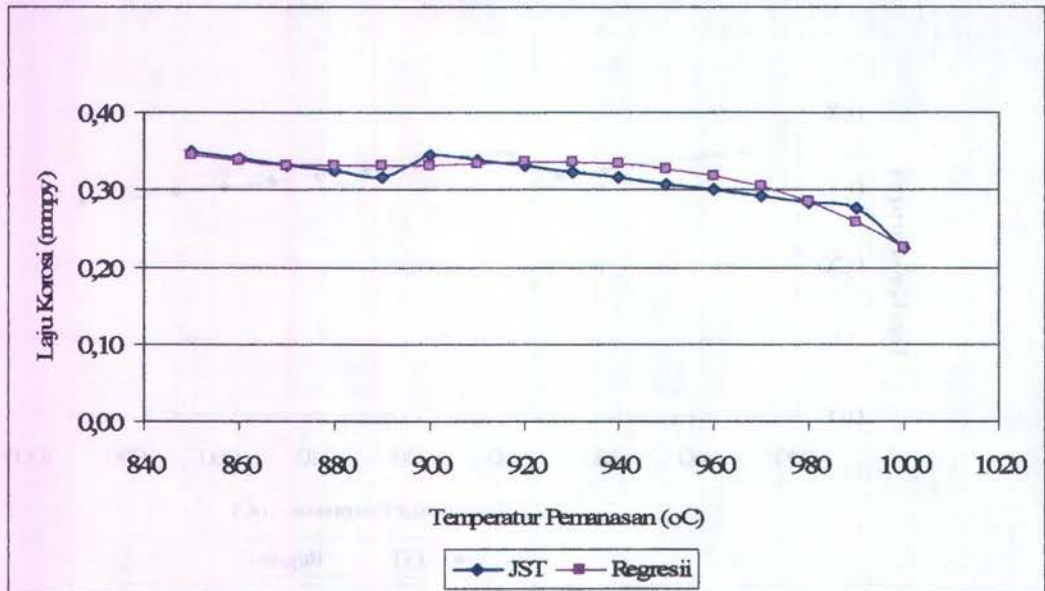


Gambar 4.4. Grafik perbandingan antara data asli dan data hasil perhitungan jaringan dengan laju pelatihan 0,2.

4.2.3. Pelatihan jaringan menggunakan laju pelatihan 0,3 dan momentum 0,1.

Tabel 4.6. Jaringan dilatih dengan laju pelatihan 0,3 dan momentum 0,1.

Suhu (°C)	Laju Korosi (mmpy)		Error	(%)
	Regresi	JST		
850	0,345	0,347932	-0,002932	99,157331
860	0,337	0,340146	-0,003426	98,992752
870	0,332	0,332213	-0,000093	99,971928
880	0,330	0,324142	0,006218	98,117724
890	0,331	0,315941	0,014659	95,565953
900	0,332	0,345124	-0,013124	96,197222
910	0,334	0,337825	-0,004105	98,784898
920	0,335	0,330378	0,004542	98,643714
930	0,335	0,322789	0,011971	96,423887
940	0,332	0,315066	0,017334	94,785106
950	0,327	0,307217	0,019783	93,950292
960	0,318	0,299254	0,018466	94,187865
970	0,304	0,291185	0,012535	95,872941
980	0,284	0,283024	0,001136	99,600346
990	0,258	0,274784	-0,016584	99,976249
1000	0,225	0,224983	0,000017	99,992371

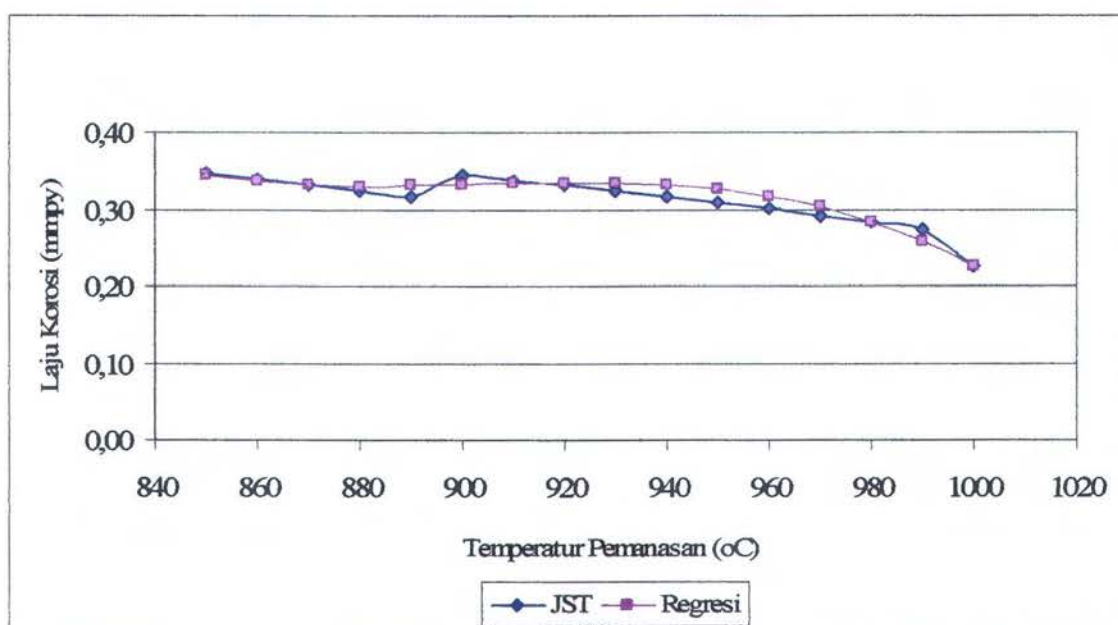


Gambar 4.5. Grafik perbandingan antara data asli dan data hasil perhitungan jaringan dengan laju pelatihan 0,3.

4.2.4. Pelatihan jaringan menggunakan laju pelatihan 0,4 dan momentum 0,1.

Tabel 4.7. Jaringan dilatih dengan laju pelatihan 0,4 dan momentum 0,1.

Suhu (°C)	Laju Korosi (mmpy)		Error	(%)
	Regresi	JST		
850	0,345	0,348067	-0,003067	99,118985
860	0,337	0,340463	-0,003743	98,900677
870	0,332	0,332502	-0,000382	99,885180
880	0,330	0,324189	0,006171	98,131959
890	0,331	0,315532	0,015068	95,442367
900	0,332	0,344532	-0,012532	96,362704
910	0,334	0,337955	-0,004235	98,746934
920	0,335	0,331040	0,003880	98,841399
930	0,335	0,323786	0,010974	96,721710
940	0,332	0,316195	0,016205	95,124820
950	0,327	0,308273	0,018727	94,272993
960	0,318	0,300027	0,017693	94,431392
970	0,304	0,291471	0,012249	95,967065
980	0,284	0,282620	0,001540	99,458009
990	0,258	0,273494	-0,015294	94,408078
1000	0,225	0,224966	0,000034	99,985012

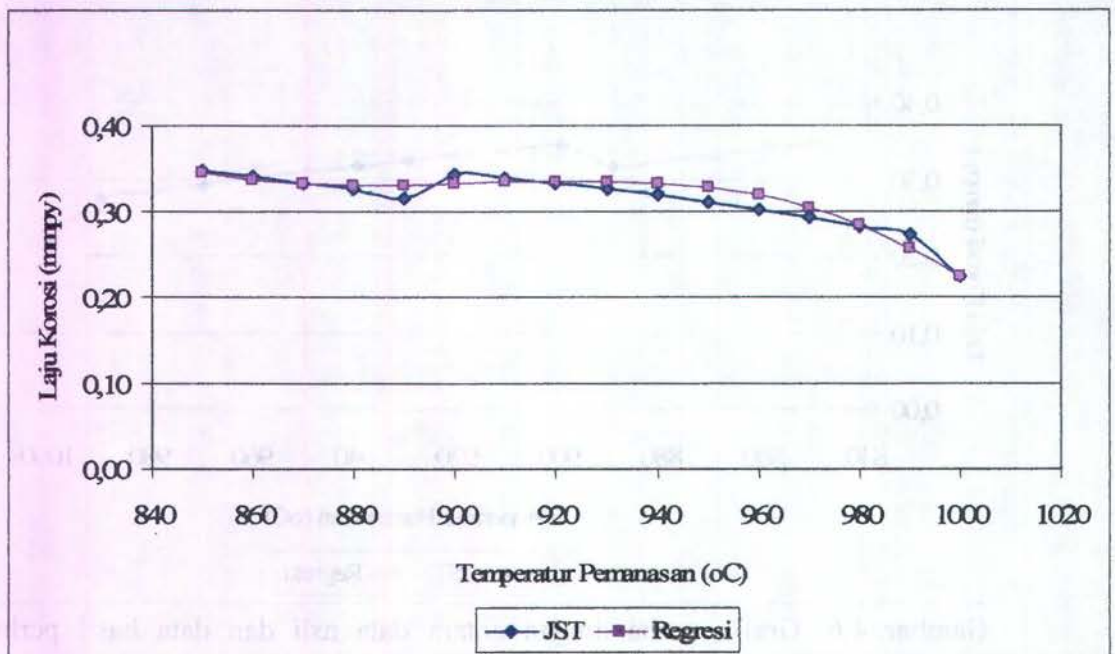


Gambar 4.6. Grafik perbandingan antara data asli dan data hasil perhitungan jaringan dengan laju pelatihan 0,4.

4.2.5. Pelatihan jaringan menggunakan laju pelatihan 0,5 dan momentum 0,1.

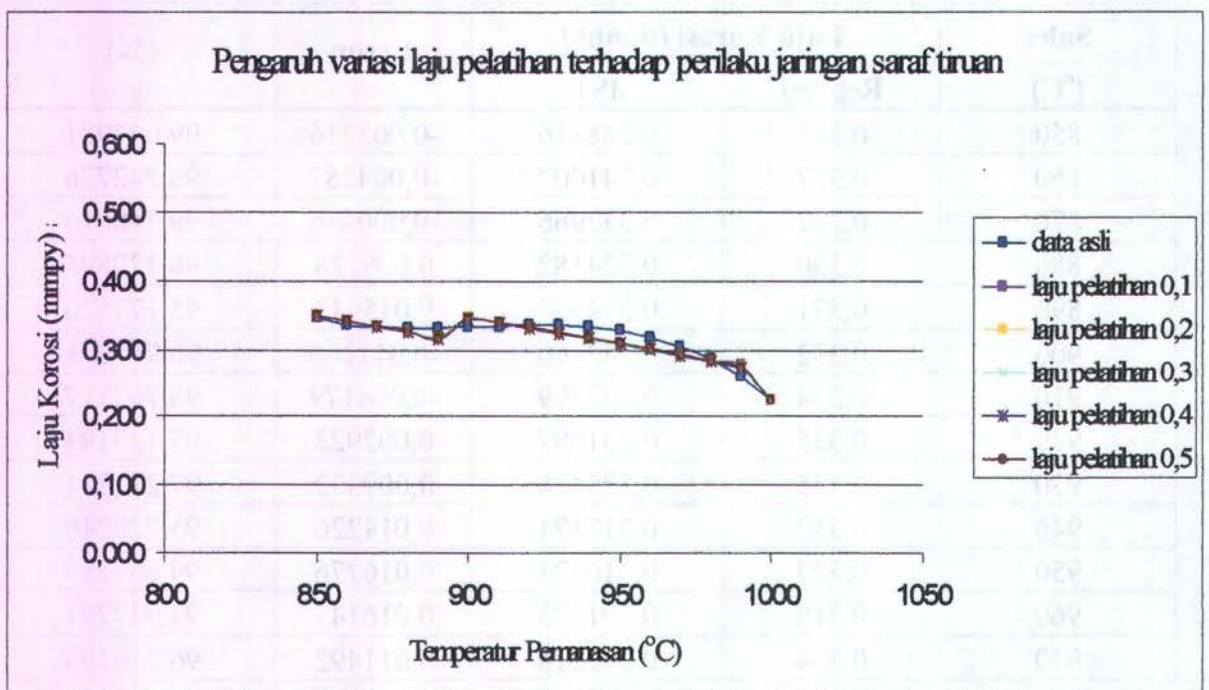
Tabel 4.8. Jaringan dilatih dengan laju pelatihan 0,5 dan momentum 0,1.

Suhu (°C)	Laju Korosi (mmpy)		Error	(%)
	Regresi	JST		
850	0,345	0,348316	-0,003316	99,047981
860	0,337	0,341007	-0,004287	98,742726
870	0,332	0,332966	-0,000846	99,746016
880	0,330	0,324182	0,006178	98,129895
890	0,331	0,314657	0,015943	95,177520
900	0,332	0,343160	-0,011160	96,747971
910	0,334	0,337899	-0,004179	98,763117
920	0,335	0,331997	0,002923	99,127194
930	0,335	0,325428	0,009332	97,212301
940	0,332	0,318174	0,014226	95,720246
950	0,327	0,310224	0,016776	94,869589
960	0,318	0,301573	0,016147	94,917791
970	0,304	0,292228	0,011492	96,216398
980	0,284	0,282209	0,001951	99,313328
990	0,258	0,271545	-0,013345	95,085455
1000	0,225	0,224929	0,000071	99,968498



Gambar 4.7. Grafik perbandingan antara data asli dan data hasil perhitungan jaringan dengan laju pelatihan 0,5.

Dari kelima grafik di atas selanjutnya dari masing-masing grafik dijadikan satu untuk mengetahui grafik mana yang hasilnya paling mendekati grafik data asli.



Gambar 4.8. Grafik pengaruh variasi learning rate terhadap perilaku masing-masing model jaringan saraf tiruan.

Tabel 4.9. Rekapitulasi error masing-masing model jaringan

$\eta = 0,1$	
α	Keakuratan (%)
0,1	97,056
0,2	97,084
0,3	97,514
0,4	97,237
0,5	97,424

Dengan melihat gambar 4.8. dan tabel 4.9. di atas, dapat didapatkan data perbandingan yang diperoleh dari pelatihan model jaringan dengan menggunakan laju pelatihan yang berbeda-beda. Data-data hasil pelatihan yang hasilnya paling mendekati data grafik data asli adalah data-data yang dihasilkan oleh model jaringan saraf tiruan yang dilatih dengan menggunakan laju pelatihan 0,3 dan momentum sebesar 0,1.

Jika diperhatikan, hampir semua model jaringan menghasilkan prosentase keakuratan yang dapat dikatakan mirip semua, hanya berbeda angka yang berada di belakang koma saja.

Dengan memperhatikan besarnya nilai error yang dihasilkan oleh masing-masing hasil pembelajaran jaringan, maka dapat ditarik kesimpulan bahwa *learning rate* sangat mempengaruhi perilaku jaringan dalam hal mengolah data. Namun demikian, belum ada suatu literatur yang menyebutkan bahwa besar kecilnya *learning rate* ini akan berbanding terbalik dengan *error* yang dihasilkan. Dalam artian jika *learning rate* yang digunakan semakin besar maka besarnya *error* yang dihasilkan semakin kecil, dan jika *laerning rate* yang digunakan kecil maka *error* yang dihasilkan oleh model jaringan saraf tiruan semakin membesar. Sehingga dalam hal ini sangat penting untuk tidak memasukkan sembarang *learning rate* ke dalam suatu proses pelatihan jaringan saraf tiruan.

4.3. Analisa Pengaruh Perubahan Monentum (η) terhadap Perilaku Jaringan

Pada bab tinjauan pustaka sudah dijelaskan bahwa momentum ditambahkan pada persamaan bobot dan bias, yang mana bobot saat ini adalah merupakan fungsi dari bobot sebelumnya.

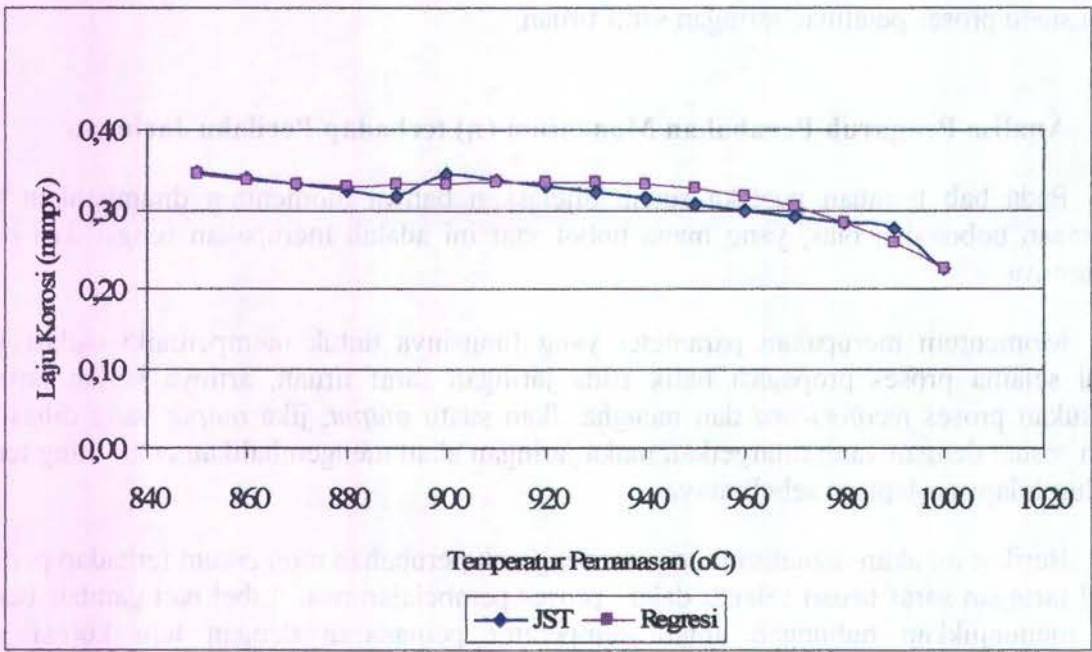
Momentum merupakan parameter yang fungsinya untuk memperbaiki osilasi yang terjadi selama proses propagasi balik pada jaringan saraf tiruan, artinya ketika jaringan melakukan proses *feedforward* dan menghasilkan suatu *output*, jika *output* yang dihasilkan belum sesuai dengan yang ditargetkan maka jaringan akan mengembalikan *error* yang terjadi ke seluruh lapisan-lapisan sebelumnya.

Berikut ini akan dianalisa mengenai pengaruh perubahan momentum terhadap perilaku model jaringan saraf tiruan selama dalam proses pembelajarannya. Tabel dan gambar berikut akan menunjukkan hubungan antara temperatur pemanasan dengan laju korosi yang dihasilkan dari masing-masing pembelajaran model jaringan saraf tiruan.

4.3.1. Pelatihan jaringan menggunakan laju pelatihan 0,1 dan momentum 0,1.

Tabel 4.10. Jaringan dilatih dengan laju pelatihan 0,1 dan momentum 0,1.

Suhu (°C)	Laju Korosi (mmpy)		Error	(%)
	Regresi	JST		
850	0,345	0,347767	-0,002780	99,204253
860	0,337	0,339829	-0,003115	99,083431
870	0,332	0,331933	0,000189	99,943001
880	0,330	0,324085	0,006286	98,097282
890	0,331	0,316291	0,014329	95,665846
900	0,332	0,345319	-0,013330	96,139975
910	0,334	0,337424	-0,003717	98,898351
920	0,335	0,329572	0,005333	98,407704
930	0,335	0,321769	0,012975	96,124066
940	0,332	0,314020	0,018363	94,475572
950	0,327	0,306332	0,020651	93,684567
960	0,318	0,298710	0,018994	94,021718
970	0,304	0,291160	0,012546	95,869277
980	0,284	0,283686	0,000461	99,837773
990	0,258	0,276295	-0,018106	93,447266
1000	0,225	0,224998	0,000009	99,995984

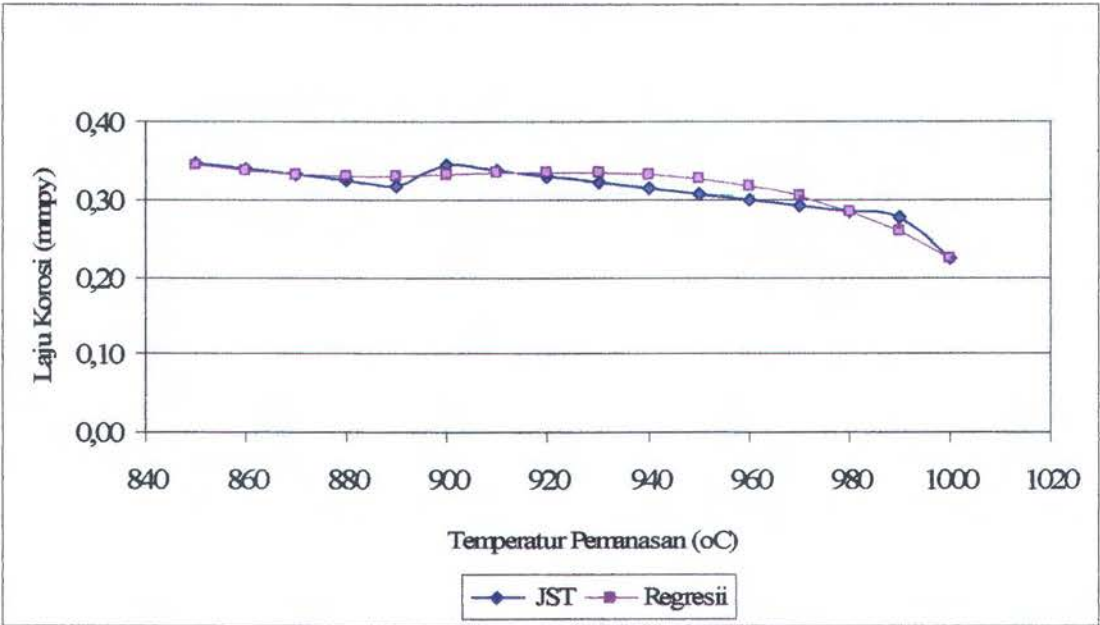


Gambar 4.9. Grafik perbandingan antara data asli dan data hasil perhitungan jaringan dengan momentum 0,1.

4.3.2. Pelatihan jaringan menggunakan laju pelatihan 0,1 dan momentum 0,2.

Tabel 4.11. Jaringan dilatih dengan laju pelatihan 0,1 dan momentum 0,2.

Suhu (°C)	Laju Korosi (mmpy)		Error	(%)
	Regresi	JST		
850	0,345	0,347780	-0,002780	99,200508
860	0,337	0,339835	-0,003115	99,083431
870	0,332	0,331931	0,000189	99,943001
880	0,330	0,324074	0,006286	98,097282
890	0,331	0,316271	0,014329	95,665846
900	0,332	0,345330	-0,013330	96,139975
910	0,334	0,337437	-0,003717	98,898351
920	0,335	0,329587	0,005333	98,407704
930	0,335	0,321785	0,012975	96,124066
940	0,332	0,314037	0,018363	94,475572
950	0,327	0,306349	0,020651	93,684567
960	0,318	0,298726	0,018994	94,021718
970	0,304	0,291174	0,012546	95,869277
980	0,284	0,283699	0,000461	99,837773
990	0,258	0,276306	-0,018106	93,447266
1000	0,225	0,224991	0,000009	99,995984

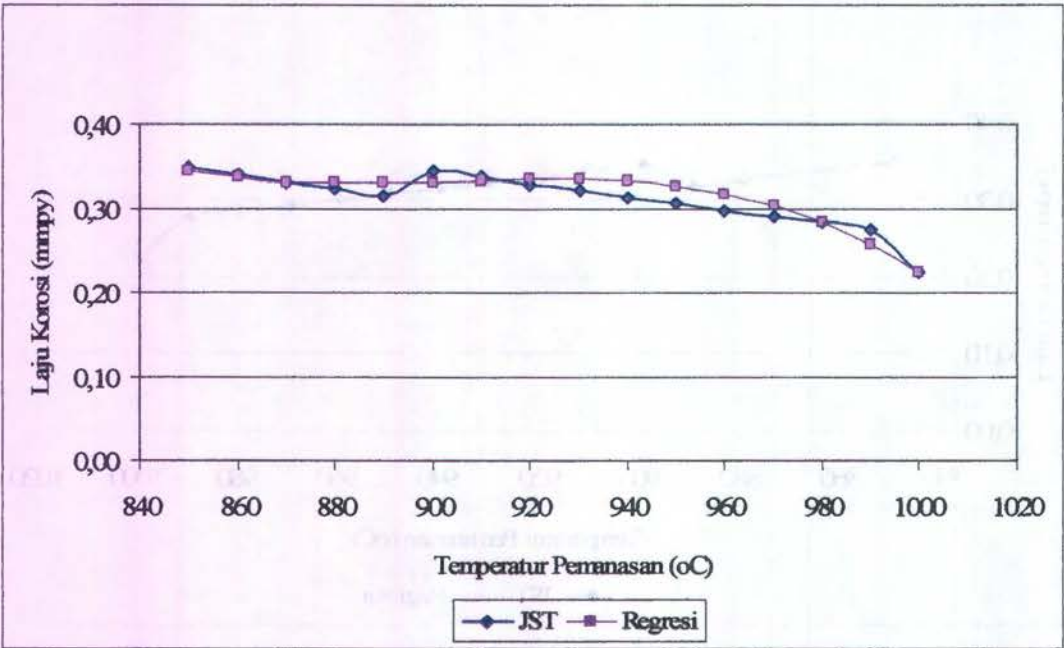


Gambar 4.10. Grafik perbandingan antara data asli dan data hasil perhitungan jaringan dengan momentum 0,2.

4.3.3. Pelatihan jaringan menggunakan laju pelatihan 0,1 dan momentum 0,3.

Tabel 4.12. Jaringan dilatih dengan laju pelatihan 0,1 dan momentum 0,3.

Suhu (°C)	Laju Korosi (mmpy)		Error	(%)
	Regresi	JST		
850	0,345	0,347798	-0,002798	99,195586
860	0,337	0,339842	-0,003122	99,081251
870	0,332	0,331929	0,000191	99,942358
880	0,330	0,324063	0,006297	98,093798
890	0,331	0,316251	0,014349	95,659619
900	0,332	0,345345	-0,013345	96,135766
910	0,334	0,337454	-0,003734	98,893465
920	0,335	0,329605	0,005315	98,413071
930	0,335	0,321804	0,012956	96,129765
940	0,332	0,314057	0,018343	94,481598
950	0,327	0,306369	0,020631	93,690952
960	0,318	0,298747	0,018973	94,028539
970	0,304	0,291197	0,012523	95,876667
980	0,284	0,283722	0,000438	99,845954
990	0,258	0,276330	-0,018130	93,439111
1000	0,225	0,224986	0,000014	99,993850

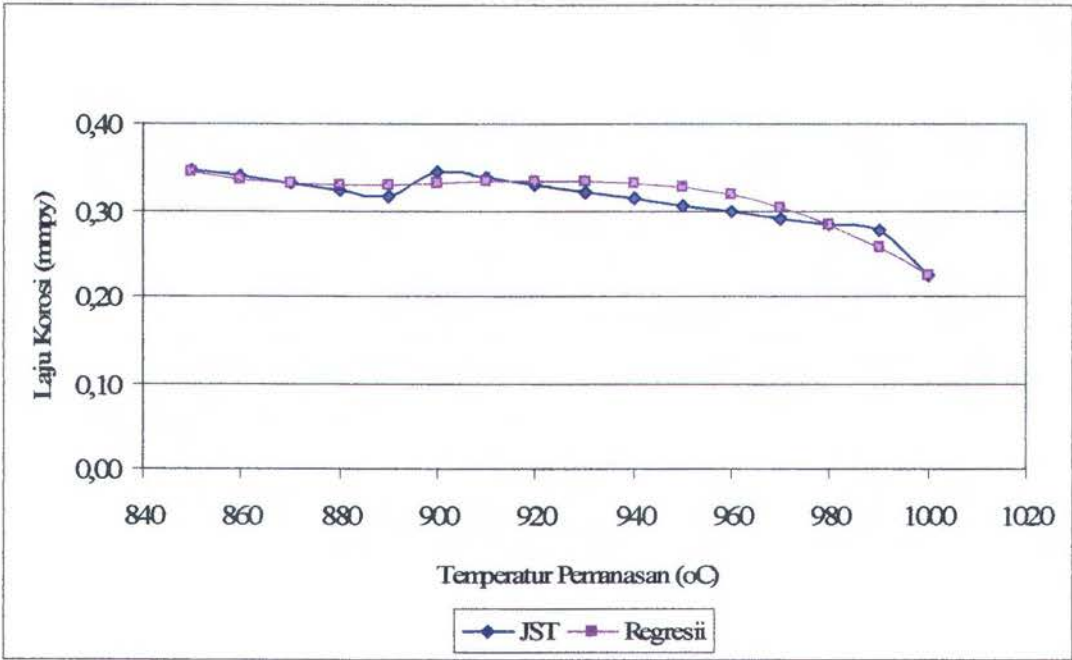


Gambar 4.11. Grafik perbandingan antara data asli dan data hasil perhitungan jaringan dengan momentum 0,3.

4.3.4. Pelatihan jaringan menggunakan laju pelatihan 0,1 dan momentum 0,4.

Tabel 4.13. Jaringan dilatih dengan laju pelatihan 0,1 dan momentum 0,4.

Suhu (°C)	Laju Korosi (mmpy)		Error	(%)
	Regresi	JST		
850	0,345	0,347816	0,005150	99,190302
860	0,337	0,339850	0,004794	99,078913
870	0,332	0,331926	0,008069	99,941723
880	0,330	0,324051	0,014130	98,090224
890	0,331	0,316230	-0,014764	95,653265
900	0,332	0,345364	-0,005475	96,130490
910	0,334	0,337475	0,004092	98,887326
920	0,335	0,329628	0,013092	98,419861
930	0,335	0,321828	0,020703	96,137047
940	0,332	0,314057	0,026031	94,481598
950	0,327	0,306369	0,028253	93,690952
960	0,318	0,298747	0,026523	94,028539
970	0,304	0,291197	0,019998	95,876667
980	0,284	0,283722	0,007830	99,845954
990	0,258	0,276330	0,033214	93,439111
1000	0,225	0,224986	0,000014	99,993850

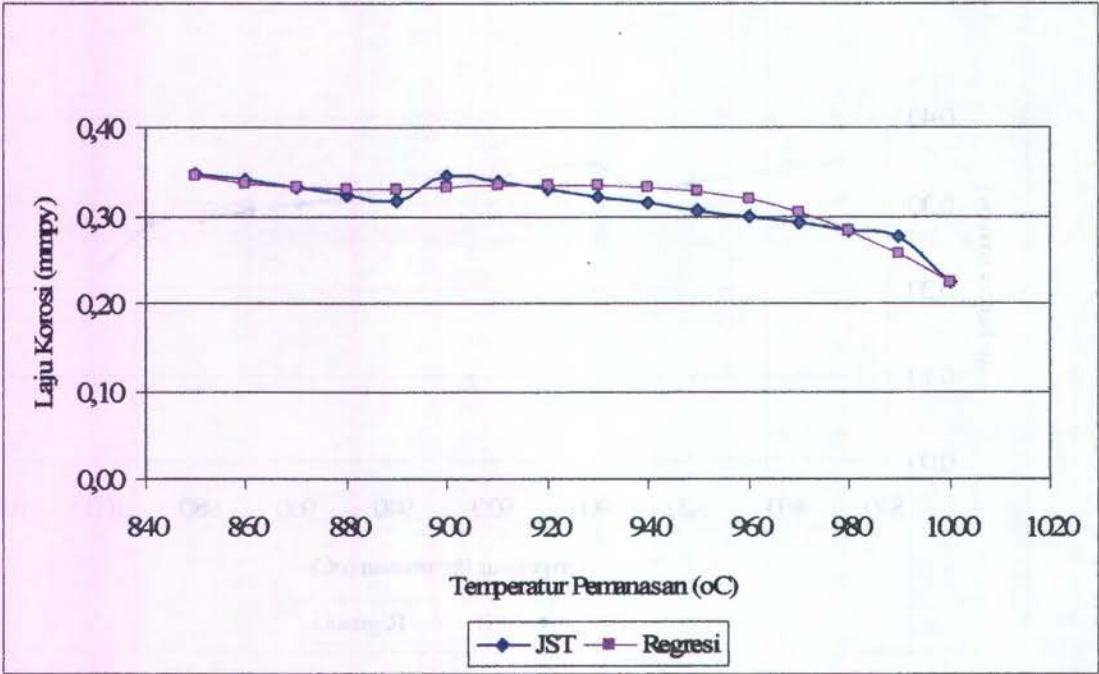


Gambar 4.12. Grafik perbandingan antara data asli dan data hasil perhitungan jaringan dengan momentum 0,4.

4.3.5. Pelatihan jaringan menggunakan laju pelatihan 0,1 dan momentum 0,5.

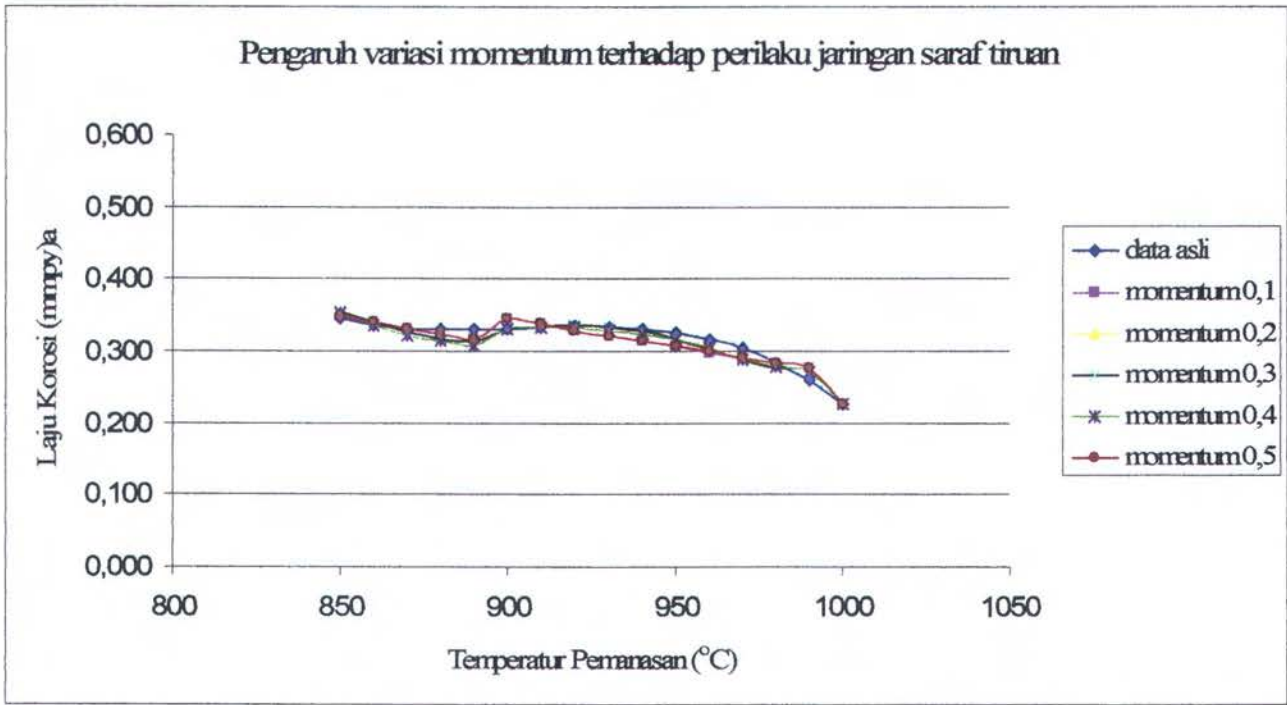
Tabel 4.14. Jaringan dilatih dengan laju pelatihan 0,1 dan momentum 0,5.

Suhu (°C)	Laju Korosi (mmpy)		Error	(%)
	Regresi	JST		
850	0,345	0,347838	-0,002838	99,184227
860	0,337	0,339858	-0,003138	99,076775
870	0,332	0,331919	0,000201	99,939605
880	0,330	0,324029	0,006331	98,083655
890	0,331	0,316193	0,014407	95,642202
900	0,332	0,345378	-0,013378	96,126556
910	0,334	0,337497	-0,003777	98,881011
920	0,335	0,329655	0,005265	98,428109
930	0,335	0,321861	0,012899	96,146729
940	0,332	0,314119	0,018281	94,500219
950	0,327	0,306435	0,020565	93,711071
960	0,318	0,298816	0,018904	94,050126
970	0,304	0,291267	0,012453	95,899845
980	0,284	0,283794	0,000366	99,871052
990	0,258	0,276401	-0,018201	93,414956
1000	0,225	0,224986	0,000014	99,993740



Gambar 4.13. Grafik perbandingan antara data asli dan data hasil perhitungan jaringan dengan momentum 0,5.

Dari kelima grafik di atas selanjutnya dari masing-masing grafik dijadikan satu untuk mengetahui grafik mana yang hasilnya paling mendekati grafik data asli.



Gambar 4.14. Grafik pengaruh variasi momentum terhadap perilaku jaringan saraf tiruan.

Dari tabel 4.11 sampai 4.15 dan dari gambar 4.9 sampai dengan gambar 4.13 didapat besarnya persentase keakuratan perhitungan sebagai berikut :

Tabel 4.15. Rekapitulasi error masing-masing model jaringan

$\alpha = 0,1$	
η	Keakuratan (%)
0,1	97,0560
0,2	97,055
0,3	97,0563
0,4	97,0554
0,5	97,0593

Setelah melakukan pelatihan dan pengujian dari keseluruhan variasi laju pelatihan dan momentum pelatihan, maka didapatkan satu model jaringan yang hasil errornya paling kecil atau dengan kata lain jaringan yang mempunyai keakuratan penghitungan sebesar 97,514%. Model jaringan saraf tiruan tersebut yaitu model jaringan yang di latih dengan laju pelatihan 0,3 dan momentum 0,1.

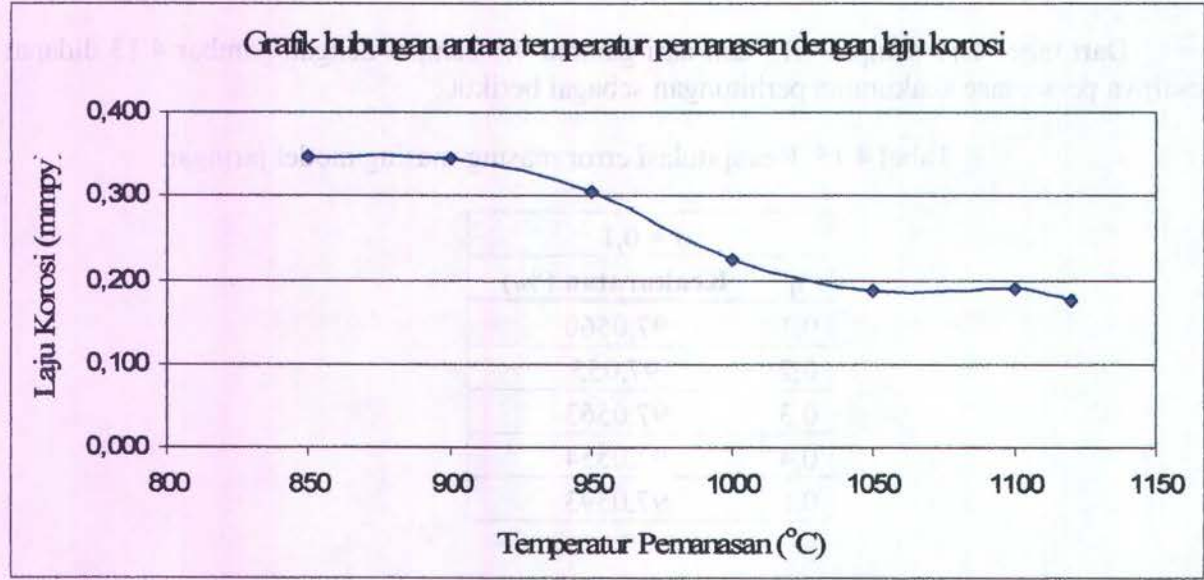
4.4. Analisa Laju Korosi dengan Jaringan Saraf Tiruan

Dengan melihat besarnya error yang dihasilkan oleh masing-masing model jaringan saraf tiruan (JST), model jaringan terbaik yang diperoleh dapat dikatakan bahwa jaringan saraf tiruan yang dibentuk sudah mampu mempelajari pola data yang diinputkan dan dapat digunakan sebagai "tool" untuk menganalisa besarnya laju korosi yang terjadi jika suhu pemanasan dinaikkan sampai suhu pemanasan tertentu. Suhu maksimum yang akan dicari besarnya laju korosi adalah 1120°C.

Dari pelatihan model jaringan saraf tiruan terpilih didapatkan hubungan temperatur pemanasan dan besarnya laju korosi seperti yang terdapat dalam tabel 4.15 dan gambar 4.15 berikut ini.

Tabel 4.16. Laju korosi yang didapatkan dari JST.

Suhu Pemanasan (°C)	Laju Korosi (mmpy)
850	0,348
900	0,345
950	0,306
1000	0,225
1050	0,187
1100	0,189
1120	0,175



Gambar 4.15. Grafik hubungan antara temperatur pemanasan dan laju korosi yang dihitung dengan menggunakan JST.

Dari grafik 4.15 di atas telah didapatkan bahwa laju korosi yang terbentuk pada proses perlakuan pemanasan pasca pengelasan yang terjadi pada baja tahan karat AISI 316, yaitu

semakin naik suhu pemanasan yang digunakan untuk perlakuan panas, maka laju korosi yang ditimbulkan adalah semakin menurun. Pada pemanasan tertinggi yaitu suhu 1120°C didapatkan nilai laju korosi yang terjadi paling minimum sebesar 0,175 mmpy.

Adanya perubahan laju korosi ini disebabkan karena adanya pengaruh presipitasi karbida krom. Presipitasi karbida krom yaitu terjadinya pengendapan karbida krom pada baja tahan karat AISI 316 ketika dikenai proses pengelasan. Presipitasi karbida krom terjadi ketika baja tahan karat mengalami proses pemanasan (pengelasan) sampai suhu antara 510°C sampai 788°C sehingga material menjadi sensitif terhadap timbulnya intergranular corrosion.

Akibat dari presipitasi karbida krom adalah akan menimbulkan intergranular corrosion. Intergranular corrosion terjadi jika krom keluar dari larutan, dan daerah yang berdekatan dengan batas butir akan kekurangan krom. Karbida krom pada daerah batas butir tidak mengalami korosi. Korosi akan menyerang pada daerah yang dekat dengan batas butir, karena pada daerah ini telah kehilangan krom sehingga daya tahan terhadap korosi menurun.

Seperti yang telah diketahui ketahanan terhadap timbulnya korosi pada baja tahan karat diperoleh dari unsur kromnya, dimana untuk baja tahan karat AISI 316 memiliki unsur krom sebesar 16-18 %. Sehingga dengan adanya perlakuan panas pasca pengelasan ini karbida krom yang pada mulanya mengendap pada satu daerah tertentu (batas butir) maka akan larut kembali. Dengan semakin bertambahnya suhu pemanasan pada proses perlakuan panas, maka panas yang diterima oleh material semakin merata dan endapan karbida krom yang ada akan kembali larut menjadi austenit yang homogen, sehingga laju korosi yang terjadi semakin melambat.

Hasil perhitungan besarnya laju korosi yang dihasilkan oleh masing-masing suhu pemanasan dengan jaringan saraf tiruan ini sesuai dengan penelitian yang telah dilakukan oleh Sulistianto (2007), yaitu dengan semakin bertambahnya panas yang diberikan pada material baja tahan karat AISI 316 yang telah dikenai perlakuan pemanasan (pengelasan) maka karbida krom yang mengendap pada batas butir akan larut kembali dan menyebabkan ketahanan baja tahan karat AISI 316 terhadap korosi menjadi lebih baik.

Hasil perhitungan dengan menggunakan jaringan saraf tiruan ini sesuai dengan apa yang dinyatakan oleh Jones (1996). Jones (1996) menyatakan bahwa suhu pemanasan pada proses perlakuan panas pasca pengelasan yang berada pada kisaran 1120°C akan menyebabkan semua karbida krom yang mengendap pada batas butir larut kembali. Dengan asumsi kembali larutnya karbida krom ini, maka ketahanan material baja tahan karat AISI 316 terhadap timbulnya korosi akan semakin besar, sehingga laju korosi yang ditimbulkan akan semakin menjadi kecil.

Bab 5. Kesimpulan dan Saran

5.1. Kesimpulan

Setelah melakukan penelitian mengenai peramalan besarnya laju korosi dengan penggunaan *artificial neural network* (JST) dan melakukan analisa dari hasil error yang dihasilkan oleh jaringan maka dapat diperoleh beberapa kesimpulan berikut ini :

1. Prosedur pembentukan model jaringan saraf tiruan dilakukan dengan menggunakan dua tahap yaitu tahap pembelajaran model dan tahap pengujian model. Pada tahap pembelajaran model, parameter yang diperlukan adalah meliputi pemberian bobot awal, laju pelatihan dan momentum. Ketepatan dalam menentukan parameter tersebut akan mempercepat proses pembelajaran model jaringan saraf tiruan. Setelah model melakukan tahap pembelajaran kemudian model melakukan tahap uji validitas. Dari hasil pengujian kemudian dilakukan pengukuran terhadap ketepatan klasifikasi model jaringan, sehingga diperoleh model jaringan yang paling baik.
2. Keakuratan yang dihasilkan oleh model JST mencapai 97,514%. Dengan memperhatikan persentase error yang dihasilkan dan dengan segala kelemahan yang dimiliki oleh model jaringan saraf tiruan, maka dapat disimpulkan bahwa model jaringan saraf tiruan ini sudah memiliki suatu pola yang baik dalam proses pembelajarannya.
3. Dari perhitungan laju korosi menggunakan model jaringan saraf tiruan ini diperoleh data-data yang menyebutkan bahwa temperatur pemanasan yang paling baik adalah 1120°C yaitu sebesar 0,175 mmpy. Sedangkan untuk suhu-suhu yang lain besarnya laju korosi yang dihasilkan antara lain : suhu 850°C memiliki laju korosi sebesar 0,345 mmpy, suhu 900°C memiliki laju korosi sebesar 0,332 mmpy, suhu 950°C memiliki laju korosi sebesar 0,327 mmpy, suhu 1000°C memiliki laju korosi sebesar 0,225 mmpy, suhu 1050°C memiliki laju korosi sebesar 0,187 mmpy, dan suhu 1100°C sebesar 0,189 mmpy.
4. Suhu pemanasan pasca proses pengelasan diambil paling besar 1120°C karena jika suhu dinaikkan lagi maka material akan mencapai titik lelehnya dan akan menyebabkan material baja tahan karat AISI 316 menjadi rusak.

5.2. Saran

Saran yang dapat diajukan agar percobaan berikutnya dapat lebih baik dan dapat menyempurnakan pembuatan model jaringan saraf tiruan yang telah dilakukan dalam tugas akhir ini adalah :

1. Data-data yang digunakan lebih bervariasi. Variasi dapat berupa berbagai macam jenis pengelasan, berbagai macam jenis elektroda, atau mungkin posisi pengelasan yang berbeda-beda.
2. Pemanfaatan model jaringan saraf tiruan terhadap permasalahan-permasalahan yang ada tidak hanya dengan pelatihan jaringan yang terawasi (*supervised learning*) saja, namun juga dengan memanfaatkan jaringan yang tanpa terawasi (*unsupervised learning*), dimana model jaringan dibentuk tanpa diketahui target dan outputnya terlebih dahulu.

DAFTAR PUSTAKA

- Chapra, Steven C dan Raymond P Canale, '*Numerical Methods for Engineers with Personal Computer Applications*', McGraw-Hill Book Company, New York, 1985.
- Desiani, Anita dan Muhammad Arhami, '*Konsep Kecerdasan Buatan*', Penerbit Andi, Yogyakarta, 2006.
- Fausett, Laurence, '*Fundamentals of Neural Networks, Architecture, Algorithms, And Applications*', Florida Institute of Technology, Prentice Hall International.Inc, 1994.
- Fontana, M. G, '*Corrosion Engineering*', McGraw-Hill Book Company, New York, 1987.
- Haykin, S, '*Neural Network a Comprehensive Foundation*', New Jersey : Prentice Hall, 1994.
- Hery Purnomo, Mauridhi, '*Dasar Algoritma Cerdas*'. PENS ITS, 2002.
- Jones, Denny A, '*Principle and Prevention of Corrosion*', New Jersey : Prentice Hall, 1996.
- Kusumadewi, Sri, '*Membangun Jaringan Syaraf Tiruan Menggunakan Matlab & Excel Link*', Graha Ilmu, Yogyakarta, 2004.
- Pandjaitan, Lanny W, '*Dasar-dasar Komputasi Cerdas*', Penerbit Andi, Yogyakarta, 2007.
- Peckner, D dan I. M. Bernstein, '*Hand Book of Stainless Steels*', McGraw-Hill Book Company, New York, 1977.
- Puspitaningrum, Diyah, '*Pengantar Jaringan Saraf Tiruan*', Penerbit Andi, Yogyakarta, 2006.
- Suherman, Wahid, '*Pengetahuan Bahan*', Jurusan Teknik Mesin, ITS, Surabaya, 1988.
- Totten, G.E, Bates, C.E dan Clinton, N.A, '*Quenchants and Quenching Technology*', ASM International, 1991.
- Trethewey, K. R dan J. Chamberlain, '*Korosi Untuk Mahasiswa dan Rekayasawan*', PT. Gramedia Pustaka Utama, Jakarta, 1991.
- Wiryosumarto, H dan T. Okumura, '*Teknologi Pengelasan Logam*', PT. Pradnya Paramita, Jakarta, 2000.
- Yani, Eli, 2005. "*Pengantar Jaringan Saraf Tiruan*".
<http://trirezqiariantoro.files.wordpress.com/2007/05/jaringan_syaraf_tiruan.pdf>

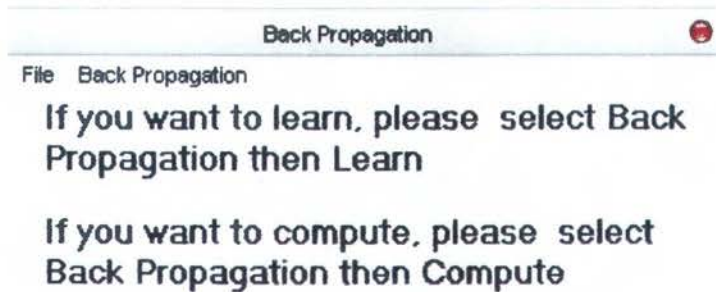
" Hari ini adalah hari esok yang membuat Anda optimis kemarin "

LAMPIRAN

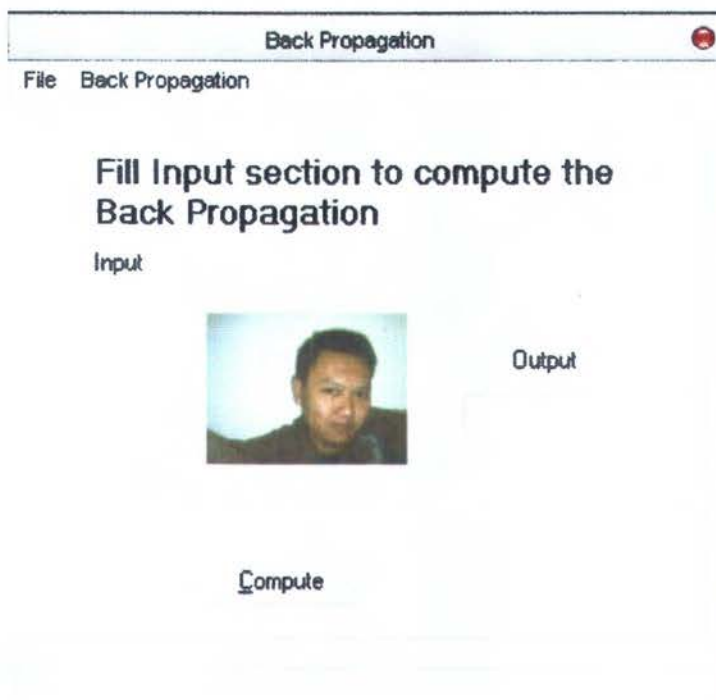
LAMPIRAN

⇒ Cara penggunaan program

1. Pastikan dikomputer sudah terinstal software Delphi 7
2. Aktifkan program delphi (berbentuk delphi project)
3. Klik "Run" (F9)



4. Klik "Back Propagation" kemudian pilih "compute"



5. Masukkan input, yang merupakan suhu pemanasan pasca pengelasan

Back Propagation

File Back Propagation

Fill Input section to compute the Back Propagation

Input

Ribuan ← 0

Ratusan ← 8

Puluhan ← 5

Satuan ← 0

Output

Compute

6. Klik "Compute"

Back Propagation

File Back Propagation

Fill Input section to compute the Back Propagation

Input

0

8

5

0

Output

0.34779773225

Compute

Computation complete

7. Penghitungan selesai. Hasilnya ditunjukkan pada kolom "Output" yang merupakan besarnya laju korosi untuk suhu pemanasan pasca proses pengelasan 850°C dengan satuan mmpy (milimeter per year).

⇒ Script pemrograman

```
unit formMain;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Menus, ExtCtrls, ComCtrls,  
BackPro, jpeg, Math;
```

```
type
```

```
TfMain = class(TForm)  
    MainMenu1: TMainMenu;  
    File1: TMenuItem;  
    OpenInputFile1: TMenuItem;  
    Exit1: TMenuItem;  
    BackPropagation1: TMenuItem;  
    Learn1: TMenuItem;  
    Compute1: TMenuItem;  
    Label1: TLabel;  
    pnlCompute: TPanel;  
    lbInput: TLabel;  
    eInput1: TEdit;  
    eInput2: TEdit;  
    eInput3: TEdit;  
    eInput4: TEdit;  
    btCompute: TButton;  
    eOutput: TEdit;  
    lbOutput: TLabel;  
    OpenDialog1: TOpenDialog;  
    pnlLearn: TPanel;  
    ePattern: TEdit;  
    Label4: TLabel;  
    btLearn: TButton;  
    StatusBar1: TStatusBar;  
    Label3: TLabel;  
    Label2: TLabel;  
    Image1: TImage;  
    procedure OpenInputFile1Click(Sender: TObject);  
    procedure FormCreate(Sender: TObject);  
    procedure Exit1Click(Sender: TObject);  
    procedure Learn1Click(Sender: TObject);  
    procedure FormClose(Sender: TObject; var Action: TCloseAction);  
    procedure btLearnClick(Sender: TObject);  
    procedure Compute1Click(Sender: TObject);  
    procedure btComputeClick(Sender: TObject);  
private  
    { Private declarations }
```

```

procedure SetLengthWeight;
procedure ReadInputTarget;
procedure WriteWeight(filename: String; Aij: TMultiDoubleArray; i, j: Integer);
function ReadWeight(filename: String; i, j: Integer): TMultiDoubleArray;
procedure ReadInput;
procedure Calculate;
public
  { Public declarations }
end;

const
  // for BackPropagation
  INPUT_LAYER = 4;
  HIDDEN_LAYER = 5;
  OUTPUT_LAYER = 1;

var
  fMain: TfMain;

  // for BackPropagation
  Xi: array of array [1..INPUT_LAYER] of Double;
  Zin_j, Zj: array [1..HIDDEN_LAYER] of Double;
  Yin_k, Yk: array [1..OUTPUT_LAYER] of Double;
  Tk: array of array [1..OUTPUT_LAYER] of Double;
  Vij: TMultiDoubleArray ;
  Wjk: TMultiDoubleArray;
  Vtij: TMultiDoubleArray ;
  Wtjk: TMultiDoubleArray;
  Target: Integer;

  CurrentPath: String;
  TotPattern: Integer;

  Stopped: Boolean;

implementation

{$R *.dfm}

procedure TfMain.OpenInputFile1Click(Sender: TObject);
begin
  OpenFileDialog1.InitialDir := CurrentPath + '\..\Input';
  OpenFileDialog1.Execute;
end;

procedure TfMain.FormCreate(Sender: TObject);
begin
  // Set length input and target
  SetLength(Xi, 1);

```



```

CurrentPath := GetCurrentDir;
try
  ReadParam(CurrentPath);
except
  MessageBox(Handle, ' Please check file param.ini in Additional folder.
', 'Warning!!!', MB_OK);
end;
end;

```

```

procedure TfMain.Exit1Click(Sender: TObject);
begin
  Application.Terminate;
end;

```

```

procedure TfMain.Learn1Click(Sender: TObject);
begin
  pnlLearn.Visible := True;
  pnlCompute.Visible := False;
  Label1.Visible := False;
  Label2.Visible := False;
end;

```

```

procedure TfMain.btLearnClick(Sender: TObject);
var
  i, j, k: Integer;
  loop, noPattern: Integer;
  finish: Boolean;
  delta_k: array [1..OUTPUT_LAYER] of Double;
  delta_in_j, delta_j: array [1..HIDDEN_LAYER] of Double;
  delta_Wjk: TMultiDoubleArray;
  delta_Vij: TMultiDoubleArray;
  error: Double;
begin
  if (btLearn.Caption = '&Cancel Learn') then
    begin
      Stopped := True;
      btLearn.Caption := '&Learn';
    end;

  if (ePattern.Text = '') then
    begin
      MessageBox(Handle, ' Fill the box for total pattern to learn!
', 'Warning!!!', MB_OK);
    end
  else
    begin
      btLearn.Caption := '&Cancel Learn';
      Application.ProcessMessages;
      StatusBar1.SimpleText := 'Process learning Back Propagation...';
    end;

```

```

//initial lenght of weight
SetLengthWeight;
delta_Wjk :=
SetLengthMultiDoubleArray(delta_Wjk,HIDDEN_LAYER,OUTPUT_LAYER);
delta_Vij :=
SetLengthMultiDoubleArray(delta_Vij,INPUT_LAYER,HIDDEN_LAYER);

// initial weight at random
for i:=0 to INPUT_LAYER do
  for j:=1 to HIDDEN_LAYER do
    begin
      Vij[i,j] := RandomWeight(-0.5,0.5); //Input-Hidden Layer
      Vtij[i,j] := 0;
    end;

  for j:=0 to HIDDEN_LAYER do
    for k:=1 to OUTPUT_LAYER do
      begin
        Wjk[j,k] := RandomWeight(-0.5,0.5); //Hidden-Output Layer
        Wtjk[j,k] := 0;
      end;

// Read input and target from file
ReadInputTarget;

// learning process
loop := 0;
repeat
  Application.ProcessMessages;
  StatusBar1.SimpleText := 'Process learning Back Propagation... ' +
    IntToStr(loop) + ' Epoch';
  loop := loop + 1;
  noPattern := 0;
  error := 0;
  repeat
    noPattern := noPattern + 1;
  finish := False;

// Calculate Hidden_Layer
for j:=1 to HIDDEN_LAYER do
  begin
    Zin_j[j] := ValOfNode(Xi[noPattern],j,Vij);
    // Activation Hidden_Layer
    Zj[j] := Activation(Zin_j[j]);
  end;

// Calculate Output_Layer
for k:=1 to OUTPUT_LAYER do
  begin
    Yin_k[k] := ValOfNode(Zj,k,Wjk);

```



```

// Activationed Output_Layer
Yk[k] := Activation(Yin_k[k]);
end;

error := error + ErrorVal(Tk[noPattern], Yk, OUTPUT_LAYER);
// if (MoreError(CurrentPath + '\log\' ,Tk,Yk,OUTPUT_LAYER, noFile)) then
// begin
// Calculate Error Output_Layer
for k:=1 to OUTPUT_LAYER do
// Calculate delta weight and bias Output-Hidden Layer
delta_k[k] := (Tk[noPattern][k] - Yk[k]) * DifActivation(Yin_k[k]);

delta_Wjk := DeltaWeight(LEARN_RATE, MOMENTUM, delta_k, Zj,
Wjk, Wtjk);
Wtjk := Wjk;

// Calculate Error Hidden_Layer
for j:=1 to HIDDEN_LAYER do
begin
// Calculate delta weight and bias Hidden-Input Layer
delta_in_j[j] := 0;
for k:=1 to OUTPUT_LAYER do
delta_in_j[j] := delta_in_j[j] + (delta_k[k] * Wjk[j,k]);
delta_j[j] := delta_in_j[j] * DifActivation(Zin_j[j]);
end;

delta_Vij := DeltaWeight(LEARN_RATE, MOMENTUM, delta_j,
Xi[noPattern], Vij, Vtij);
Vtij := Vij;

// Calculate update weight
Wjk := UpdateWeight(Wjk, delta_Wjk, HIDDEN_LAYER,
OUTPUT_LAYER);
Vij := UpdateWeight(Vij, delta_Vij, INPUT_LAYER, HIDDEN_LAYER);
until(noPattern = TotPattern);
if (error <= ERROR_MAX) then finish := True;
if (loop = LOOPING_MAX) then finish := True;
until (finish or Stopped);

if (finish) then
begin
WriteWeight('Wjk', Wjk, HIDDEN_LAYER, OUTPUT_LAYER);
WriteWeight('Vij', Vij, INPUT_LAYER, HIDDEN_LAYER);

MessageBox(Handle, ' Congratulation. Learning has complete!
','Sucess!!!',MB_OK);
btLearn.Caption := '&Learn';
end
else
MessageBox(Handle, ' Learning has been canceled! ', 'Unsucessful!!!',MB_OK);

```

```

    StatusBar1.SimpleText := "";
    btLearn.Caption := '&Learn';
end;
end;

```

```

procedure TfMain.ReadInputTarget;

```

```

var

```

```

    myFile: TextFile;
    x, y: Integer;
    number: Char;
    tempString: String;

```

```

begin

```

```

    // Set length input and target

```

```

    SetLength(Xi, StrToInt(ePattern.Text)+1);

```

```

    SetLength(Tk, StrToInt(ePattern.Text)+1);

```

```

    {*****READ INPUT*****}

```

```

    // Try to open the Test.txt file for writing to

```

```

    AssignFile(myFile, CurrentPath + '\..\Input\input.dat');

```

```

    Reset(myFile);

```

```

    // Display the file contents

```

```

    tempString := "";

```

```

    y:= 1;

```

```

    while not Eof(myFile) do

```

```

    begin

```

```

        x := 1;

```

```

        while not Eoln(myFile) do

```

```

            begin

```

```

                // Read one letter at a time

```

```

                Read(myFile, number);

```

```

                if (number < #13) then

```

```

                    if (number < ' ') then tempString := tempString + number

```

```

                else

```

```

                    begin

```

```

                        Xi[y][x] := StrToFloat(tempString)/Power(10,x-1);

```

```

                        tempString := "";

```

```

                        Inc(x);

```

```

                    end

```

```

                else

```

```

                    begin

```

```

                        ReadLn(myFile);

```

```

                        Inc(y);

```

```

                        x := 1;

```

```

                    end;

```

```

            end;

```

```

        // Go to the next line

```

```

        ReadLn(myFile);

```

```

        Inc(y);

```



```

end;

// Close the file
TotPattern := y - 1;
CloseFile(myFile);

{*****READ TARGET*****}
// Try to open the Test.txt file for writing to
AssignFile(myFile, CurrentPath + '\..\Input\target.dat');
Reset(myFile);

// Display the file contents
tempString := "";
y:= 1;
while not Eof(myFile) do
begin
  x:=1;
  while not Eoln(myFile) do
  begin
    // Read one letter at a time
    Read(myFile, number);
    if (number < #13) then
      if (number < ' ') then tempString := tempString + number
      else
        begin
          Tk[y][x] := StrToFloat(tempString);
          tempString := "";
          Inc(x);
        end
      else
        begin
          ReadLn(myFile);
          Inc(y);
          x:=1;
        end;
      end;
    // Go to the next line
    ReadLn(myFile);
    Inc(y);
  end;

// Close the file
CloseFile(myFile);
end;

procedure TfmMain.WriteWeight(filename: String; Aij: TMultiDoubleArray; i, j:
Integer);
var
  m, n: Integer;
  myFile: TextFile;

```

```

begin
  // Try to open the Test.txt file for writing to
  AssignFile(myFile, CurrentPath + '\weight\' + filename + '.dat');
  ReWrite(myFile);
  for m:=0 to i do
    begin
      for n:=0 to j do
        begin
          Write(myFile, (FloatToStr(Aij[m,n]) + ' '));
        end;
        Writeln(myFile);
      end;

  // Close the file
  CloseFile(myFile);
end;

procedure TfMain.SetLengthWeight;
begin
  Vij := SetLengthMultiDoubleArray(Vij, INPUT_LAYER, HIDDEN_LAYER);
  Wjk := SetLengthMultiDoubleArray(Wjk, HIDDEN_LAYER, OUTPUT_LAYER);
  Vtij := SetLengthMultiDoubleArray(Vtij, INPUT_LAYER, HIDDEN_LAYER);
  Wtjk := SetLengthMultiDoubleArray(Wtjk, HIDDEN_LAYER, OUTPUT_LAYER);
end;

procedure TfMain.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Stopped := True;
end;

procedure TfMain.Compute1Click(Sender: TObject);
begin
  pnlCompute.Visible := True;
  pnlLearn.Visible := False;
  Label1.Visible := False;
  Label2.Visible := False;
end;

procedure TfMain.btComputeClick(Sender: TObject);
var
  seq: Integer;
  i, j, x, y: Integer;
  pixel: TColor;
  segment: Integer;
  imageData: TImage;
begin
  if (eInput1.Text = "") OR (eInput2.Text = "") OR (eInput3.Text = "") OR (eInput4.Text = "") then
    begin

```



```

    MessageBox(Handle, ' Fill Input completely! ', 'Warning!!!', MB_OK);
end
else
begin
    Application.ProcessMessages;
    StatusBar1.SimpleText := 'Process computation...';
    eOutput.Text := '';

    // Read weights from file
    SetLengthWeight;
    Wjk := ReadWeight('Wjk', HIDDEN_LAYER, OUTPUT_LAYER);
    Vij := ReadWeight('Vij', INPUT_LAYER, HIDDEN_LAYER);

    // Read inputs from image
    ReadInput;

    // Calculate output and do prediction
    Calculate;

    // Show result
    eOutput.Text := FloatToStr(Yk[1]);

    StatusBar1.SimpleText := 'Computation complete';
end;
end;

function TfMain.ReadWeight(filename: String; i, j: Integer): TMultiDoubleArray;
var
    myFile: TextFile;
    number: Char;
    temp: TMultiDoubleArray;
    tempString: String;
    m, n: Integer;
begin
    temp := SetLengthMultiDoubleArray(temp, i, j);

    // Try to open the Test.txt file for writing to
    AssignFile(myFile, CurrentPath + '\weight\' + filename + '.dat');
    Reset(myFile);
    tempString := '';
    // Display the file contents
    m := 0;
    while not Eof(myFile) do
        begin
            n := 0;
            while not Eoln(myFile) do
                begin
                    // Read one letter at a time
                    Read(myFile, number);
                    if (number < #13) then

```

```

    if (number <> ' ') then tempString := tempString + number
    else
        begin
            temp[m,n] := StrToFloat(tempString);
            tempString := "";
            n := n + 1;
        end
    else
        begin
            ReadLn(myFile);
            m := m + 1;
            n := 0;
        end;
    end;
    // Go to the next line
    ReadLn(myFile);
    m := m + 1;
end;

```

```

// Close the file
CloseFile(myFile);

```

```

Result := temp;
end;

```

```

procedure TfMain.ReadInput;
begin
    Xi[0][1] := StrToInt(eInput1.Text);
    Xi[0][2] := StrToInt(eInput2.Text)/10;
    Xi[0][3] := StrToInt(eInput3.Text)/100;
    Xi[0][4] := StrToInt(eInput4.Text)/1000;
end;

```

```

procedure TfMain.Calculate;
var
    j, k: Integer;
begin
    // Calculate Input_Layer
    for j:=1 to HIDDEN_LAYER do
        begin
            Zin_j[j] := ValOfNode(Xi[0][j],Vij);
            // Activation Hidden_Layer
            Zj[j] := Activation(Zin_j[j]);
        end;
    end;

```

```

// Calculate Output_Layer
for k:=1 to OUTPUT_LAYER do
    begin

```



```
Yin_k[k] := ValOfNode(Zj,k,Wjk);  
// Activated Output_Layer  
Yk[k] := Activation(Yin_k[k]);  
end;  
end;  
  
end.
```

⇒ Daftar Notasi

x	= vektor input pelatihan
t_p	= vektor target output. = $(t_1, t_2, t_3, \dots, t_n)$
X_i	= Input dari unit i
x_i	= Unit ke-1 pada lapisan masukan
Z_j	= Unit ke- j pada lapisan tersembunyi
z_{in_j}	= keluaran untuk unit Z_j
z_j	= nilai aktivasi dari unit Z_j
Y_k	= unit ke- k pada lapisan keluaran
y_{in_k}	= net masukan untuk unit Y_k
y_k	= nilai aktivasi dari unit Y_k
w_{0k}	= nilai penimbang sambungan pada bias untuk unit Y_k
w_{jk}	= nilai penimbang sambungan dari Z_{ij} ke unit Y_k
Δw_{jk}	= selisih antara $w_{jk}(t)$ dengan $w_{jk}(t+1)$
v_{0j}	= nilai penimbang sambungan pada bias untuk unit Z_j
v_{ij}	= nilai penimbang sambungan dari unit X_i ke unit Z_j
Δv_{ij}	= selisih antara $v_{ij}(t)$ dengan $v_{ij}(t+1)$
δ_k	= faktor pengaturan nilai penimbang sambungan pada lapisan keluaran
δ_j	= faktor pengaturan nilai penimbang sambungan pada lapisan tersembunyi
α	= konstanta laju pelatihan (learning rate) $0 < \alpha < 1$

BIODATA PENULIS



Dika Purnama, lahir di Tulungagung 28 Agustus 1985. Merupakan anak kedua dari dua bersaudara, pasangan Ayahanda Suparlan dan Ibunda Purwati. Awal pendidikan penulis dimulai dari SDN 025 Air Emas Ukui, Riau sampai kelas 6 catur wulan 2, kemudian studi dilanjutkan di SDN II Suwaluh, Tulungagung. Jenjang pendidikan selanjutnya ditempuh penulis di MTsN Bandung, Tulungagung, SMUN Boyolangu Tulungagung dan terakhir di Jurusan Teknik Perkapalan, Fakultas Teknologi Kelautan, Institut Teknologi Sepuluh Nopember Surabaya. Selama

menjadi mahasiswa Teknik Perkapalan ITS, penulis aktif di Himpunan Mahasiswa Jurusan (HMJ) Teknik Perkapalan sebagai staff Hubungan Luar Negeri pada periode 2005 dan sebagai Kepala Departemen Kesekretariatan pada periode 2006. Selain aktif di HMJ, penulis juga aktif di UKM Pencak Silat Persaudaraan Setia Hati Terate (PSHT) dan jabatan tertinggi yang pernah dicapai adalah dipercaya untuk menjadi ketua umum pada periode kepemimpinan 2006.