BAB IV
PERANCANGAN DAN PEMBAHASAN

4.1 Data Yang Digunakan

Secara umum, gambaran data yang penulis peroleh dapat dilihat pada lampiran. Data tersebut adalah data hasil proses logging disajikan dalam bentuk ASCII (*American Standard Code for International Intercahange*) terdiri atas 13 kolom. Kolom–kolom tersebut dapat dijelaskan seperti di bawah ini.

- Kolom ke-1 : Depth
- Kolom ke-2 : Log SP
- Kolom ke-3 : ILD (*Deep Resistivity*)
- Kolom ke-4 : ILM (*Medium Resistivity*)
- Kolom ke-5 : SFLU (*SFL Resistivity*)
- Kolom ke-6 : CALI (*Caliper 3*)
- Kolom ke-7 : MSFL (*Micro Spherically Focused Log*)
- Kolom ke-8 : GR (*Gamma-Ray Log*)
- Kolom ke-9 : DT (*Delta T*)
- Kolom ke-10 : RHOB (*Compensated Formation Density*)
- Kolom ke-11 : NHPI (*Neutron Log Porosity*)
- Kolom ke-12 : DRHO (*Density Log Corection*)
- Kolom ke-13 : PEF (*Photoelectric Factor*)

Untuk memudahkan pembacaan data pada program JST yang dibuat dengan bahasa *MATLAB* 6.5, maka data di atas ditulis ulang dalam file *.txt yaitu dengan cara memotong bagian header-nya. Setelah itu, data tersebut di-edit

dengan menggunakan program MATLAB 6.5 supaya mendapatkan data dalam bentuk kolom yang teratur dan mudah dimengerti.

4.2 Strategi Perancangan JST (Jaringan Syaraf Tiruan)

4.2.1 Pandangan umum

Seperti telah dijelaskan pada bagian terdahulu bahwa jaringan syaraf tiruan sangat efektif dan efisien untuk digunakan pada proses pengenalan pola dari sejumlah data tertentu dan juga dapat mengestimasi data lainnya yang tidak diketahui dengan cara melakukan proses pembelajaran pada data yang ada. Hal ini pun dapat dipakai dalam bidang eksplorasi contohnya : untuk mengestimasi nilai kecepatan gelombang seismik dari sejumlah data logging.

Kecepatan gelombang seismik adalah besaran fisis yang berguna untuk analisis sumur produksi. Nilai kecepatan gelombang seismik dapat diperoleh dengan berbagai metode. Dipandang dari segi ekonomis, jaringan syaraf tiruan adalah metode yang cukup efisien untuk mengestimasi nilai kecepatan gelombang seismik. Oleh karena itu, penulis mencoba untuk menggunakan JST untuk untuk mengestimasi nilai kecepatan gelombang seismik dari sejumlah data logging. Algoritma JST yang akan digunakan adalah backpropagation (propagasi balik) dengan fungsi pembelajaran Levenberg-Marquardt.

Sebelum membangun JST backpropagation, ada beberapa hal yang harus kita tentukan yaitu :

- Arsitektur JST yang akan digunakan,
- Data input dan target,

- Jumlah set data yang akan dilatih,
- Normalisai data,
- Jumlah lapisan dan neuron,
- Fungsi aktivasi,
- Fungsi pembelajaran yang akan digunakan,
- Inisialisasi bobot sinapsis dan bias untuk setiap neuron,
- Jumlah epoh maksimum, kinerja tujuan, *show step*, dan *learning rate.*

4.2.2 Data masukan dan target

Pada jaringan syaraf tiruan kita mengenal adanya data masukan/input dan target. Data masukan/input tersebut akan dilatih dengan algoritma pembelajaran tertentu sehingga menghasilkan output/keluaran. Selanjutnya, output tersebut akan dibandingkan dengan target. Jika selisih/*error* antara target dan output kecil atau mendekati nol, maka JST yang kita buat sudah dapat dikatakan akurat.

Pada kasus ini, data masukan/input yang akan digunakan adalah data logging, sedangkan target yang digunakan adalah data transit time (DT).

4.2.3 Jumlah set data

Secara keseluluhan jumlah data untuk setiap kolom ada 6754. Sebagian dari data tersebut akan dipakai untuk proses pelatihan pada JST dan sebagian lagi akan dipakai sebagai data *checking.* Data yang akan dipakai pada proses pelatihan sebanyak 4000 data dan sisanya akan dipakai untuk data checking.

4.2.4 *Preprocessing* untuk normalisasi data

Sebelum dilakukan pelatihan, seringkali dilakukan normalisasi pada input dan target agar data-data tersebut berada dalam bentuk normal. Data dalam bentuk normal adalah data-data tersebut berada pada range tertentu atau mempunyai *mean*/nilai rata-rata pada nilai tertentu. Pada dasarnya proses normalisasi data ada 3 metode, yaitu :

- *Decimal scaling*, yaitu dengan cara mentransformasikan data pada range [-1,1]. Cara ini dapat dilakukan dengan formulasi seperti di bawah ini :

$$v'(i) = v(i)/10^k \tag{4.1}$$

dengan,

v(i) = data sebelum di-normalisasi

v'(i) = data sesudah di-normalisasi

- *Min-max normalization*, yaitu dengan menskalakan data-data tersebut pada range yang diinginkan. Biasanya range tersebut terletak pada [0,1]. Cara ini dapat dilakukan dengan formulasi seperti di bawah ini :

$$v'(i) = \frac{(v(i) - \min A)}{(\max A - \min A) * (\max A' - \min A')} + \min A' \tag{4.2}$$

dengan,

[min A,max A] adalah range mula-mula dan [min A',max A'] adalah range akhir.

- *Zero-mean normalization*, yaitu dengan cara mentransformasikan data-data tersebut ke bentuk normal dengan nilai rata-rata/mean = 0 dan standar deviasi = 1. Formulasinya adalah seperti di bawah ini :

$$v' = (v - meanA) / std_devA \quad (4.3)$$

dengan,

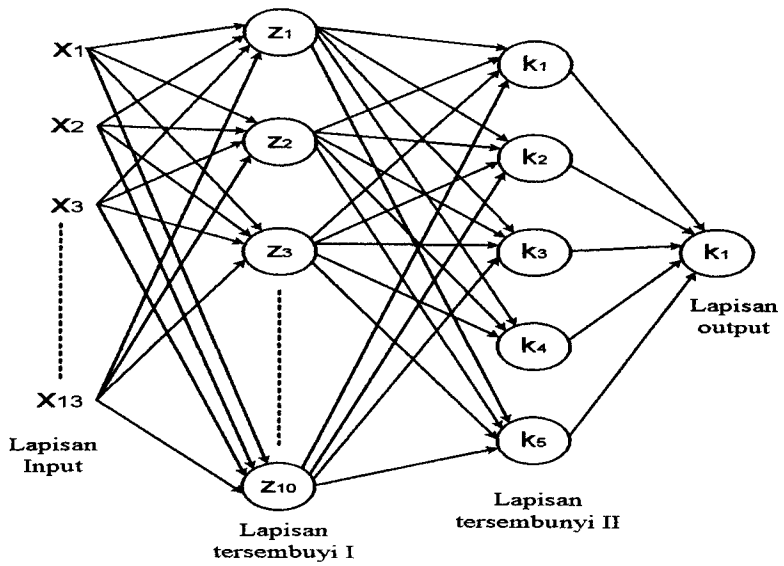
mean A = nilai rata-rata dari data-data yang dimaksud

std_dev A = standar deviasi

Dari 3 metode di atas, zero-mean normalization adalah cara yang paling efektif karena formulasinya lebih sederhana dan prosesnya lebih cepat. Oleh karena itu, pada penelitian ini metode yang akan dipakai adalah zero-mean normalization.

4.2.5 Jumlah lapisan, neuron, dan fungsi aktivasi yang digunakan

Secara umum arsitektur JST yang akan digunakan untuk estimasi kecepatan gelombang seismik dapat dilihat pada gambar di bawah ini.



Gambar 4.1 Arsitektur JST yang akan dipakai pada proses pelatihan

Pada gambar di atas dapat kita lihat bahwa arsitektur JST ini terdiri atas :

- Lapisan input yang terdiri atas 13 neuron. Data input merupakan data logging yang terdiri atas 13 masukan,
- Lapisan tersembunyi pertama yang terdiri atas 10 neuron,
- Lapisan tersembunyi kedua yang terdiri atas 5 neuron,
- Lapisan output yang terdiri atas 1 neuron. Output jaringan adalah nilai kecepatan gelombang seismik.

Arsitektur JST di atas disebut juga arsitektur 13-10-5-1.

Fungsi aktivasi yang digunakan untuk tiap lapisan adalah :

- Lapisan tersembunyi pertama menggunakan fungsi aktivasi tansig,
- Lapisan tersembunyi kedua menggunakan fungsi aktivasi logsig,
- Lapisan output menggunakan fungsi aktivasi purelin atau fungsi identitas.

Secara umum, kelebihan dan kekurangan untuk fungsi aktivasi di atas telah dijelaskan pada bab III.

4.2.6 Inisialisasi bobot sinapsis dan bias untuk setiap neuron

Pada JST yang penulis buat ini, bobot sinapsis dan bias diinisialisasikan sesuai dengan inisialisasi fungsi aktivasi yang digunakan oleh tiap lapisan. Pada program MATLAB 6.5, inisialisasi bobot sinapsis dan bias dapat dilakukan dengan syntax :

net = init(net)

dengan,

net = jaringan syaraf yang ada dikembalikan sesuai dengan inisialisasi fungsinya.

4.2.7 Maksimum epoh, kinerja tujuan, show step, dan learning rate

Pada JST backpropagation dengan fungsi pelatihan Levenberg-Marquard, selain yang telah dibahas pada bagian sebelumnya, ada parameter-parameter lain yang harus di-set, yaitu :

- Maksimum epoh

  Menunjukkan jumlah epoh maksimum yang boleh dilakukan selama proses pelatihan. Iterasi akan dihentikan apabila nilai epoh melebihi nilai maksimum epoh yang telah ditentukan,

- *Goal* atau kinerja tujuan

  Menunjukkan target nilai fungsi kinerja. Iterasi akan dihentikan apabila fungsi kinerja kurang dari atau sama dengan kinerja tujuan,

- *Learning rate*

  Menunjukkan laju pelatihan/pembelajaran pada proses pelatihan JST. Semakin besar nilai learning rate akan berimplikasi pada semakin besarnya langkah pembelajaran. Jika learning rate di-set terlalu besar maka algoritma menjadi tidak stabil dan jika di-set terlalu kecil maka algoritma akan konvergen dalam waktu yang sangat lama,

- *Show step*

  Menunjukkan jumlah epoh berselang yang akan ditunjukkan kemampuannya.

Untuk studi kasus ini, parameter-parameter di atas di-set seperti di bawah ini :

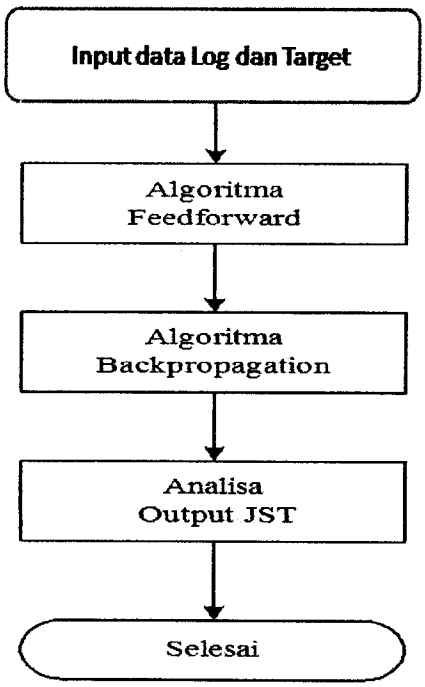
- Maksimum epoh = 5000
- Goal atau kinerja tujuan = 1e-2

- Learning rate = 0,5
- Show step = 200.

4.2.8 Pemecahan masalah

Sebelum membangun JST backpropagation untuk mengestimasi kecepatan gelombang seismik, perlu kita buat algoritma pemecahan masalahnya. Secara singkat, pemecahan masalah tersebut dapat dijelaskan dengan diagram alir seperti di bawah ini.

Diagram Alir Pemecahan Masalah



Gambar 4.2 Diagram alir pemecahan masalah JST untuk estimasi kecepatan gelombang seismik

Pada gambar di atas kita dapat melihat 2 proses penting, yaitu :

- Feedforward

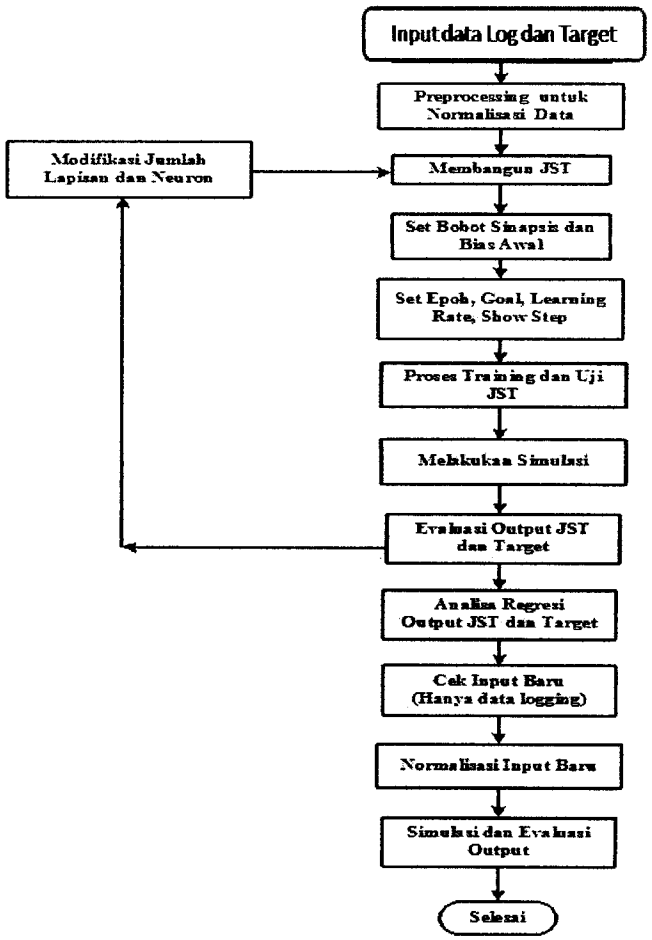
- Backpropagation

Algoritma dan diagram alir untuk proses feedforward dan backpropagation secara lengkap dapat dilihat pada bab III.

4.2.9 Alur program JST untuk mengestimasi kecepatan gelombang seismik.

Berikut ini adalah diagram alir program JST untuk mengestimasi kecepatan gelombang seismik dari sejumlah data logging.

Alur Program JST untuk Estimasi Kecepatan Seismik



Gambar 4.3 Alur program JST untuk estimasi kecepatan gelombang seismik

4.3 Hasil Uji JST dan Analisa

4.3.1 Analisa hasil pelatihan JST

Hasil uji JST dengan bahasa komputasi MATLAB 6.5 untuk studi kasus ini dapat dilihat seperti di bawah ini.

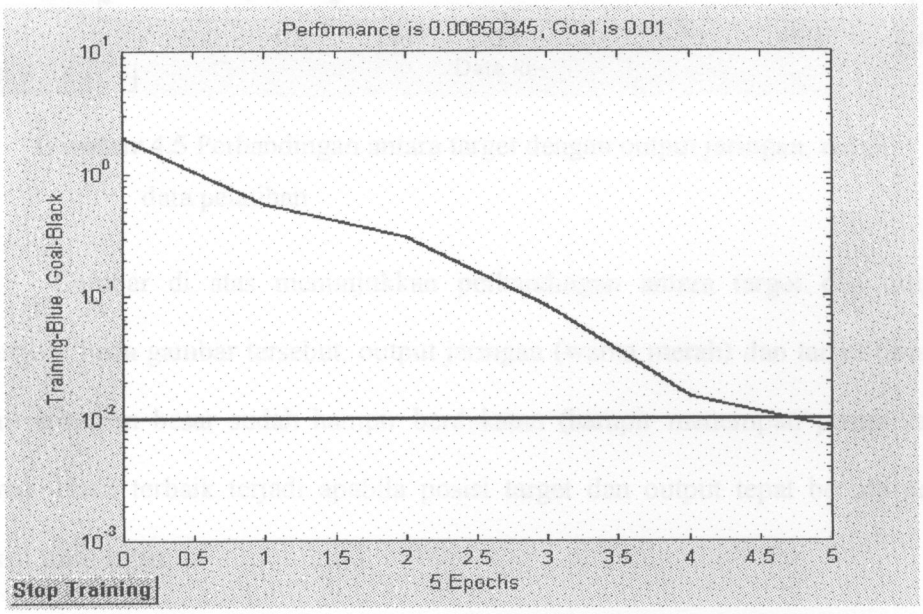
Hasil pelatihan :

TRAINLM, Epoch 0/5000, MSE 1.98693/0.01, Gradient 8099.1/1e-010

TRAINLM, Epoch 5/5000, MSE 0.00850345/0.01, Gradient 769.484/1e-010

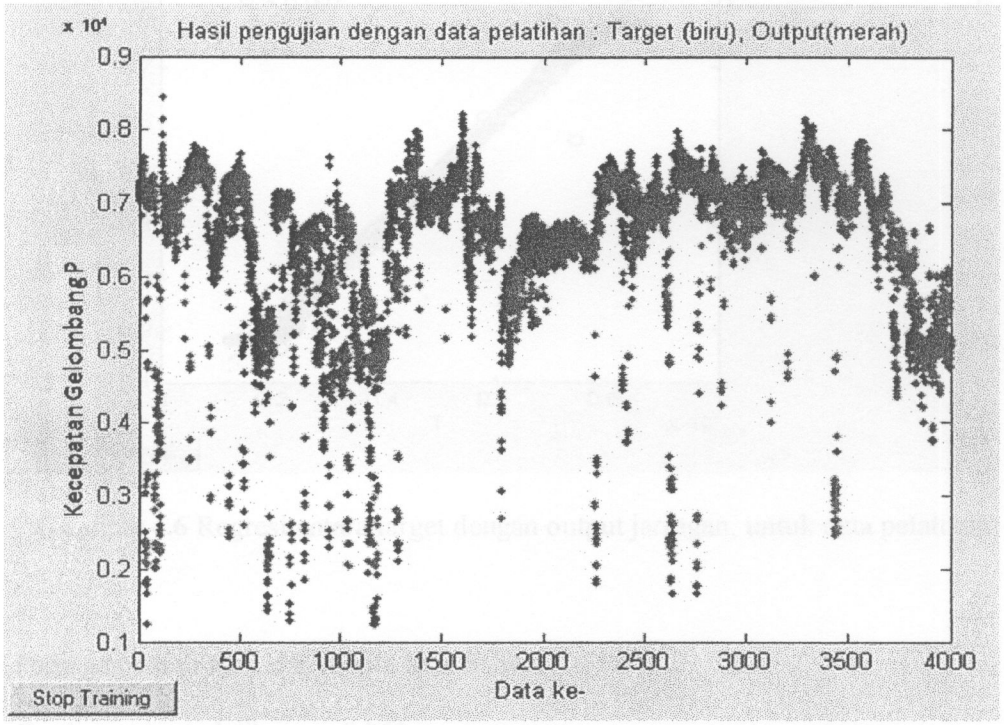
TRAINLM, Performance goal met.

Maksud dari hasil di atas adalah bahwa iterasi akan dihentikan pada epoh ke-5, karena fungsi kinerja (MSE = 0,0085034 < 0,01). Gambar di bawah ini menunjukkan proses pada setiap epoh.



Gambar 4.4 Proses pembelajaran JST

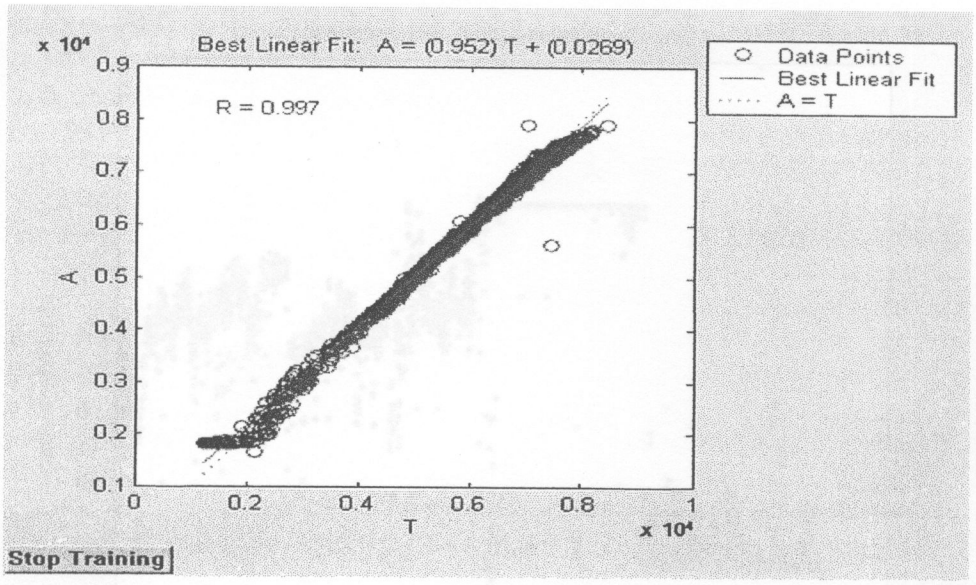
Hasil simulasi antara output JST dengan target dapat dilihat pada gambar di berikut ini.



Gambar 4.5 Perbandingan antara target dengan output jaringan, untuk data pelatihan

Gambar di atas menunjukkan perbandingan antara target dan output jaringan. Pada gambar tersebut, output jaringan (warna merah) dan target (warna biru) sebagian besar sudah hampir berdekatan (hampir menempati posisi yang sama). Hasil terbaik terjadi apabila posisi target dan output tepat berada pada posisi yang sama.

Apabila output JST dan target dianalisis dengan menggunakan regresi linear, maka akan diperoleh gambar seperti berikut ini.



Gambar 4.6 Regresi antara target dengan output jaringan, untuk data pelatihan

Pada gambar di atas, dapat kita lihat bahwa :

Gradien garis terbaik $m_1 = 0{,}952$

Konstanta a = 0,0269

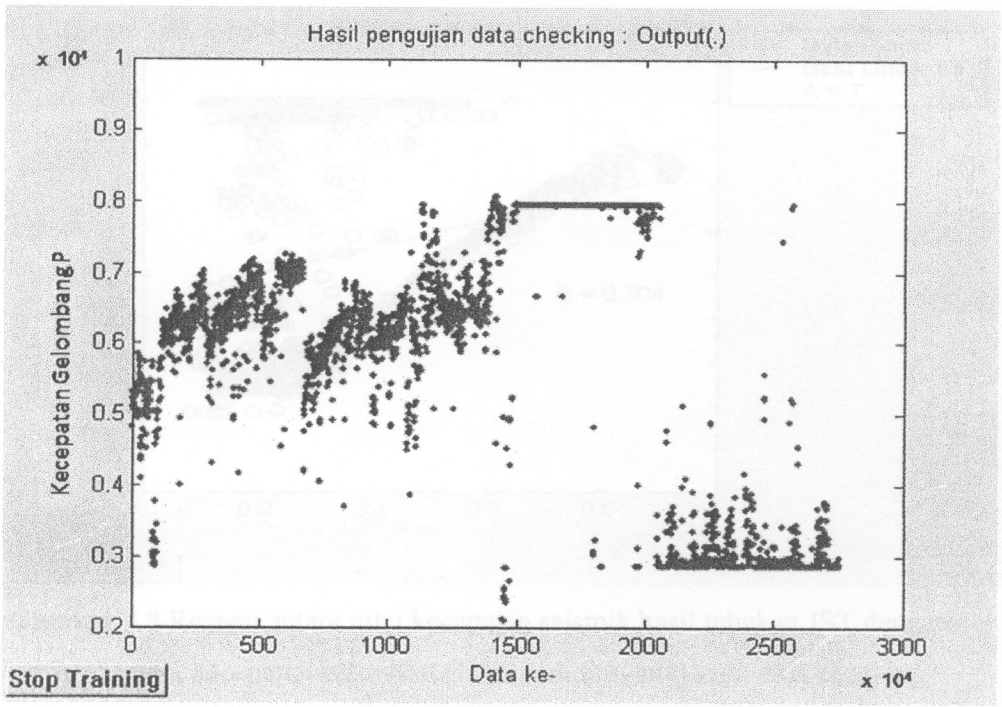
Persamaan garis untuk kecocokan terbaik : $A = 0{,}952T + 0{,}0269$

Koefisien korelasi R = 0,997

Koefisien korelasi bernilai 0,997 (mendekati 1), menunjukkan hasil yang baik untuk kecocokan output jaringan dan target.

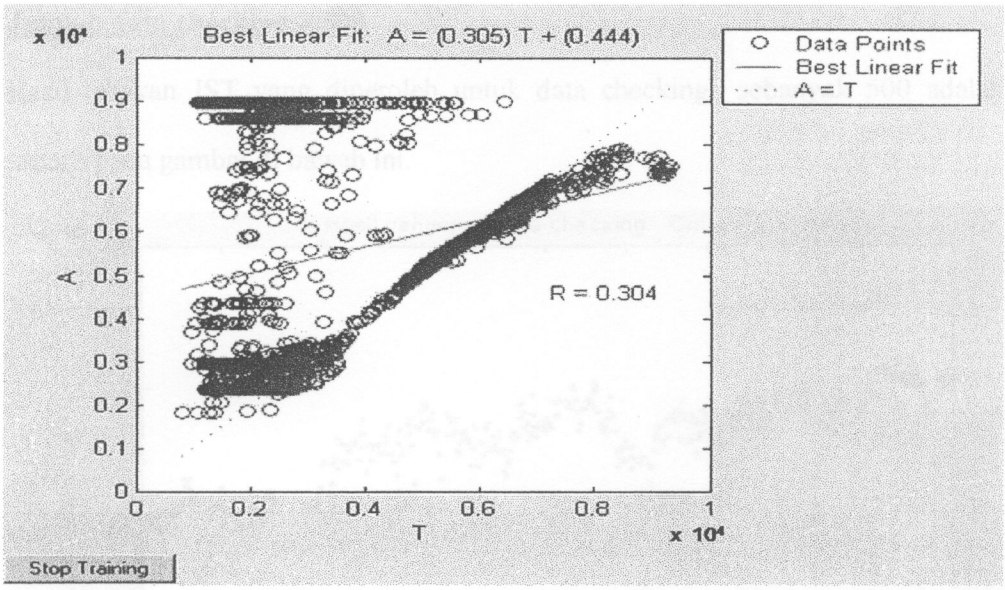
4.3.2 Hasil tebakan JST

Jika kita lakukan pengujian terhadap data-data yang tidak ikut dilatih (Data Checking sebanyak 2754 data) akan diperoleh output jaringan seperti gambar berikut ini.



Gambar 4.7 Hasil tebakan JST untuk data checking sebanyak 2754 data

Untuk mengetahui kualitas data output hasil tebakan JST, maka kita harus mencocokkan data kecepatan seismik hasil tebakan JST dengan data hasil perhitungan. Cara ini dapat dilakukan dengan menggunakan analisis regresi terhadap hasil tebakan JST dan hasil perhitungan. Hasil yang diperoleh dari analisis regresi dapat dilihat pada gambar di bawah ini.



Gambar 4.8 Regresi antara nilai kecepatan seismik hasil tebakan JST dengan nilai kecepatan seismik melalui perhitungan (untuk data checking sebanyak 2754 data)

Pada gambar di atas, dapat kita lihat bahwa :

Gradien garis terbaik $m_1 = 0{,}305$

Konstanta a = 0,444

Persamaan garis untuk kecocokan terbaik : $A = 0{,}305T + 0{,}444$

Koefisien korelasi $R = 0{,}304$

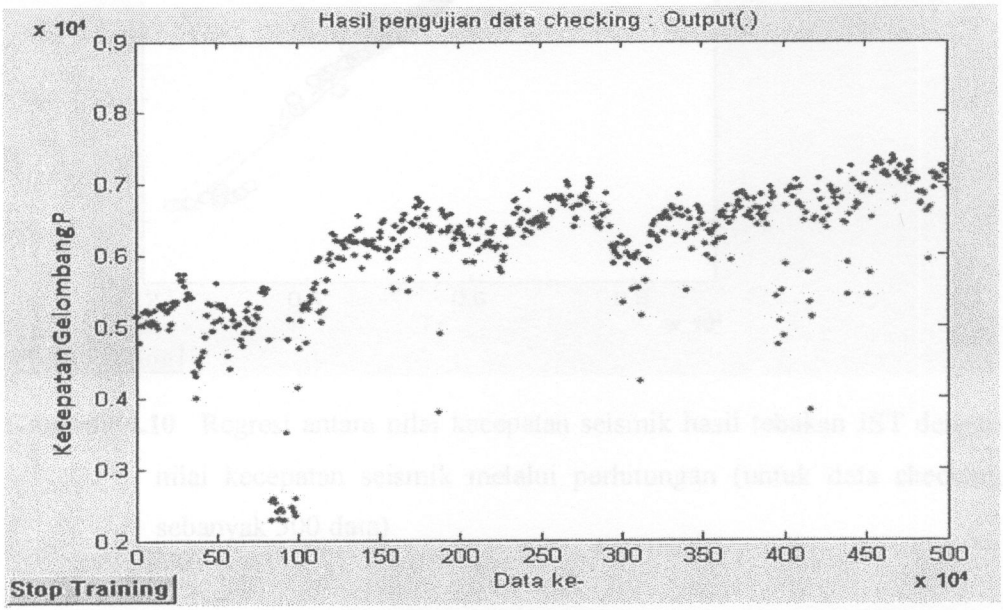
Nilai koefisien korelasi 0,304 menunjukkan bahwa hasil tebakan JST dan hasil perhitungan tidak terlalu cocok.

4.3.3 Hasil tebakan JST untuk jumlah data checking yang berbeda

Untuk mengetahui pengaruh jumlah data checking terhadap data hasil tebakan JST, maka kita harus membandingkan hasil tebakan JST untuk jumlah data checking sebanyak 500, 1000 dan 1500.

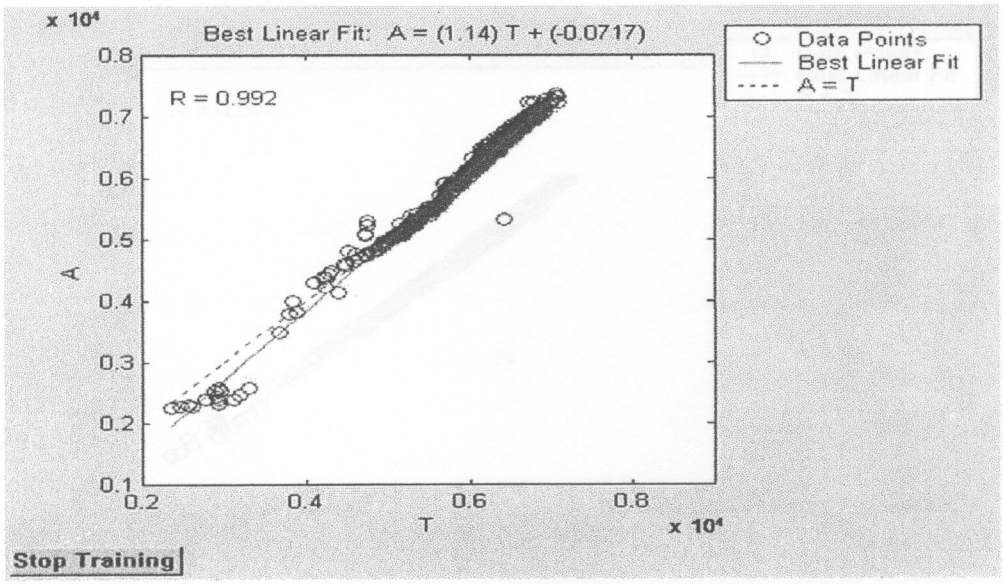
Jumlah data checking = 500

Hasil tebakan JST yang diperoleh untuk data checking sebanyak 500 adalah seperti pada gambar di bawah ini.



Gambar 4.9 Hasil tebakan JST untuk data checking sebanyak 500 data

Dari hasil di atas, kemudian kita buat regresi antara hasil tebakan JST dan hasil perhitungan untuk data checking = 500 data. Hasil regresi yang didapat adalah :



Gambar 4.10 Regresi antara nilai kecepatan seismik hasil tebakan JST dengan nilai kecepatan seismik melalui perhitungan (untuk data checking sebanyak 500 data)

Pada gambar di atas, dapat kita lihat bahwa :

Gradien garis terbaik $m_1 = 1,14$

Konstanta a = -0,0717

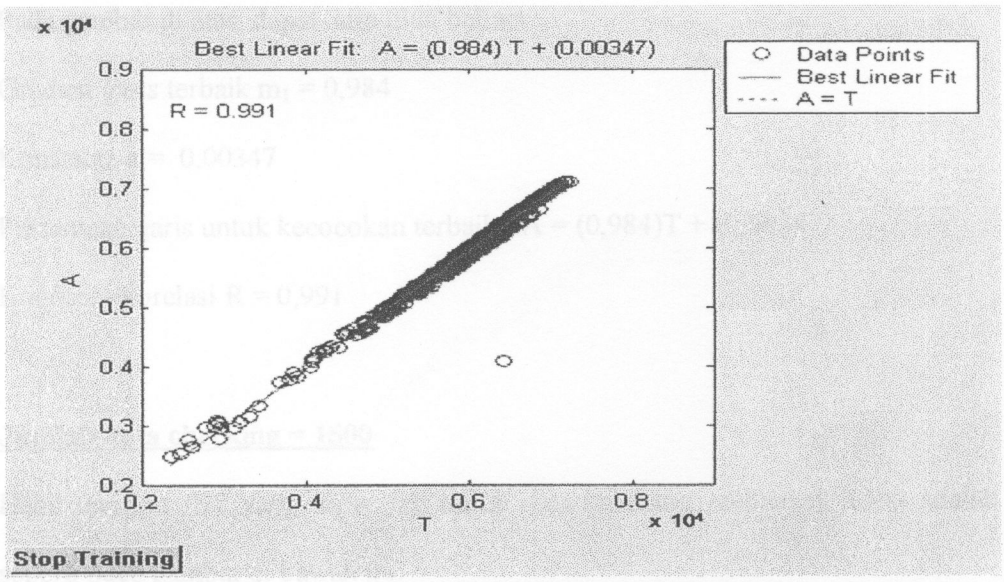
Persamaan garis untuk kecocokan terbaik : $A = (1,14)T - (0,0717)$

Koefisien korelasi $R = 0,992$

Nilai koefisien korelasi 0,992 menunjukkan adanya kecocokan antara data hasil tebakan JST dan hasil perhitungan.

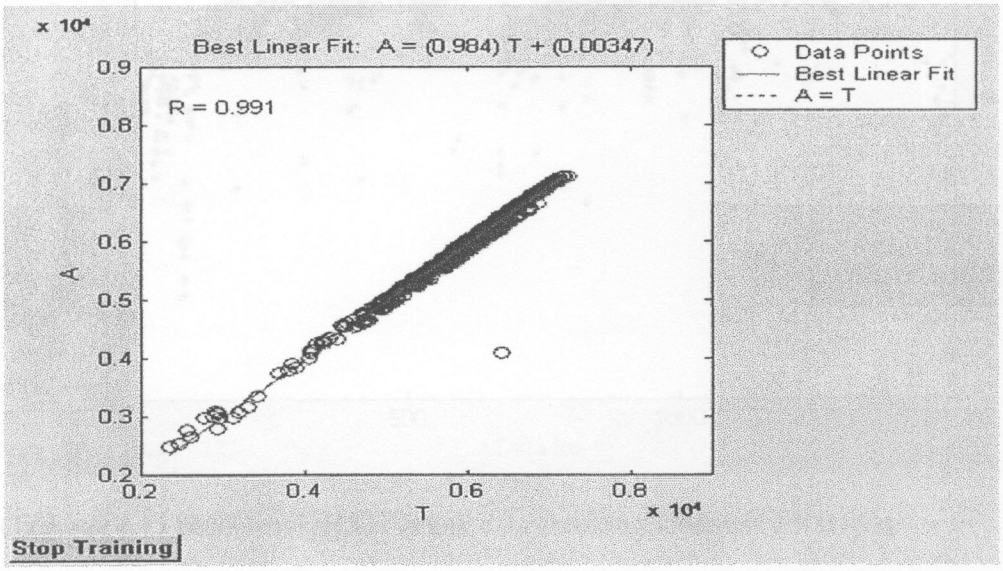
Jumlah data checking = 1000

Hasil tebakan JST yang diperoleh untuk data checking sebanyak 1000 adalah seperti pada gambar berikut ini.



Gambar 4.11 Hasil tebakan JST untuk data checking sebanyak 1000 data

Dari hasil di atas, kemudian kita buat regresi antara hasil tebakan JST dan hasil perhitungan untuk data checking = 1000 data. Hasil regresi yang didapat adalah :



Gambar 4.12 Regresi antara nilai kecepatan seismik hasil tebakan JST dengan nilai patan seismik melalui perhitungan (untuk data checking sebanyak 1000 data)

Pada gambar di atas, dapat kita lihat bahwa :

Gradien garis terbaik $m_1 = 0{,}984$

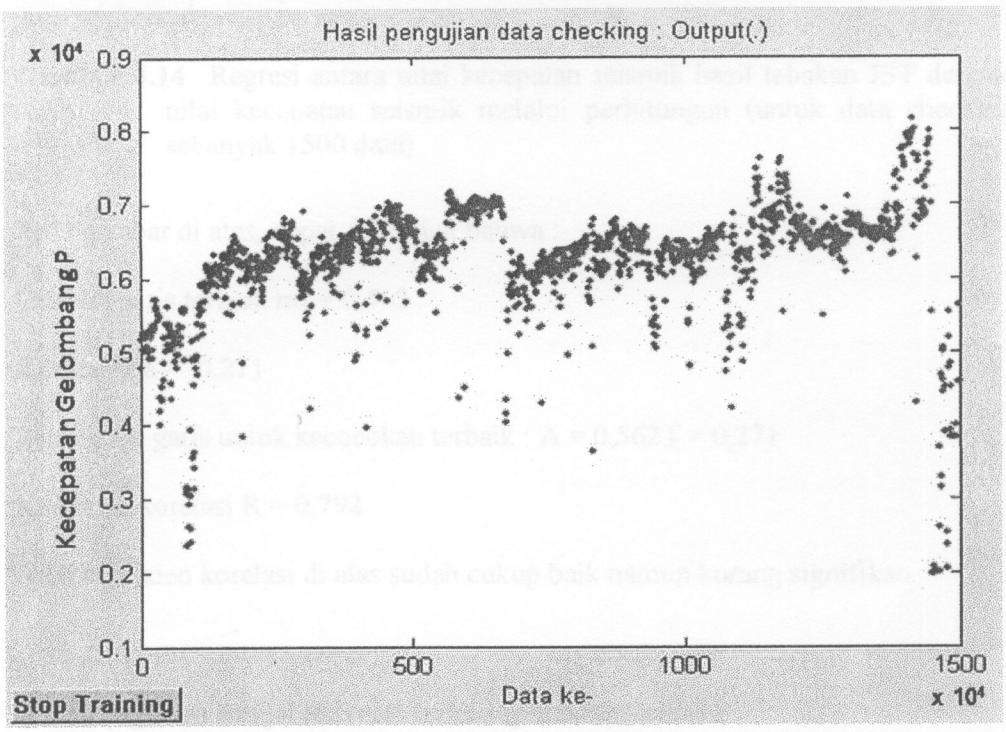
Konstanta a = 0,00347

Persamaan garis untuk kecocokan terbaik : A = (0,984)T + (0,00347)

Koefisien korelasi R = 0,991

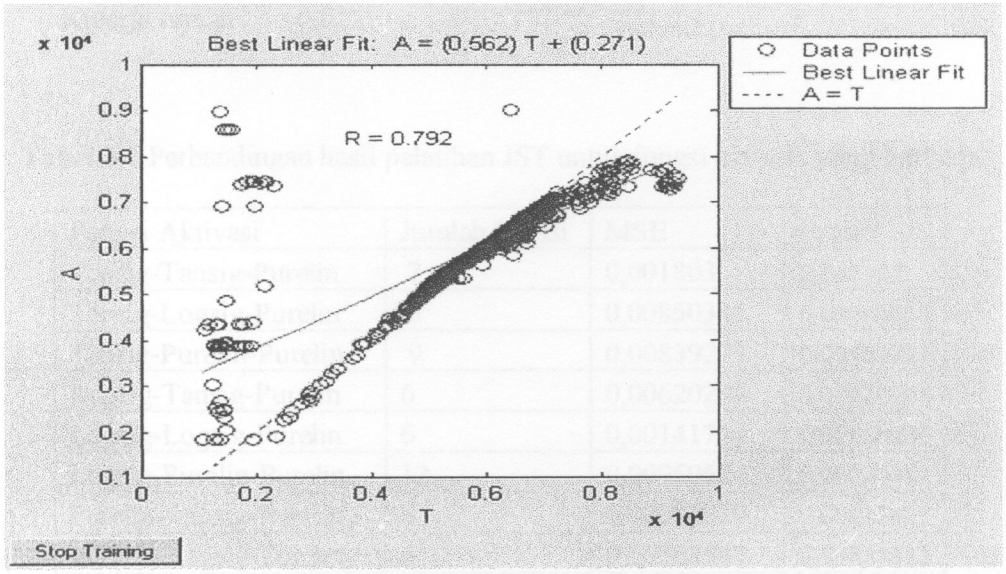
Jumlah data checking = 1500

Hasil tebakan JST yang diperoleh untuk data checking sebanyak 1500 adalah seperti pada gambar di bawah ini.



Gambar 4.13 Hasil tebakan JST untuk data checking sebanyak 1500 data

Sementara hasil regresinya dapat dilihat pada gambar di bawah ini.



Gambar 4.14 Regresi antara nilai kecepatan seismik hasil tebakan JST dengan nilai kecepatan seismik melalui perhitungan (untuk data checking sebanyak 1500 data)

Pada gambar di atas, dapat kita lihat bahwa :

Gradien garis terbaik $m_1 = 0{,}562$

Konstanta a = 0,271

Persamaan garis untuk kecocokan terbaik : $A = 0{,}562T + 0{,}271$

Koefisien korelasi $R = 0{,}792$

Nilai koefisien korelasi di atas sudah cukup baik namun kurang signifikan.

4.3.4 Pengaruh fungsi aktivasi terhadap hasil pelatihan

Di bawah ini adalah pengaruh fungsi aktivasi terhadap hasil pelatihan JST. Parameter yang dijadikan tolok ukur untuk membandingkannya adalah jumlah iterasi dan MSE (Mean Square Error), dengan :

Maksimum epoh = 5000

Kinerja tujuan = 0,01

Tabel 4.1 Perbandingan hasil pelatihan JST untuk fungsi aktivasi yang berbeda

Fungsi Aktivasi	Jumlah Iterasi	MSE	Error
Tansig-Tansig-Purelin	7	0,001803	0,008197
Tansig-Logsig-Purelin	5	0,00850345	0,00149655
Tansig-Purelin-Purelin	9	0,00839273	0,00160727
Logsig-Tansig-Purelin	6	0,00620296	0,00379704
Logsig-Logsig-Purelin	6	0,00141754	0,00858246
Logsig-Purelin-Purelin	12	0,0095953	0,0004047
Purelin-Tansig-Purelin	8	0,00023579	0,00976421
Purelin-Logsig-Purelin	6	0,00568587	0,00431413

Pada tabel di atas, kita dapat melihat bahwa untuk setiap kombinasi fungsi aktivasi yang dipakai pada proses pelatihan JST, kinerja tujuan tercapai pada iterasi tertentu dengan nilai MSE tertentu pula. Jumlah iterasi menunjukkan lamanya lajunya pembelajaran sedangkan nilai MSE menunjukkan ketelitian hasil pelatihan pada JST. Nilai MSE yang baik ialah yang nilainya lebih dekat dengan kinerja tujuan/goal, karena jika MSE mendekati kinerja tujuan maka error-nya kecil. Jadi MSE berbanding terbalik dengan error. Jumlah iterasi paling sedikit dan error terkecil terjadi pada kombinasi fungsi aktivasi Tansig-Logsig-Purelin.

Akan tetapi, ada kombinasi fungsi aktivasi lainnya tidak dapat mencapai kinerja tujuan meskipun jumlah epohnya besar. Di samping itu, pengaruh singularitas pada fungsi aktivasi menyebabkan kinerja tujuan tidak tercapai walaupun iterasinya belum mencapai maksimum. Keadaan ini terjadi pada kombinasi fungsi aktivasi seperti di bawah ini.

Tabel 4.2 Pengaruh fungsi aktivasi terhadap kinerja tujuan

Fungsi Aktivasi	Jumlah Iterasi	Kinerja tujuan
Tansig-Tansig-Tansig	200	Tidak tercapai
Tansig-Tansig-Logsig	200	Tidak tercapai
Tansig-Logsig-Tansig	200	Tidak tercapai
Tansig-Purelin-Tansig	1	Tidak tercapai
Logsig-Purelin-Tansig	400	Tidak tercapai
Purelin-Purelin-Purelin	1	Tidak tercapai

4.3.5 Pengaruh arsitektur JST yang berbeda terhadap hasil pelatihan

Arsitektur JST yang digunakan pada studi kasus ini adalah arsitektur 13-10-5-1. Pada bagian ini, akan dibandingkan pengaruh arsitektur JST yang digunakan terhadap hasil pelatihan JST. Parameter yang dijadikan tolok ukur adalah MSE yang dihasilkan dan jumlah iterasi yang terjadi untuk mencapai kinerja tujuan. Hasil yang didapat adalah sebagai berikut.

Tabel 4.3 Pengaruh arsitektur JST yang berbeda terhadap jumlah iterasi dan nilai MSE

Arsitektur JST	Jumlah iterasi	MSE	Error
13 - 5 - 5 - 1	29	0,00747	0,00253
13 - 10 - 5 -1	5	0,0085	0,0015
13 - 15 - 5 - 1	5	0,0056	0,0044
13 - 20 - 5 - 1	7	0,00698	0,00302
13 - 25 - 5 - 1	7	0,00529	0,00471
13 - 30 - 5 - 1	7	0,00453	0,00547
13 - 5 - 10 - 1	7	0,00848	0,00152
13 - 5 - 15 -1	6	0,00365	0,00635
13 - 5 - 20 - 1	7	0,00684	0,00316
13 - 10 - 10 -1	7	0,00465	0,00535
13 - 15 - 10 - 1	9	0,00206	0,00794
13 - 15 - 15 - 1	5	0,00645	0,00355

dengan,

Maksimum epoh = 5000

Kinerja tujuan = 0,01

Pada tabel di atas, kita dapat melihat bahwa nilai MSE terkecil atau error terbesar untuk mencapai kinerja tujuan terjadi pada arsitektur JST 13 – 15 – 10 – 1, yaitu sebesar 0.00206 dan nilai MSE terbesar atau error terkecil terjadi pada arsitektur 13 – 10 – 5 – 1, yaitu sebesar 0.00850. Sementara, jumlah iterasi terkecil untuk mencapai kinerja tujuan terjadi pada arsitektur 13 – 10 – 5 – 1, 13 – 15 – 5 – 1, dan 13 – 15 – 15 – 1, yaitu sebanyak 5 kali iterasi.

Banyaknya iterasi menunjukkan kecepatan proses pembelajaran JST untuk mencapai kinerja tujuan.