

**PERBANDINGAN METODE *QUASI NEWTON* DENGAN
METODE *GRADIENT DESCENT* UNTUK MENG-*UPDATE*
BOBOT PADA JARINGAN SARAF TIRUAN (JST)
(Studi Kasus pada Data Saham Harian Bank Danamon, Tbk)**

SKRIPSI

Oleh:
FERMIA WINDARI
0310940020-94



**PROGRAM STUDI MATEMATIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2008**

**PERBANDINGAN METODE *QUASI NEWTON* DENGAN
METODE *GRADIENT DESCENT* UNTUK MENG-*UPDATE*
BOBOT PADA JARINGAN SARAF TIRUAN (JST)
(Studi Kasus pada Data Saham Harian Bank Danamon, Tbk)**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar
Sarjana Sains dalam bidang Matematika

oleh :

FERMIA WINDARI

0310940020-94



**PROGRAM STUDI MATEMATIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2008**

LEMBAR PENGESAHAN SKRIPSI

**PERBANDINGAN METODE *QUASI NEWTON* DENGAN
METODE *GRADIENT DESCENT* UNTUK MENG-*UPDATE*
BOBOT PADA JARINGAN SARAF TIRUAN (JST)
(Studi Kasus pada Data Saham Harian Bank Danamon, Tbk)**

oleh :
FERMIA WINDARI
0310940020-94

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 21 Pebruari 2008
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Sains dalam bidang Matematika

Pembimbing I

Pembimbing II

Syaiful Anam, S.Si, M.T
NIP. 132 300 237

Dra. Trisilowati, M.Sc
NIP. 131 837 955

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Agus Suryanto, M.Sc
NIP. 132 126 049

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : FERMI WINDARI
NIM : 0310940020 - 94
Jurusan : MATEMATIKA
Penulisan Tugas Akhir Berjudul :

**PERBANDINGAN METODE *QUASI NEWTON* DENGAN
METODE *GRADIENT DESCENT* UNTUK MENG-*UPDATE*
BOBOT PADA JARINGAN SARAF TIRUAN (JST)
(Studi Kasus pada Data Saham Harian Bank Danamon, Tbk)**

Dengan ini menyatakan bahwa :

1. Isi Skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Skripsi ini.
2. Apabila di kemudian hari ternyata Skripsi yang saya tulis terbukti hasil jiplakan, maka saya bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 21 Februari 2008
Yang menyatakan,

(FERMI WINDARI)
NIM. 0310940020 - 94

**PERBANDINGAN METODE *QUASI NEWTON* DENGAN
METODE *GRADIENT DESCENT* UNTUK MENG-*UPDATE*
BOBOT PADA JARINGAN SARAF TIRUAN (JST)
(Studi Kasus pada Data Saham Harian Bank Danamon, Tbk)**

ABSTRAK

Jaringan Saraf Tiruan (JST) adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan saraf manusia, yang dapat dilatih untuk mengenali suatu obyek yang memiliki pola tertentu. Untuk memperoleh JST yang mampu menyelesaikan permasalahan diperlukan beberapa tahapan diantaranya membangun jaringan (*design*), melatih jaringan (*training*) dan terakhir menguji kemampuan jaringan (*testing*).

Pada tahap melatih JST digunakan model *Backpropagation*. Model ini menggunakan metode *Gradient Descent* dalam proses perubahan bobotnya. Namun metode ini memiliki kelemahan yakni proses pelatihannya membutuhkan waktu yang cukup lama untuk mencapai kekonvergenan dari nilai bobot baru yang meminimumkan kesalahan. Untuk itu, dalam skripsi ini digunakan metode lain yaitu metode *Quasi Newton* sebagai pembandingnya. Adapun tujuan dari skripsi ini yaitu mengetahui perbandingan dari kedua metode dan membandingkan hasil prediksi harga penutupan dari data saham harian Bank Danamon, Tbk. Sebagai kesimpulan, model *Backpropagation* dengan metode *Quasi Newton* lebih baik dibandingkan metode *Gradient Descent* karena metode ini mampu mencapai kekonvergenan dalam jumlah iterasi dan waktu yang relatif singkat.

Kata kunci : Jaringan Saraf Tiruan (JST), *Backpropagation*, metode *Gradient Descent*, metode *Quasi Newton*.

**THE COMPARISON OF QUASI NEWTON METHOD AND
GRADIENT DESCENT METHOD FOR UPDATE OF
WEIGHT ON ARTIFICIAL NEURAL NETWORK
(Study of case at stock exchange of Danamon Bank, Tbk)**

ABSTRACT

Artificial Neural Network (ANN) is an information processing system similar to human neural system which is capable to be trained to characterize an object with certain pattern. To get neural network which capable of problem, solving several steps are needed, such as designing the network's architecture, training the network and simulating the network.

In the training step, Backpropagation model is used in neural network. This model uses Gradient Descent method on its weight update process. However, this method has disadvantage that is relatively slow training process in achieving the convergence of new weight value which minimize the error. Finally, in this final project is used other method that is Quasi Newton method. The purpose of this final project is to compare both of the methods and the result of prediction from stock exchange of Danamon Bank, Tbk. As a result, the Backpropagation model with Quasi Newton method is better than Gradient Descent method because Quasi Newton method capable of achieving the convergence in sum of iteration and a few time .

Keywords : Artificial Neural Network (ANN), Backpropagation, Gradient Descent method, Quasi Newton method.

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT, yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga tugas akhir ini dapat diselesaikan dengan baik. Shalawat serta salam tetap tercurahkan kepada Nabi Muhammad SAW.

Dalam penulisan tugas akhir ini, banyak dukungan dan bantuan yang penulis terima. Oleh sebab itu, penulis ingin berterima kasih kepada:

1. Syaiful Anam, S.Si, M.T, selaku pembimbing I atas segala bimbingan, nasehat, dan sarannya, serta waktu yang telah diberikan kepada penulis selama penulisan tugas akhir ini.
2. Dra. Trisilowati, M.Sc, selaku Pembimbing II atas dorongan, semangat, kritik dan sarannya, serta waktu yang telah diberikan kepada penulis selama penulisan tugas akhir ini.
3. Drs. Hery Subagio, M.kes, selaku penasehat akademik atas segala masukan dan saran selama penulis menempuh kuliah.
4. Dr. Agus Suryanto, M.Sc selaku Ketua Jurusan Matematika atas semua bantuannya dalam kelancaran kelulusan penulis.
5. Dr. Wuryansari, M.K.,M.Si selaku Ketua Program Studi Matematika atas semua bantuannya dalam kelancaran kelulusan penulis.
6. Ibunda dan ayahanda tercinta atas segala doa, kasih sayang, motivasi, didikan dan bimbingannya yang tak pernah kenal lelah, beliau berikan kepada penulis secara tulus dan ikhlas serta adik yang selalu membuat penulis terhibur.
7. Bapak dan Ibu Dosen jurusan Matematika yang telah memberikan bekal ilmu kepada penulis serta Staf Tata Usaha jurusan Matematika atas semua bantuannya.
8. Semua sahabat, teman-teman seperjuangan Matematika angkatan 2003 yang tercinta, kakak-kakak tingkat Matematika 2002 serta adik-adik tingkat 2004 dan 2005 atas semua semangat, motivasi dan dukungannya.
9. Serta semua pihak yang tidak dapat penulis sebutkan satu persatu.

Akhirnya, semoga Allah SWT yang akan membalas semuanya. Penulis menyadari sepenuhnya bahwa dalam penulisan tugas akhir ini tidak lepas dari kesalahan atau kekurangan. Oleh sebab itu, saran dan kritik yang membangun sangat diharapkan oleh penulis.

Akhir kata, semoga tulisan ini bermanfaat bagi penulis
khususnya dan semua pihak pada umumnya.

Malang,

Penulis

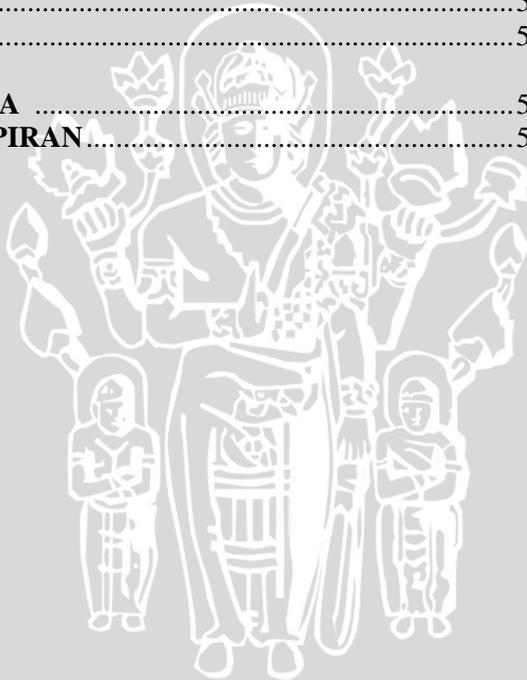
UNIVERSITAS BRAWIJAYA



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERNYATAAN	ii
HALAMAN PENGESAHAN	iii
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
BAB II TINJAUAN PUSTAKA	
2.1 Optimasi	3
2.1.1 Metode <i>Gradient Descent</i>	5
2.1.2 Metode <i>Quasi Newton</i>	8
2.2 Jaringan Syaraf Tiruan	14
2.2.1 Arsitektur JST	15
2.2.2 Fungsi aktivasi	17
2.2.3 Proses Pembelajaran	17
2.2.4 Model Pembelajaran <i>Backpropagation</i>	18
BAB III HASIL DAN PEMBAHASAN	
3.1 Penerapan Metode <i>Gradient-descent</i> Pada Model <i>Backpropagation</i>	22
3.2 Penerapan Metode Quasi Newton Pada Model <i>Backpropagation</i>	24
3.3 Membangun JST	31
3.4 Normalisasi Data	32
3.5 Hasil Pembelajaran	33

3.5.1 Hasil Pembelajaran <i>Backpropagation</i> dengan Metode <i>Gradient Descent</i>	33
3.5.2 Hasil Pembelajaran <i>Backpropagation</i> dengan Metode <i>Quasi Newton</i>	36
3.6 Analisa Perbandingan Hasil Pembelajaran.....	39
3.7 Pemilihan Model Arsitektur Terbaik.....	40
3.7.1 Nilai MSE	41
3.7.2 Jumlah iterasi	45
3.7.3 Total Waktu.....	47
3.8 Hasil Proses Simulasi	49
BAB IV KESIMPULAN DAN SARAN	
4.1 Kesimpulan	54
4.2 Saran	55
DAFTAR PUSTAKA	56
LAMPIRAN-LAMPIRAN	58



BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Perkembangan teknologi yang semakin canggih mendorong manusia untuk terus mencoba mengembangkan penalaran berpikir termasuk kepada komputer. Hal ini dikarenakan peranan komputer yang begitu besar sebagai alat yang membantu manusia dalam bekerja dan diharapkan komputer mampu memberi jawaban seperti manusia. Salah satu bentuknya dengan mengembangkan jaringan saraf tiruan (JST) yang memiliki konsep mirip jaringan otak manusia.

Dalam perkembangannya, JST dapat diaplikasikan dalam berbagai bidang seperti optimasi, pengenalan pola, peramalan, diagnosa dan lain sebagainya (Patterson, 1996). Untuk memperoleh JST yang mampu menyelesaikan permasalahan maka perlu dilakukan beberapa langkah diantaranya membangun jaringan (*design*), melatih jaringan (*training*), dan terakhir menguji kemampuan jaringan (*testing*) (Koivo, 2006).

Pada langkah pelatihan JST digunakan model *Backpropagation*. Model ini menggunakan metode *Gradient Descent* dalam perubahan bobotnya untuk meminimumkan *error*. Namun metode ini memiliki beberapa kelemahan diantaranya dalam pemilihan laju pembelajaran, dimana laju pembelajaran yang terlalu besar menyebabkan algoritma menjadi tidak stabil. Sebaliknya jika laju pembelajaran yang diberikan terlalu kecil maka algoritma akan sangat lama mencapai kekonvergenan (Kusumadewi, 2004).

Oleh karena itu, dalam skripsi ini digunakan metode yang lain yaitu metode *Quasi Newton*. Metode ini merupakan metode penurunan dengan menggunakan pendekatan terhadap matriks *Hessian* atau terhadap invers matriks *Hessian* (Fallgren, 2006). Untuk lebih mempermudah pemahaman, maka kedua metode tersebut diimplementasikan pada suatu studi kasus, dalam hal ini untuk memprediksi harga saham penutupan pada data perdagangan saham harian Bank Danamon, Tbk.

1.2 RUMUSAN MASALAH

Rumusan masalah yang dibahas dalam skripsi ini adalah sebagai berikut :

1. Bagaimana implementasi metode *Gradient Descent* dan metode *Quasi Newton* pada algoritma pembelajaran *Backpropagation* ?
2. Bagaimana perbandingan metode *Quasi Newton* dan metode *Gradient Descent* pada model pembelajaran *Backpropagation* setelah diaplikasikan pada prediksi harga saham penutupan dari data perdagangan saham harian Bank Danamon, Tbk ?

1.3 TUJUAN

Berdasarkan pada permasalahan yang telah diuraikan di atas, sehingga tujuan dari penulisan skripsi ini adalah :

1. Menerapkan metode *Gradient Descent* dan metode *Quasi-Newton* pada algoritma pembelajaran *Backpropagation*.
2. Membandingkan metode *Quasi-Newton* dan metode *Gradient Descent* pada pembelajaran *Backpropagation* setelah diaplikasikan pada prediksi harga saham penutupan dari data perdagangan saham harian Bank Danamon, Tbk.

1.4 BATASAN MASALAH

Topik mengenai JST cukup luas sehingga perlu diberikan batasan masalah sebagai berikut :

1. Arsitektur JST yang dibangun menggunakan satu lapisan tersembunyi (*hidden layer*).
2. Laju pembelajaran yang digunakan bersifat statis.

BAB II TINJAUAN PUSTAKA

2.1 OPTIMASI

Optimasi merupakan suatu cara untuk mengambil keputusan. Di dalam matematika optimasi diartikan sebagai suatu metode untuk meminimumkan atau memaksimumkan fungsi tujuan dengan kendala berdasarkan variabel-variabel yang dicari. Permasalahan optimasi dapat ditulis sebagai

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \quad (2.1)$$

$$\text{dengan kendala berbentuk } \begin{cases} c_i(\mathbf{x}) = 0, & i \in \xi \\ c_i(\mathbf{x}) \geq 0, & i \in \zeta \end{cases}$$

di mana f adalah fungsi obyektif

c_i adalah fungsi bernilai skalar dari variabel \mathbf{x}
 ξ dan ζ adalah himpunan dari indeks

(Nocedal dan Stephen, 1999).

Definisi 2.1 (Minimum Lokal)

Diberikan $\mathbf{x}^* \in S \subseteq \mathbb{R}^n$, \mathbf{x}^* dikatakan sebagai minimum lokal dari fungsi f dalam S jika terdapat persekitaran $N(\mathbf{x}^*)$ di dalam S sedemikian sehingga jika $\mathbf{x} \in N(\mathbf{x}^*)$ maka berlaku $f(\mathbf{x}^*) \leq f(\mathbf{x})$ (Leader, 2004).

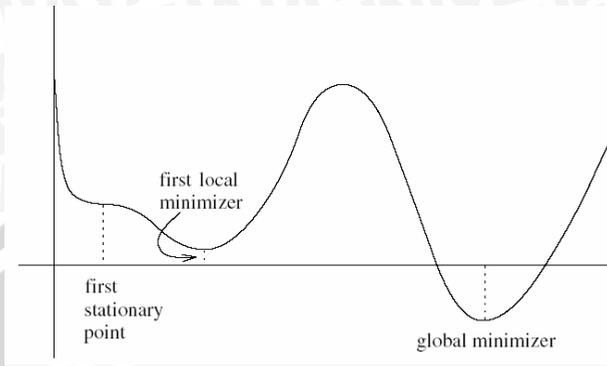
Definisi 2.2 (Minimum Global)

Diberikan $\mathbf{x}^* \in S \subseteq \mathbb{R}^n$, \mathbf{x}^* disebut sebagai minimum global dari fungsi f jika untuk semua $\mathbf{x} \in S$ berlaku

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad (2.2)$$

(Bronshtein, dkk. 2007).

Untuk lebih memahami mengenai minimum lokal (local minimizer) dan minimum global (global minimizer), berikut dijelaskan pada Gambar 2.1.



Gambar 2.1 Minimum lokal dan minimum global

Definisi 2.3 (Gradient)

Misalkan $f(\mathbf{x})$ adalah fungsi dari \mathbf{x} , dimana $\mathbf{x} = (x_1, x_2, \dots, x_n)$ suatu vektor dalam himpunan terbuka \mathbb{R}^n , sedemikian sehingga $\frac{\partial f(\mathbf{x})}{\partial x_k}$ ada untuk $k = 1, 2, \dots, n$. *Gradient* dari $f(\mathbf{x})$, dinotasikan dengan $\nabla f(\mathbf{x})$, adalah vektor kolom

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right)^T \quad (2.3)$$

(Mathews dan Kurtis, 2004).

Dalam skripsi ini, kajian lebih difokuskan pada permasalahan optimasi nonlinier tanpa kendala, sehingga diperoleh optimasi yang berbentuk

$$\min f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^n \quad (2.4)$$

dimana $f : \mathbb{R}^n \rightarrow \mathbb{R}$ merupakan suatu fungsi yang dapat diturunkan dan kontinu. Kondisi yang diperlukan untuk memperoleh titik \mathbf{x}^* menjadi optimal adalah

$$\nabla f(\mathbf{x}^*) = 0. \quad (2.5)$$

Pada umumnya permasalahan tersebut akan dipecahkan menggunakan algoritma perulangan yang membentuk sebuah barisan dari titik-titik $x_0, x_1, \dots \in \text{dom } f$, dengan $f(x_k) \rightarrow f^*$ dan $k \rightarrow \infty$.

Secara umum untuk bisa menyelesaikan persamaan (2.5) dibentuk barisan

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k \quad (2.6)$$

di mana $\alpha_k > 0$ adalah ukuran langkah

\mathbf{d}_k adalah arah penurunan

(Boyd dan Lieven, 2004).

2.1.1 Metode *Gradient Descent*

Metode *Gradient Descent* merupakan salah satu metode untuk meminimumkan suatu masalah optimasi tanpa kendala. Suatu fungsi akan dikatakan menurun jika memenuhi

$$f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k) \quad (2.7)$$

dengan arah penurunan yang harus memenuhi

$$\nabla f(\mathbf{x}_k)^T \mathbf{d}_k < 0. \quad (2.8)$$

Pada metode *Gradient Descent* arah penurunan dari fungsi dipilih dengan mengambil *gradient* bernilai negatif atau $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$.

(Boyd dan Lieven, 2004).

Algoritma metode *Gradient Descent* adalah sebagai berikut :

Langkah 1 : $\mathbf{x}_0 \in \mathbb{R}^n$

Langkah 2 : Pilih $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$

Langkah 3 : Pilih $\alpha_k > 0$

Langkah 4 : Hitung $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$, $k = k + 1$

Langkah 5 : Berhenti hingga kriteria penghentian $\|\nabla f(\mathbf{x}_k)\| < \varepsilon$, jika tidak kembali ke langkah 2.

(Buchanan dan Peter, 1992).

Teorema 2.1 (Bolzano-Weierstrass)

Setiap subhimpunan tak berhingga yang terbatas dalam \mathbb{R}^n mempunyai titik limit (Lewin, 1993).

Teorema 2.2

Misalkan fungsi $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ terdiferensial kontinu pada himpunan $S = \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$, S terbatas dan tertutup. Maka setiap titik limit \mathbf{x}^* dari himpunan $\{\mathbf{x}_k\}$ memenuhi $\nabla f(\mathbf{x}^*) = 0$.

Bukti:

Pembuktian dilakukan dengan menggunakan kontradiksi. Karena $\{x_k\} \subset S$ terbatas dan tertutup, menurut Teorema 2.1 terdapat paling sedikit satu titik limit dari $\{x_k\}$. Misalkan x^* titik limit dari $\{x_k\}$. Andaikan x_k konvergen ke x^* , namun $\nabla f(x^*) \neq 0$.

Karena $d^* = -\nabla f(x^*) \neq 0$ maka terdapat $\alpha > 0$ sedemikian hingga $\delta = f(x^*) - f(x^* + \alpha d^*) > 0$.

Perhatikan barisan d_k berikut :

$$d_1 = -\nabla f(x_1 = x_0 + \alpha_0 d_0)$$

$$d_2 = -\nabla f(x_2 = x_1 + \alpha_1 d_1)$$

:

$$d_k = -\nabla f(x_k = x_{k-1} + \alpha_{k-1} d_{k-1})$$

Karena $x_k \rightarrow x^*$ dan f terdiferensial maka $d_k \rightarrow d^* = -\nabla f(x^*)$ dan $x_k + \alpha d_k \rightarrow x^* + \alpha d^*$. Untuk k cukup besar didapatkan

$$f(x_k + \alpha d_k) \leq f(x^* + \alpha d^*) + \frac{\delta}{2} = f(x^* + \alpha d^*) + \delta - \frac{\delta}{2} = f(x^*) - \frac{\delta}{2}. \text{ Di}$$

lain pihak, $f(x^*) < f(x_k + \alpha d_k) \leq f(x_k + \alpha d_k) \leq f(x^*) - \frac{\delta}{2}$. Hal ini

adalah sebuah kontradiksi, karena $\frac{\delta}{2}$ adalah sebuah bilangan positif,

seharusnya $f(x^*) > f(x^*) - \frac{\delta}{2}$. Jadi $\nabla f(x^*) = 0$ (Freund, 2004).

Definisi 2.4 (Himpunan Konvek)

Misalkan himpunan S berada dalam \mathbb{R}^n , S disebut sebagai himpunan konvek jika terdapat dua buah titik yang membentuk garis lurus di dalam S dan titik tengahnya berada dalam S untuk setiap $x, y \in S$, dimana

$$\lambda x + (1 - \lambda)y \in S, \quad \lambda \in [0, 1] \quad (2.9)$$

(Nocedal dan Stephen, 1999).

Definisi 2.5 (Bentuk kuadrat)

Bentuk kuadrat dari suatu fungsi adalah hasil kali dari dua peubah. Bentuk kuadrat dari x_1, x_2, \dots, x_n dapat ditulis sebagai

$$[x_1 \ x_2 \ \dots \ x_n] A \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (2.10)$$

dengan A matriks simetrik berukuran $n \times n$ (Anton, 1987).

Contoh bentuk kuadrat dari

$$x_1^2 + 7x_2^2 - 3x_3^2 + 4x_1x_2 - 2x_1x_3 + 6x_2x_3$$

adalah

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 2 & 7 & 3 \\ -1 & 3 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Definisi 2.6 (Definit Positif)

Bentuk kuadrat $\mathbf{x}^T A \mathbf{x}$ disebut sebagai definit positif (*positive definite*) jika $\mathbf{x}^T A \mathbf{x} > 0$ untuk semua $\mathbf{x} \neq 0$, sedangkan matriks simetrik A disebut matriks definit positif jika $\mathbf{x}^T A \mathbf{x}$ adalah bentuk kuadrat definit positif (Anton, 1987).

Teorema 2.3

Matriks simetrik A adalah definit positif jika dan hanya jika determinan setiap submatriks utama adalah positif. Submatriks utama A adalah submatriks yang terbentuk dari baris r pertama dan kolom r dari A untuk $r=1, 2, \dots, n$ (Anton, 1987).

Contoh matriks definit positif

Diberikan $A = \begin{bmatrix} 2 & -1 & -3 \\ -1 & 2 & 4 \\ -3 & 4 & 9 \end{bmatrix}$

disebut positif definit karena $A_1 = |2| = 2$,

$$A_2 = \begin{vmatrix} 2 & -1 \\ -1 & 2 \end{vmatrix} = 3 \quad \text{dan} \quad A_3 = \begin{vmatrix} 2 & -1 & -3 \\ -1 & 2 & 4 \\ -3 & 4 & 9 \end{vmatrix} = 1$$

karena semua submatriks utama A adalah positif.

Definisi 2.7 (Matriks Hessian)

Misalkan $f(\mathbf{x})$ adalah fungsi dari \mathbf{x} , dimana $\mathbf{x} = (x_1, x_2, \dots, x_n)$ suatu vektor dalam himpunan terbuka \mathbb{R}^n , sedemikian sehingga $\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j}$ ada untuk $i, j = 1, 2, \dots, n$. Matriks Hessian dari $f(\mathbf{x})$, dinotasikan dengan $Hf(\mathbf{x})$ dengan ukuran matriks $n \times n$

$$Hf(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} & \dots & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n^2} \end{bmatrix} \quad (2.11)$$

(Mathews dan Kurtis, 2004).

Definisi 2.8 (Konvergen Superlinier)

Terdapat $\{\mathbf{x}_k\}$ merupakan suatu barisan dalam \mathbb{R}^n yang konvergen ke \mathbf{x}^* , selanjutnya barisan tersebut dikatakan konvergen superlinier jika

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|} = 0 \quad (2.12)$$

(Nocedal dan Stephen, 1999).

2.1.2 Metode Quasi Newton

Metode *Quasi Newton* adalah salah satu metode untuk meminimumkan suatu fungsi. Metode ini menggunakan pendekatan terhadap matriks Hessian atau terhadap invers matriks Hessian tanpa harus menghitung langsung nilai dari turunan kedua yang dianggap terlalu rumit (Fallgren, 2006).

Dengan memandang suatu pendekatan terhadap fungsi dengan menggunakan deret Taylor orde dua diperoleh bentuk

$$f(\mathbf{x}_k + \mathbf{d}_k) \approx f(\mathbf{x}_k) + \mathbf{d}_k^T \nabla f(\mathbf{x}_k) + \frac{1}{2} \mathbf{d}_k^T Hf(\mathbf{x}_k) \mathbf{d}_k \quad (2.13)$$

dan menganggap fungsi tersebut akan mencapai optimal jika

$$\nabla f(\mathbf{x}_k + \mathbf{d}_k) = 0. \quad (2.14)$$

maka dari persamaan (2.14) diperoleh

$$\nabla f(\mathbf{x}_k + \mathbf{d}_k) \approx \nabla f(\mathbf{x}_k) + Hf(\mathbf{x}_k) \mathbf{d}_k = 0 \quad (2.15)$$

$$Hf(\mathbf{x}_k) \mathbf{d}_k = -\nabla f(\mathbf{x}_k) \quad (2.16)$$

untuk mempermudah penulisan matriks *Hessian* yang bernilai definit positif maka dilakukan perubahan notasi yaitu

$$Hf(\mathbf{x}_k) = H_k. \quad (2.17)$$

Dengan memandang permasalahan optimasi pada persamaan (2.4), bentuk persamaan (2.6) menjadi

$$\mathbf{x}_{k+1} - \mathbf{x}_k = \alpha_k \mathbf{d}_k \quad (2.18)$$

misal $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ diperoleh

$$\mathbf{s}_k = \alpha_k \mathbf{d}_k. \quad (2.19)$$

Sebagai nilai awal, digunakan matriks identitas karena informasi awal belum tersedia sehingga metode *Quasi Newton* pada mulanya sama seperti metode *Gradient Descent*. Setelah dihitung \mathbf{x}_{k+1} akan diperoleh pendekatan matriks *Hessian* baru yang berbentuk

$$H_{k+1} = H_k + \Delta H_k \quad (2.20)$$

dengan

$$\begin{aligned} \nabla f(\mathbf{x}_{k+1}) &= \nabla f(\mathbf{x}_k + \mathbf{x}_{k+1} - \mathbf{x}_k) \\ &= \nabla f(\mathbf{x}_k + \mathbf{s}_k) \\ &= \nabla f(\mathbf{x}_k) + H_k \mathbf{s}_k + \dots \end{aligned}$$

$$H_k \mathbf{s}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k) \quad (2.21)$$

sehingga H_{k+1} yang diartikan sebagai pendekatan terhadap matriks *Hessian* harus memenuhi persamaan (2.21), dengan memisalkan

$$\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k) \quad (2.22)$$

diperoleh

$$H_{k+1} \mathbf{s}_k = \mathbf{y}_k. \quad (2.23)$$

Persamaan (2.23) disebut sebagai kondisi *Quasi Newton* untuk pendekatan terhadap matriks *Hessian*. Selanjutnya dari persamaan (2.20) dan didefinisikan

$$\Delta H_k = \beta \mathbf{y}_k \mathbf{y}_k^T + \gamma \mathbf{w}_k \mathbf{w}_k^T \quad (2.24)$$

dengan
$$\mathbf{w}_k = H_k \mathbf{s}_k \quad (2.25)$$

$$\beta = \frac{1}{\mathbf{y}_k^T \mathbf{s}_k} \quad (2.26)$$

$$\gamma = -\frac{1}{\mathbf{w}_k^T \mathbf{s}_k} \quad (2.27)$$

sehingga dari persamaan (2.20), (2.24)-(2.27) diperoleh bentuk sebagai berikut :

$$H_{k+1} = H_k + \beta \mathbf{y}_k \mathbf{y}_k^T + \gamma \mathbf{w}_k \mathbf{w}_k^T \quad (2.28)$$

$$H_{k+1} = H_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} - \frac{H_k \mathbf{s}_k (\mathbf{H}_k \mathbf{s}_k)^T}{\mathbf{s}_k^T H_k \mathbf{s}_k} \quad (2.29)$$

Persamaan (2.29) dikenal dengan istilah *formula BFGS* untuk pendekatan terhadap matriks *Hessian*, rumus ini diberi nama sesuai dengan nama penemunya yakni Broyden, Fletcher, Goldfard dan Shanno.

Menurut Buchanan dan Peter (1992), pendekatan terhadap matriks *Hessian* haruslah matriks simetrik ukuran $n \times n$ yang bernilai definit positif, karena dengan adanya syarat tersebut dapat menjamin fungsi berjalan menuju nilai minimum. Syarat tersebut berbentuk

$$\nabla f(\mathbf{x}_k)^T \mathbf{d}_k = -\nabla f(\mathbf{x}_k)^T H_k^{-1} \nabla f(\mathbf{x}_k) < 0. \quad (2.30)$$

Dengan memandang kembali persamaan (2.19) diperoleh

$$\mathbf{d}_k = \mathbf{s}_k / \alpha_k \quad (2.31)$$

dari persamaan (2.31) jika dilakukan substitusi pada persamaan (2.16) diperoleh

$$H_k \mathbf{s}_k = -\alpha_k \nabla f(\mathbf{x}_k). \quad (2.32)$$

Dengan demikian, jika persamaan (2.32) dan (2.19) disubstitusi pada persamaan (2.29) diperoleh

$$H_{k+1} = H_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\alpha_k \mathbf{y}_k^T \mathbf{d}_k} - \frac{(-\alpha_k \nabla f(\mathbf{x}_k))(-\alpha_k \nabla f(\mathbf{x}_k))^T}{-\alpha_k \nabla f(\mathbf{x}_k)^T \alpha_k \mathbf{d}_k} \quad (2.33)$$

$$H_{k+1} = H_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\alpha_k \mathbf{y}_k^T \mathbf{d}_k} + \frac{(\nabla f(\mathbf{x}_k))(\nabla f(\mathbf{x}_k))^T}{(\nabla f(\mathbf{x}_k))^T \mathbf{d}_k} \quad (2.34)$$

(Gill, dkk. 1981).

Algoritma metode *Quasi Newton* adalah

Langkah 1: Input $\mathbf{x}_0, k = 0$ dan kriteria pemberhentian ε

Inisialisasi $H_0 = I$

Langkah 2: Hitung $\mathbf{d}_k = -H_k^{-1} \nabla f(\mathbf{x}_k)$

Langkah 3: Pilih α_k untuk meminimumkan $f(\mathbf{x}_k + \alpha_k \mathbf{d}_k)$

Langkah 4: Hitung $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{d}_k$

Langkah 5: Hitung H_{k+1} dengan menggunakan persamaan (2.34)

Langkah 6: $k \leftarrow k + 1$

Langkah 7: Berhenti ketika $(\|\nabla f(\mathbf{x}_k)\| < \varepsilon)$ jika tidak kembali ke langkah 2

(Buchanan dan Peter, 1992).

Sebelum mengulas mengenai teori kekonvergenan dari metode *Quasi Newton* sebaiknya meninjau asumsi yang akan digunakan dalam teorema tersebut.

Asumsi 2.1

1. Fungsi tujuan f memiliki turunan kedua yang kontinu.
2. Diberikan himpunan $S = \{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$ yang merupakan himpunan konvek, dimana terdapat konstanta positif m dan M sedemikian sehingga

$$m \|\mathbf{z}\|^2 \leq \mathbf{z}^T H f(\mathbf{x}_k) \mathbf{z} \leq M \|\mathbf{z}\|^2 \quad (2.35)$$

untuk semua $\mathbf{z} \in \mathbb{R}^n$ dan $\mathbf{x}_k \in S$.

(Nocedal dan Stephen, 1999).

Teorema 2.4 (Konvergen global dari formula BFGS)

Jika diberikan H_0 sebagai sembarang matriks awal yang simetrik dan definit positif serta diberikan pula \mathbf{x}_0 sebagai nilai awal yang memenuhi asumsi 2.1, maka barisan $\{\mathbf{x}_k\}$ yang terbentuk akan konvergen ke nilai minimal \mathbf{x}^* dari f .

(Nocedal dan Stephen, 1999).

Bukti:

Misal

$$m_k = \frac{\mathbf{y}_k^T \mathbf{s}_k}{\mathbf{s}_k^T \mathbf{s}_k} \quad \text{dan} \quad M_k = \frac{\mathbf{y}_k^T \mathbf{y}_k}{\mathbf{y}_k^T \mathbf{s}_k} \quad (2.36)$$

dan didefinisikan

$$m_k \geq m \quad \text{serta} \quad M_k \leq M. \quad (2.37)$$

Dengan menghitung *trace* dari persamaan (2.29) diperoleh

$$\text{trace}(H_{k+1}) = \text{trace}(H_k) + \frac{\|\mathbf{y}_k\|^2}{\mathbf{y}_k^T \mathbf{s}_k} - \frac{\|H_k \mathbf{s}_k\|^2}{\mathbf{s}_k^T H_k \mathbf{s}_k} \quad (2.38)$$

didefinisikan

$$\det(H_{k+1}) = \det(H_k) \frac{\mathbf{y}_k^T \mathbf{s}_k}{\mathbf{s}_k^T H_k \mathbf{s}_k} \quad (2.39)$$

$$\cos \theta_k = \frac{\mathbf{s}_k^T H_k \mathbf{s}_k}{\|\mathbf{s}_k\| \|H_k \mathbf{s}_k\|}, \quad q_k = \frac{\mathbf{s}_k^T H_k \mathbf{s}_k}{\mathbf{s}_k^T \mathbf{s}_k}. \quad (2.40)$$

Dari persamaan (2.39) dan (2.40) diperoleh

$$\frac{\|H_k \mathbf{s}_k\|^2}{\mathbf{s}_k^T H_k \mathbf{s}_k} = \frac{\|H_k \mathbf{s}_k\|^2 \|\mathbf{s}_k\|^2}{(\mathbf{s}_k^T H_k \mathbf{s}_k)^2} \frac{\mathbf{s}_k^T H_k \mathbf{s}_k}{\|\mathbf{s}_k\|^2} = \frac{q_k}{\cos^2 \theta_k} \quad (2.41)$$

dan

$$\det(H_{k+1}) = \det(H_k) \frac{\mathbf{y}_k^T \mathbf{s}_k}{\mathbf{s}_k^T \mathbf{s}_k} \frac{\mathbf{s}_k^T \mathbf{s}_k}{\mathbf{s}_k^T H_k \mathbf{s}_k} = \det(H_k) \frac{m_k}{q_k}. \quad (2.42)$$

Jika *trace* dan determinan dari matriks definit positif H digabungkan maka didefinisikan suatu fungsi sebagai berikut:

$$\psi(H) = \text{trace}(H) - \ln(\det(H)). \quad (2.43)$$

Dengan menggunakan persamaan (2.37) dan persamaan (2.38)-(2.43) diperoleh

$$\psi(H_{k+1}) = \text{trace}(H_{k+1}) - \ln(\det(H_{k+1})) \quad (2.44)$$

$$\begin{aligned} \psi(H_{k+1}) &= \psi(H_k) + M_k - \frac{q_k}{\cos^2 \theta_k} - \ln(\det(H_k)) - \ln m_k + \ln q_k \\ &= \psi(H_k) + (M_k - \ln m_k - 1) + \left[1 - \frac{q_k}{\cos^2 \theta_k} + \ln \frac{q_k}{\cos^2 \theta_k} \right] + \ln \cos^2 \theta_k. \end{aligned} \quad (2.45)$$

Pandang suatu fungsi yang berbentuk

$$h(t) = 1 - t + \ln t \leq 0, \text{ untuk } t > 0 \quad (2.46)$$

dengan menggunakan persamaan (2.37) maka persamaan (2.45) menjadi

$$0 < \psi(H_{k+1}) \leq \psi(H_k) + ck + \sum_{j=1}^k \ln \cos^2 \theta_j \quad (2.47)$$

dengan menganggap $c = M - \ln m - 1$ adalah positif.

Selanjutnya didefinisikan suatu hasil Zoutendijk yang diartikan sebagai limit untuk menentukan konvergen global, yaitu

$$\sum_{k=0}^{\infty} \cos^2 \theta_j \|\nabla f(\mathbf{x}_k)\|^2 < \infty \quad (2.48)$$

dengan mengingat kembali arah penurunan dari metode *Quasi Newton* dan mengartikan $\cos \theta_k$ sebagai sudut yang dibentuk antara arah penurunan *Gradient Descent* dengan arah penurunan dari *Quasi Newton*. Sehingga barisan yang dibentuk akan konvergen jika dan hanya jika $\cos \theta_k \rightarrow 0$.

Pembuktian dilakukan secara kontradiksi, dengan mengasumsikan $\cos \theta_k \rightarrow 0$, disana terdapat $k_1 > 0$ untuk semua $j > k_1$, maka

$$\ln \cos^2 \theta_j < -2c \quad (2.49)$$

dengan c adalah suatu konstanta yang telah didefinisikan di atas. Dengan menganggap bahwa persamaan (2.49) benar maka untuk $k > k_1$ dari persamaan (2.47) diperoleh

$$\begin{aligned}
0 < \psi(H_k) + ck + \sum_{j=1}^{k_1} \ln \cos^2 \theta_j + \sum_{j=k_1+1}^k (-2c_j) \\
= \psi(H_k) + \sum_{j=1}^{k_1} \ln \cos^2 \theta_j + 2ck_1 - ck.
\end{aligned} \tag{2.50}$$

Untuk k yang sangat besar maka ruas kanan dari pertidaksamaan di atas bernilai negatif, terjadi kontradiksi. Sehingga terdapat subbarisan dari indeks $\{jk\}$ sedemikian sehingga $\{\cos \theta_{jk}\} \geq \delta > 0$.

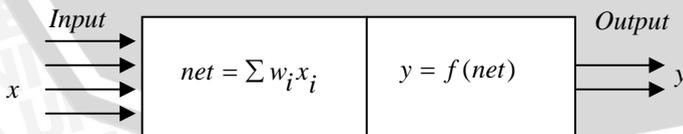
Berdasarkan hasil Zoutendjik, hal ini mengakibatkan $\liminf \|\nabla f(\mathbf{x}_k)\| \rightarrow 0$ sehingga terbukti $\mathbf{x}_k \rightarrow \mathbf{x}^*$.

2.2 JARINGAN SARAF TIRUAN

Jaringan saraf tiruan (JST) merupakan sistem pengolah informasi yang memiliki karakter seperti jaringan saraf pada otak manusia (Setiawan, 2003). JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi :

- pemrosesan informasi terjadi pada banyak elemen sederhana (neuron),
- sinyal dikirimkan antar neuron-neuron melalui penghubung-penghubung,
- penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal,
- untuk menentukan *output*, setiap neuron menggunakan fungsi aktivasi yang dikenakan pada jumlahan *input* yang diterima. (Siang, 2005).

Struktur JST yang mirip jaringan saraf manusia terdiri dari beberapa neuron yang bertugas untuk menerima informasi (*input*) dari neuron yang lain atau dari luar dengan bobot tertentu. Informasi tersebut akan diproses oleh suatu fungsi aktifasi dengan menjumlahkan nilai-nilai dari semua bobot yang datang (Kusumadewi, 2004). Hal ini seperti yang ditunjukkan oleh Gambar 2.2 yaitu



Gambar 2.2 Struktur Neuron Pada Jaringan Saraf

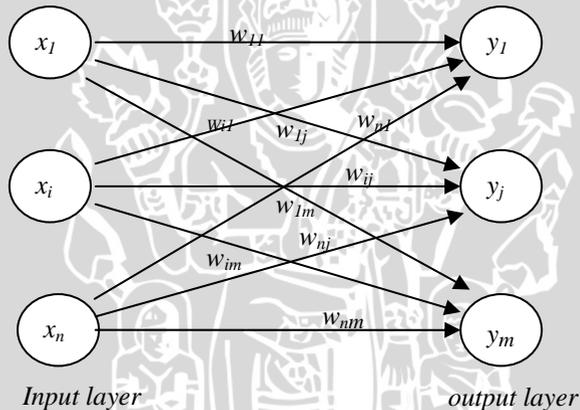
di mana: x adalah *input*
 y adalah *output*
 w adalah bobot
 $f(\text{net})$ adalah fungsi aktivasi

2.2.1 Arsitektur JST

Dalam arsitektur jaringan syaraf tiruan terdapat beberapa model diantaranya :

a. Jaringan Lapisan Tunggal (*Single Layer Network*)

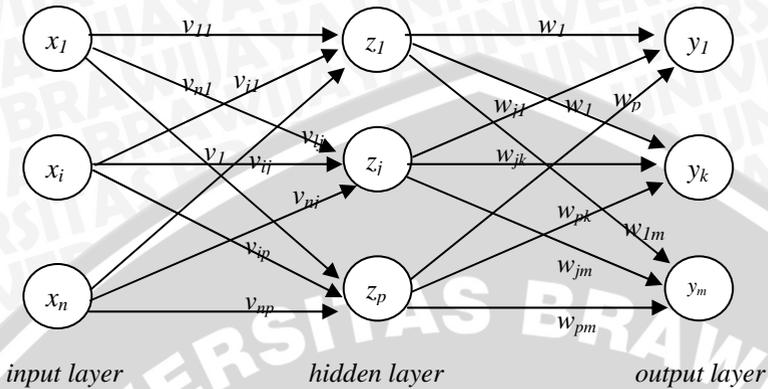
Dalam jaringan ini, neuron-neuron dapat dikelompokkan menjadi dua lapisan yaitu lapisan *input* (*input layer*) dan lapisan *output* (*output layer*). Pada Gambar 2.3 menunjukkan lapisan *input* yang memiliki n unit neuron dan lapisan *output* yang memiliki m unit neuron.



Gambar 2.3 Arsitektur Jaringan Saraf *Single Layer*

b. Jaringan Multi Lapisan (*MultiLayer Network*)

Jaringan ini merupakan perluasan dari jaringan lapisan tunggal. Diantara lapisan *input* dan lapisan *output* ditambahkan lapisan yang lain yang disebut sebagai lapisan tersembunyi (*hidden layer*), jumlah lapisan tersembunyi bisa lebih dari satu. Pada Gambar 2.4 ditunjukkan bahwa jaringan memiliki n unit *input* dan m unit *output* serta sebuah lapisan tersembunyi yang terdiri dari p buah unit.

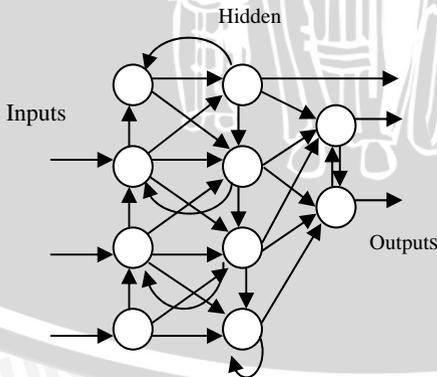


Gambar 2.4 Arsitektur Jaringan Saraf Multilayer

- di mana :
- i adalah indeks untuk *input*
 - j adalah indeks untuk *hidden*
 - k adalah indeks untuk *output*
 - x adalah *input*
 - y adalah *output*
 - z adalah *hidden*
 - w_{jk} adalah bobot antara *hidden layer* dan *output layer*
 - v_{ij} adalah bobot antara *input layer* dan *hidden layer*

c. Jaringan *Recurrent*

Model jaringan *recurrent* mirip dengan jaringan lapisan tunggal ataupun ganda. Hanya saja, ada neuron *output* yang memberikan sinyal pada unit *input* (*feedback loop*) (Siang, 2005). Berikut ini salah satu contoh yang menunjukkan arsitektur jaringan *recurrent* yang ditunjukkan pada Gambar 2.5.



Gambar 2.5 Arsitektur Jaringan *Recurrent*

2.2.2 Fungsi Aktivasi

Fungsi aktivasi digunakan untuk menentukan *output* dari JST. Di mana sebelum melakukan proses pengaktifan neuron-neuron akan dihitung melalui kombinasi linier antara *input* dengan bobotnya ditambah bobot dari luar atau lebih dikenal dengan bias, dengan bentuk :

$$f(\text{net}) = f(\sum x_i w_i + b). \quad (2.51)$$

Dalam JST dikenal tiga fungsi aktivasi yaitu :

1. Fungsi tangga

Bentuk fungsinya adalah

$$f(\text{net}) = \begin{cases} 1 & \text{jika net} \geq 0 \\ 0 & \text{jika net} < 0. \end{cases} \quad (2.52)$$

2. Fungsi Sigmoid biner

Bentuk fungsi sigmoid biner adalah

$$f(\text{net}) = \frac{1}{1 + e^{-\sigma \cdot \text{net}}}. \quad (2.53)$$

Dimana σ merupakan parameter kecuraman fungsi, fungsi sigmoid biner bernilai [0,1]. Fungsi ini paling sering digunakan dalam JST. Pada fungsi sigmoid terdapat juga fungsi yang bernilai [-1,1] atau dikenal sebagai fungsi sigmoid bipolar, dengan bentuk fungsi

$$f(\text{net}) = \frac{2}{1 + e^{-\text{net}}} - 1. \quad (2.54)$$

3. Fungsi Identitas

Bentuk fungsi identitas adalah

$$f(\text{net}) = \text{net}. \quad (2.55)$$

Fungsi identitas dipakai apabila nilai keluaran jaringan yang diinginkan berupa sembarang bilangan riil, bukan hanya pada range [0,1] atau [-1,1] (Siang, 2005).

2.2.3 Proses Pembelajaran

Dalam jaringan syaraf tiruan terdapat dua macam pembelajaran untuk memodifikasi bobotnya, yaitu pembelajaran terawasi (*Supervised learning*) dan pembelajaran tak terawasi (*Unsupervised learning*) (Siang, 2005).

1. Pembelajaran terawasi (*Supervised learning*)

Metode pembelajaran ini membutuhkan sejumlah pasangan data, yaitu data masukan dan target keluaran yang dipakai untuk melatih jaringan hingga diperoleh bobot yang diinginkan. Dalam setiap pembelajarannya masukan yang diberikan akan diproses untuk menghasilkan suatu keluaran. Selisih antara keluaran jaringan dengan target keluaran merupakan *error* yang terjadi. Jaringan akan memodifikasi bobot sesuai dengan kesalahan tersebut (Siang, 2005).

2. Pembelajaran tak terawasi (*Unsupervised learning*).

Pada pembelajaran ini tidak ada target keluaran yang digunakan sebagai acuan (Kusumadewi, 2004). Selanjutnya perubahan bobot yang terjadi pada saat pembelajaran dilakukan berdasarkan parameter tertentu yang telah dimodifikasi menurut ukuran tertentu (Siang, 2005).

Dalam skripsi ini metode pelatihan/pembelajaran yang digunakan adalah pembelajaran terawasi dengan model *Backpropagation* yang akan dibahas pada subbab selanjutnya. Sebelumnya, akan ditelaah terlebih dahulu mengenai *error* yang dihasilkan pada saat pelatihan JST. *Error* tersebut diartikan sebagai fungsi tujuan yang akan diminimumkan dengan perubahan bobot dan bias dari masing-masing lapisan yang didefinisikan sebagai

$$E_s = \frac{1}{2} \sum_{k=1}^m (t_k - y_k)^2 \quad (2.56)$$

dimana E_s adalah fungsi kesalahan pada data ke- s ($s = 1, 2, \dots$)

t_k adalah target

y_k adalah *output* jaringan

m adalah jumlah unit *output* (Patterson, 1996).

2.2.4 Pembelajaran Model *Backpropagation*

Model pembelajaran *Backpropagation* adalah model pembelajaran yang digunakan untuk jaringan multilayer. Metode optimasi yang biasa digunakan dalam pembelajaran *Backpropagation* adalah metode *Gradient Descent*. Dalam metode ini bobot jaringan bergerak sepanjang arah negatif dari *gradient* (Demuth, dkk. 1998).

Pembelajaran *Backpropagation* bertujuan untuk meminimumkan *error* yang terjadi. Selanjutnya dalam algoritma pembelajaran *Backpropagation* terdapat kriteria pemberhentian yang didefinisikan sebagai *Mean Square Error* (MSE) yaitu

$$MSE = \frac{\sum_{s=1}^r E_s}{r} \quad (2.57)$$

dengan r adalah jumlah dari semua data yang dilatih

(Patterson, 1996).

Pada dasarnya pelatihan *Backpropagation* terdiri dari tiga tahap yaitu

1. Propagasi maju (*feedforward*)
2. Propagasi mundur (*Backpropagation*)
3. Perubahan bobot.

Dimulai dari tahap *feedforward* dimana *input* (x_i) yang masuk pada *input layer* akan dikalikan dengan bobot (v_{ij}) yang terhubung dengan *hidden layer* dan menghasilkan z_j setelah dikenai fungsi aktivasi. Selanjutnya z_j dikalikan dengan bobot (w_{jk}) yang terhubung dengan *output layer* menghasilkan y_k .

Tahap selanjutnya yaitu pembelajaran *Backpropagation* yang menghitung kesalahan (δ_k) yang terjadi antara selisih dari ($t_k - y_k$) dimana t_k merupakan target yang diberikan. Nilai δ_k dapat menghitung perubahan bobot w_{jk} dan v_{ij} yang nantinya akan memperoleh bobot baru yang mampu meminimumkan *error* dari JST.

Algoritma pembelajaran *Backpropagation* yaitu :

Input : Data Pembelajaran (X), data target (T), V, V0, W, W0,
laju pembelajaran (α), error_mak, iterasi_mak.

Output : V, V0, W, W0, MSE, iterasi.

Algoritma

1. Inisialisasi bobot dengan bilangan acak yang kecil.
2. Tetapkan: Iterasi_mak, error_mak, laju pembelajaran (α).
3. Selama (iterasi < iterasi_mak) dan (MSE > error_mak), kerjakan langkah berikut:
 - a. Iterasi = Iterasi + 1
 - b. Untuk setiap pasangan elemen akan dilakukan pembelajaran sebagai berikut :

Feedforward :

- i. setiap unit masukan ($x_i, i = 1, 2, \dots, n$) menerima sinyal x_i dan meneruskannya ke unit tersembunyi.
- ii. Hitung jumlah sinyal-sinyal terbobot masukan di unit tersembunyi ($z_j, j = 1, 2, \dots, p$) :

$$z_{-in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \cdot \quad (2.58)$$

Selanjutnya gunakan fungsi aktivasi yang diinginkan untuk menghitung sinyal keluaran, yang berbentuk

$$z_j = f(z_{-in_j}) \cdot \quad (2.59)$$

- iii. Hitung jumlah dari sinyal-sinyal terbobot masukan di unit keluaran ($y_k, k = 1, 2, \dots, m$) :

$$y_{-in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \cdot \quad (2.60)$$

Selanjutnya gunakan fungsi aktivasi yang diinginkan untuk menghitung sinyal keluaran, yang berbentuk

$$y_k = f(y_{-in_k}) \cdot \quad (2.61)$$

Backpropagation :

- i. Hitung faktor δ di unit keluaran berdasarkan *error* yang terjadi ($y_k, k = 1, 2, \dots, m$) :

$$\delta_k = (t_k - y_k) f'(y_{-in_k}) \quad (2.62)$$

dimana δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layer sebelumnya. Kemudian hitung suku perubahan bobot w_{jk} dengan laju percepatan (*learning rate*) α yang berbentuk

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (2.63)$$

hitung juga perubahan bobot bias w_{0k} yaitu

$$\Delta w_{0k} = \alpha \delta_k \quad (2.64)$$

dengan $k = 1, 2, \dots, m$ dan $j = 0, 1, \dots, p$.

- ii. Setiap unit tersembunyi ($z_j, j = 1, \dots, p$) menjumlahkan delta masukannya dari setiap unit setelahnya yaitu

$$\gamma_{-in j} = \sum_{k=1}^m \delta_k w_{jk} \cdot \quad (2.65)$$

Untuk menghitung informasi kesalahannya, kalikan dengan turunan fungsi aktivasinya :

$$\gamma_j = \gamma_{-in j} f'(z_{-in j}) \cdot \quad (2.66)$$

Kemudian hitung suku perubahan bobot v_{ij} yang akan dipakai untuk merubah bobot v_{ij} :

$$\Delta v_{ij} = \alpha \gamma_j x_i \quad (2.67)$$

hitung juga perubahan bobot bias v_{0j} yaitu

$$\Delta v_{0j} = \alpha \gamma_j \quad (2.68)$$

dengan $j = 1, 2, \dots, p$ dan $i = 0, 1, \dots, n$.

Perubahan Bobot :

Perubahan bobot dari setiap unit keluaran (y_k) dan bias berbentuk

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \cdot \quad (2.69)$$

Perubahan bobot dari setiap unit *hidden* (z_j) dan bias berbentuk

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \cdot \quad (2.70)$$

4. Tes kondisi berhenti.

(Kusumadewi, 2004).

BAB III PEMBAHASAN

Sebelum mengulas lebih jauh mengenai penerapan dari subbab 2.1.1 dan 2.1.2 pada model pembelajaran *Backpropagation*, perlu diketahui bahwa notasi \mathbf{x} dan $f(\mathbf{x})$ dalam subbab 2.1.1 dan 2.1.2 akan digantikan oleh bobot w dan E_s secara berturut-turut dalam model pembelajaran *Backpropagation*.

3.1 Penerapan Metode Gradient Descent Pada Model Pembelajaran Backpropagation

Pada dasarnya model pembelajaran *Backpropagation* pada JST bertujuan untuk meminimumkan *error* yang terjadi dari selisih antara *output* jaringan dengan target yang diberikan. *Error* tersebut berbentuk

$$E_s = \frac{1}{2} \sum_{k=1}^m (t_k - y_k)^2 \quad (3.1)$$

dari persamaan (3.1) diperoleh penurunan algoritma pembelajaran *Backpropagation* sebagai berikut :

1. Perubahan bobot antara *output layer* dengan *hidden layer*

a) Perubahan bobot w_{jk}

$$\begin{aligned} \frac{\partial E_s}{\partial w_{jk}} &= \frac{\partial E_s}{\partial y_k} \frac{\partial y_k}{\partial y_{-in_k}} \frac{\partial y_{-in_k}}{\partial w_{jk}} \\ &= \frac{\partial}{\partial y_k} \left(\frac{1}{2} \sum_{k=1}^m (t_k - y_k)^2 \right) \frac{\partial}{\partial y_{-in_k}} (f(y_{-in_k})) \frac{\partial}{\partial w_{jk}} (w_{k0} + \sum_{j=1}^p z_j w_{kj}) \\ \frac{\partial E_s}{\partial w_{jk}} &= - \sum_{k=1}^m (t_k - y_k) f'(y_{-in_k}) z_j \end{aligned} \quad (3.2)$$

misal

$$\delta_k = \sum_{k=1}^m (t_k - y_k) f'(y_{-in_k}) \quad (3.3)$$

sehingga

$$\frac{\partial E_s}{\partial w_{jk}} = -\delta_k z_j \quad (3.4)$$

Dengan meninjau kembali persamaan (2.8) dimana arah penurunan dari metode *Gradient Descent* adalah negatif dari *gradient* sehingga perubahan bobot w_{jk} berbentuk

$$w_{jk}(t+1) = w_{jk}(t) + \alpha \delta_k z_j \quad (3.5)$$

dimana t merupakan iterasi selama proses perhitungan.

b) Perubahan bobot bias w_{0k}

$$\begin{aligned} \frac{\partial E_s}{\partial w_{0k}} &= \frac{\partial E_s}{\partial y_k} \frac{\partial y_k}{\partial y_{-in_k}} \frac{\partial y_{-in_k}}{\partial w_{0k}} \\ &= \frac{\partial}{\partial y_k} \left(\frac{1}{2} \sum_{k=1}^m (t_k - y_k)^2 \right) \frac{\partial}{\partial y_{-in_k}} \left(f(y_{-in_k}) \right) \frac{\partial}{\partial w_{jk}} \left(w_{k0} + \sum_{j=1}^p z_j w_{kj} \right) \\ \frac{\partial E_s}{\partial w_{0k}} &= - \sum_{k=1}^m (t_k - y_k) f'(y_{-in_k}) \end{aligned} \quad (3.6)$$

dengan menggunakan persamaan (3.3) diperoleh perubahan bobot bias w_{0k} yang berbentuk

$$w_{0k}(t+1) = w_{0k}(t) + \alpha \delta_k \cdot \quad (3.7)$$

2. Perubahan bobot antara *hidden layer* dengan *input layer*

a) Perubahan bobot v_{ij}

$$\begin{aligned} \frac{\partial E_p}{\partial v_{ij}} &= \frac{\partial E_p}{\partial y_k} \frac{\partial y_k}{\partial y_{-in_k}} \frac{\partial y_{-in_k}}{\partial z_j} \frac{\partial z_j}{\partial z_{-in_j}} \frac{\partial z_{-in_j}}{\partial v_{ij}} \\ &= \frac{\partial}{\partial y_k} \left(\frac{1}{2} \sum_{k=1}^m (t_k - y_k)^2 \right) \frac{\partial}{\partial y_{-in_k}} \left(f(y_{-in_k}) \right) \frac{\partial}{\partial z_j} \left(w_{0k} + \sum_{j=1}^p z_j w_{kj} \right) \\ &\quad \frac{\partial}{\partial z_{-in_j}} \left(f(z_{-in_j}) \right) \frac{\partial}{\partial v_{ij}} \left(v_{oj} + \sum_{i=1}^n x_i v_{ji} \right) \\ \frac{\partial E_p}{\partial v_{ij}} &= - \sum_{k=1}^m (t_k - y_k) f'(y_{-in_k}) w_{jk} f'(z_{-in_j}) x_i \end{aligned} \quad (3.8)$$

misal

$$\gamma_j = \sum_{k=1}^m (t_k - y_k) f'(y_{-in_k}) w_{jk} f'(z_{-in_j}) \quad (3.9)$$

dari persamaan (3.9) yang disubstitusikan pada persamaan (3.8) diperoleh

$$\frac{\partial E_s}{\partial v_{ij}} = -\gamma_j x_i \quad (3.10)$$

Dengan menggunakan metode yang sama yakni arah penurunan yang bernilai negatif dari *gradient*, sehingga perubahan bobot v_{ij} adalah

$$v_{ij}(t+1) = v_{ij}(t) + \alpha \gamma_j x_i \cdot \quad (3.11)$$

b) Perubahan bobot bias v_{0j}

$$\begin{aligned} \frac{\partial E_s}{\partial v_{0j}} &= \frac{\partial E_s}{\partial y_k} \frac{\partial y_k}{\partial y_{-in_k}} \frac{\partial y_{-in_k}}{\partial z_j} \frac{\partial z_j}{\partial z_{-in_j}} \frac{\partial z_{-in_j}}{\partial v_{0j}} \\ &= \frac{\partial}{\partial y_k} \left(\frac{1}{2} \sum_{k=1}^m (t_k - y_k)^2 \right) \frac{\partial}{\partial y_{-in_k}} \left(f(y_{-in_k}) \right) \frac{\partial}{\partial z_j} \left(w_{0k} + \sum_{j=1}^p z_j w_{kj} \right) \\ &\quad \frac{\partial}{\partial z_{-in_j}} \left(f(z_{-in_j}) \right) \frac{\partial}{\partial v_{ij}} \left(v_{0j} + \sum_{i=1}^n x_i v_{ji} \right) \\ \frac{\partial E_s}{\partial v_{0j}} &= - \sum_{k=1}^m (t_k - y_k) f'(y_{-in_k}) w_{jk} f'(z_{-in_j}) \end{aligned} \quad (3.12)$$

substitusikan persamaan (3.9) pada persamaan (3.12) diperoleh

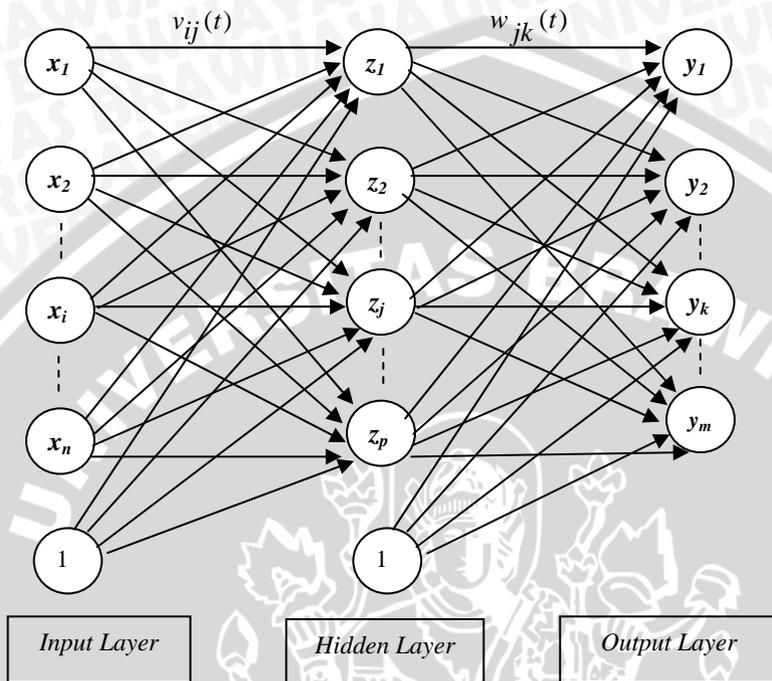
$$\frac{\partial E_s}{\partial v_{0j}} = -\gamma_j \quad (3.13)$$

pada akhirnya, perubahan bobot bias v_{0j} berbentuk

$$v_{0j}(t+1) = v_{0j}(t) + \alpha \gamma_j \cdot \quad (3.14)$$

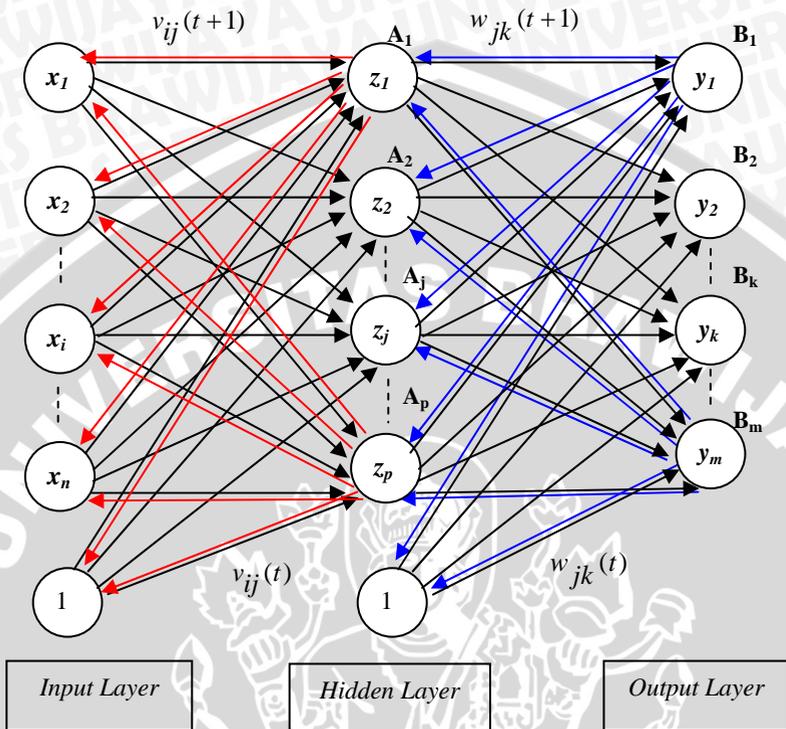
3.2 Penerapan Metode *Quasi Newton* Pada Model Pembelajaran *Backpropagation*

Konsep penerapan metode *Quasi Newton* sama seperti pada algoritma metode *Gradient Descent* yang menggunakan *error* sebagai fungsi tujuan, sehingga jika dilakukan penurunan parsial terhadap fungsi tersebut diperoleh hasil yang sama. Namun akan berbeda pada saat perubahan bobot dari masing-masing *layer*. Untuk lebih memahami penerapan metode *Quasi Newton* pada pembelajaran *Backpropagation*, perhatikan Gambar 3.1.



Gambar 3.1 Arsitektur JST dengan model pembelajaran *Backpropagation* secara umum

Gambar 3.1 merupakan JST yang memiliki arsitektur *multilayer* dengan model pembelajaran *Backpropagation*, di mana untuk masing-masing *layer* memiliki jumlah unit yang berbeda-beda. Dalam hal ini setiap unit akan menyampaikan informasi kepada unit selanjutnya dengan bobot tertentu, selanjutnya untuk perubahan bobot dari v_{ij} dan w_{jk} dijelaskan pada Gambar 3.2.



Gambar 3.2 Perubahan Bobot dari v_{ij} dan w_{jk}

Dari Gambar 3.2 terlihat bahwa perubahan bobot yang terjadi dari bobot v_{ij} dan bobot w_{jk} berbeda. Hal ini ditunjukkan dengan warna yang berbeda dari masing-masing layer. Warna biru menunjukkan perubahan bobot yang terjadi dari *output layer* ke *hidden layer* dan warna merah menunjukkan perubahan bobot yang terjadi dari *hidden layer* ke *input layer*. Masing-masing perubahan bobot sangat dipengaruhi oleh perhitungan terhadap pendekatan matriks *Hessian*. Selanjutnya perubahan bobot untuk masing-masing unit dipengaruhi oleh jumlah bobot yang masuk pada unit tersebut. Berikut penjabarannya :

1. Perubahan bobot antara *output layer* dan *hidden layer*, yaitu

$$w_{jk}(t+1) = w_{jk}(t) + \alpha B_k(t) \nabla E(w_{jk}(t)) \quad (3.15)$$

(Stanevski dan Dimiter, 2004).

dimana :

$w_{jk}(t+1)$ adalah bobot baru

$w_{jk}(t)$ adalah bobot lama

$\nabla E(w_{jk}(t))$ adalah *gradient* dari w_{jk}

$B_k(t)$ adalah pendekatan terhadap matriks *Hessian*

k adalah indeks pendekatan terhadap matriks *Hessian* untuk setiap unit pada *output layer*.

Dalam hal ini, pendekatan terhadap matriks *Hessian* akan terus *diupdate* dengan bentuk

$$B_k(t+1) = B_k(t) + \frac{\Delta(\nabla E(w_{jk}(t)))\Delta(\nabla E(w_{jk}(t)))^T}{\Delta(\nabla E(w_{jk}(t)))^T \Delta w_{jk}(t)} + \frac{\nabla E(w_{jk}(t))\nabla E(w_{jk}(t))^T}{\nabla E(w_{jk}(t))^T \left(\frac{\Delta w_{jk}(t)}{\alpha} \right)} \quad (3.16)$$

perubahan bobot w_{jk} pada persamaan (3.15) dapat dirubah ke dalam bentuk algoritma *Backpropagation* dengan menggunakan persamaan (3.4), yaitu

$$w_{jk}(t+1) = w_{jk}(t) + \alpha B_k(t)(\delta_k z_j(t)) \quad (3.17)$$

demikian pula untuk persamaan (3.16) menjadi

$$B_k(t+1) = B_k(t) + \frac{\Delta(-\delta_k z_j(t))\Delta(-\delta_k z_j(t))^T}{\Delta(-\delta_k z_j(t))^T \Delta w_{jk}(t)} + \frac{(-\delta_k z_j(t))(-\delta_k z_j(t))^T}{(-\delta_k z_j(t))^T \left(\frac{\Delta w_{jk}(t)}{\alpha} \right)} \quad (3.18)$$

2. Perubahan bobot antara *hidden layer* dan *input layer*, yaitu

$$v_{ij}(t+1) = v_{ij}(t) + \alpha A_j(t)\nabla E(v_{ij}(t)) \quad (3.19)$$

dimana

$v_{ij}(t+1)$ adalah bobot baru

$v_{ij}(t)$ adalah bobot lama

$\nabla E(v_{ij}(t))$ adalah *gradient* dari v_{ij}

$A_j(t)$ adalah pendekatan terhadap matriks *Hessian*

j adalah indeks pendekatan terhadap matriks *Hessian* untuk setiap unit pada *hidden layer*.

Pendekatan terhadap matriks *Hessian* akan terus diupdate dengan bentuk

$$A_j(t+1) = A_j(t) + \frac{\Delta(\nabla E(v_{ij}(t)))\Delta(\nabla E(v_{ij}(t)))^T}{\Delta(\nabla E(v_{ij}(t)))^T \Delta v_{ij}(t)} + \frac{\nabla E(v_{ij}(t))\nabla E(v_{ij}(t))^T}{\nabla E(v_{ij}(t))^T \left(\frac{\Delta v_{ij}(t)}{\alpha} \right)} \quad (3.20)$$

bentuk perubahan bobot v_{ij} pada persamaan (3.19) dapat dirubah ke dalam bentuk algoritma *Backpropagation* dengan menggunakan persamaan (3.10), yaitu

$$v_{ij}(t+1) = v_{ij}(t) + \alpha A_j(t)(\gamma_j x_i(t)) \quad (3.21)$$

demikian pula untuk persamaan (3.20) menjadi

$$A_j(t+1) = A_j(t) + \frac{\Delta(-\gamma_j x_i(t))\Delta(-\gamma_j x_i(t))^T}{\Delta(-\gamma_j x_i(t))^T \Delta v_{ij}(t)} + \frac{(-\gamma_j x_i(t))(-\gamma_j x_i(t))^T}{{(-\gamma_j x_i(t))^T \left(\frac{\Delta v_{ij}(t)}{\alpha} \right)}} \quad (3.22)$$

Algoritma pembelajaran *Backpropagation* dengan metode *Quasi Newton* yaitu

Input : Data Pembelajaran (X), data target (T), V0, W0, laju pembelajaran (α), error_mak, iterasi_mak.

Output : V0, W0, MSE, iterasi.

Algoritma

1. [m,n] ← ukuran (X)

[n,h] ← ukuran (V0)

[h+1,1] ← ukuran (W0)

B ← matrik identitas (ukuran h+1)

iterasi ← 0

MSE ← 1

2. for j=1 to h do
 - A (j) ← matrik identitas (ukuran n)
3. while (MSE > error_mak) and iterasi <= iterasi_mak do
 - a). Tot_error ← 0
 - b). iterasi ← iterasi+1
 - c). for k=1 to m do
 - 1) for j=1 to h do
 - i). $Z_in(j) \leftarrow (X(k,1)*V0(1,j))+(X(k,2)*V0(2,j))+$
 $(X(k,3)*V0(3,j))+ (X(k,4)*V0(4,j))+$
 $(X(k,5)*V0(5,j))+X(k,6)*V0(6,j)$
 - ii). $Z(j) \leftarrow [1/(1+\exp((-Z_in(j))^T))]$
 - 2) $Z \leftarrow [Z(1) Z(2) Z(3) Z(4) Z(5) \dots Z(h) 1]$
 - 3) $Y \leftarrow Z*W0$
 - 4) error ← T(k,1)-Y
 - 5) Dhu ← error
 - 6) Grad_W0 ← -Dhu*Z^T
 - 7) Up_W0 ← -Grad_W0
 - 8) for j=1 to h do
 - i). for i=1 to n do
 - a). $Grad_V0(i) \leftarrow -Dhu*W0(j,1)*(Z(1,j)*$
 $(1-Z(1,j)))*X(k,i)$
 - b). $Up_V0(i,j) \leftarrow -Grad_V0(i)$
 - ii). Hasil_Grad_V0(j) ← (Grad_V0)^T
 - iii). Hasil_Up_V0 ← Up_V0
 - 9) TOTAL_GRAD_V0 ← Hasil_Grad_V0
 - 10) TOTAL_Up_V0 ← Hasil_Up_V0
 - 11) V1 ← V0+alpha*TOTAL_Up_V0
 - 12) W1 ← W0+alpha*Up_W0
 - 13) for j=1 to h do
 - i). $Z_in(j) \leftarrow (X(k,1)*V1(1,j))+X(k,2)*V1(2,j))+$
 $(X(k,3)*V1(3,j))+ (k,4)*V1(4,j))+$
 $(X(k,5)*V1(5,j))+X(k,6)*V1(6,j)$
 - ii). $Z(j) \leftarrow [1/(1+\exp((-Z_in(j))^T))]$
 - 14) $Z \leftarrow [Z(1) Z(2) Z(3) Z(4) Z(5) \dots Z(h) 1]$

- 15) $Y \leftarrow Z * W1$
- 16) $error \leftarrow T(k,1) - Y$
- 17) $Dhu \leftarrow error$
- 18) $Grad_W1 \leftarrow -Dhu * Z^T$
- 19) $Per_Grad_W \leftarrow Grad_W1 - Grad_W0$
- 20) $Per_W \leftarrow W1 - W0$
- 21) $C \leftarrow Per_Grad_W * (Per_Grad_W)^T$
- 22) $D \leftarrow (Per_Grad_W)^T * Per_W$
- 23) $E \leftarrow Grad_W0 * (Grad_W0)^T$
- 24) $F \leftarrow (Grad_W0)^T * (Per_W / \alpha)$
- 25) $B1 \leftarrow B + (C/D) + (E/F)$
- 26) $Up_W1 \leftarrow -B1^{-1} * (Grad_W1)$
- 27) $cek_W \leftarrow (Grad_W1)^T * Up_W1$
- 28) if $cek_W > 0$ then
 - i). $B \leftarrow$ matrik identitas (ukuran $h+1$)
 - ii). $Up_W1 \leftarrow Grad_W0$
- 29) $B \leftarrow B1$
- 30) $W2 \leftarrow W1 + \alpha * Up_W1$
- 31) for $j=1$ to h do
 - i). for $i=1$ to n do
 - a. $Grad_V1(i) \leftarrow -Dhu * W1(j,1) * (Z(1,j) * (1 - Z(1,j))) * X(k,i)$
 - ii). $Hasil_Grad_V1(j) \leftarrow (Grad_V1)^T$
- 32) $TOTAL_GRAD_V1 \leftarrow Hasil_Grad_V1$
- 33) for $j=1$ to h do
 - i). $Per_Grad_V(j) \leftarrow TOTAL_GRAD_V1(j) - TOTAL_GRAD_V0(j)$
 - ii). $Per_Up_V(j) \leftarrow TOTAL_Up_V0(j)$
 - iii). $O \leftarrow Per_Grad_V(j) * (Per_Grad_V(j))^T$
 - iv). $P \leftarrow (Per_Grad_V(j))^T * Per_Up_V(j)$
 - v). $Q \leftarrow TOTAL_GRAD_V0(j) * (TOTAL_GRAD_V0(j))^T$

- vi). $R \leftarrow (TOTAL_GRAD_V0(j))^T * (Per_Up_V(j) / \alpha)$
- vii). $A1(j) \leftarrow A(j) + (O/P) + (Q/R)$
- viii). $Up_V \leftarrow -A1(j)^{-1} * TOTAL_GRAD_V1(j)$
- ix). $cek_V(j) \leftarrow (TOTAL_GRAD_V1)^T * Up_V$
- x). if $cek_V(j) > 0$ then
 - a). $A1(j) \leftarrow$ matrik identitas (ukuran n)
 - b). $Up_V \leftarrow TOTAL_GRAD_V0(j)$
- xi). $V2(j) \leftarrow V1(j) + \alpha * Up_V$

34) $A \leftarrow A1$

35) $V2_TOTAL \leftarrow V2$

36) $W0 \leftarrow W2$

37) $V0 \leftarrow V2_TOTAL$

d). $MSE(\text{iterasi}) \leftarrow Tot_error/m$

4. stop.

3.3 Membangun JST

Sebelum melakukan pembelajaran terhadap JST untuk mengenali pola dari data saham harian milik Bank Danamon, Tbk tahun 2004 (Lampiran 3), terlebih dahulu ditentukan faktor-faktor yang mempengaruhi JST, antara lain :

1. Arsitektur jaringan

Penentuan arsitektur jaringan tidak terlepas dari banyaknya variabel-variabel yang mempengaruhi data saham harian milik Bank Danamon, Tbk tahun 2004. Variabel-variabel tersebut antara lain: harga sebelum, harga tertinggi, harga terendah, harga penutupan, volume penjualan dan *stock* indeks. Dalam skripsi ini, variabel-variabel yang digunakan sebagai *input* jaringan berjumlah 5 variabel yaitu harga sebelum, harga tertinggi, harga terendah, volume penjualan dan *stock* indeks sedangkan variabel yang dijadikan sebagai *output* jaringan yaitu harga penutupan. Hal ini dilakukan untuk bisa mengambil keputusan, menjual atau membeli saham di hari berikutnya.

Sebaliknya untuk jumlah *hidden* dalam jaringan ditentukan secara uji coba (*trial and error*). Tujuannya untuk mencari arsitektur manakah yang paling sesuai untuk mengenali pola

secara akurat. Oleh karena itu, dalam skripsi ini diambil model arsitektur antara lain: 5-6-1, 5-9-1, 5-12-1 dan 5-17-1, artinya jaringan tersebut dibangun oleh 5 neuron di *input layer* dan 1 neuron di *output layer* sedangkan jumlah neuron di *hidden layer* berbeda-beda.

2. Fungsi aktivasi

Dalam skripsi ini, dipilih fungsi sigmoid untuk *hidden layer* dan fungsi identitas untuk *output layer*.

3. Laju pembelajaran

Pada skripsi ini, laju pembelajaran yang diberikan bersifat statis. Hal ini dimaksudkan untuk melihat bagaimana pengaruh yang diberikan terhadap kemampuan JST dalam mengenali pola. Dalam hal ini dipilih laju pembelajaran (α) antara lain: 0,05; 0,13; 0,2; 0,35 dan 0,6.

Berikutnya kriteria pemberhentian dari pembelajaran JST dengan model *Backpropagation* adalah target kesalahan sebesar 0,00001 dan jumlah iterasi maksimum sebesar 15000. Artinya jika pembelajaran JST telah mencapai salah satu dari kriteria tersebut maka proses pembelajaran akan berhenti, jika tidak maka pembelajaran akan terus berjalan.

3.4 Normalisasi Data

Sebelum data saham harian milik Bank Danamon, Tbk tahun 2004 (Lampiran 3) diproses ke dalam JST, data tersebut dinormalisasikan terlebih dahulu untuk mempermudah perhitungan. Dalam skripsi ini, total seluruh data yang digunakan berjumlah 241 data, dengan perincian 90% untuk data pelatihan dan 10% sisanya digunakan sebagai data test. Berikut rumus normalisasi yang digunakan

$$x_i^* = \frac{x_i - \mu}{\sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n - 1}}}$$

dimana : i adalah indeks untuk seluruh data

x_i adalah data yang belum dinormalkan

x_i^* adalah data yang telah dinormalkan

μ adalah rata-rata data (Walpole, 1995).

3.5 Hasil Pembelajaran

Penerapan metode *Gradient Descent* dan metode *Quasi Newton* pada pembelajaran *Backpropagation* memperlihatkan hasil yang berbeda, seperti yang dibahas pada subbab berikut ini.

3.5.1 Hasil Pembelajaran *Backpropagation* dengan Metode *Gradient Descent*

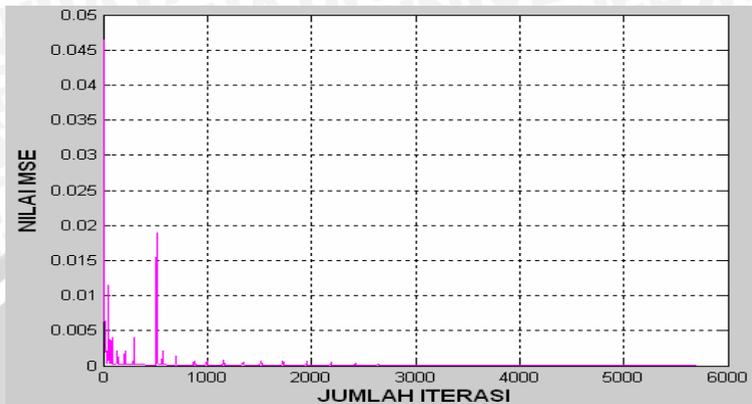
Hasil pembelajaran *Backpropagation* dengan model yang berbeda dan nilai laju pelatihan yang bersifat statis diperoleh hasil yang berbeda seperti yang terlihat pada Tabel 3.1.

Tabel 3.1 Hasil pembelajaran *Backpropagation* dengan metode *Gradient Descent*.

Model	Alpha	MSE	Iterasi ke-	Waktu(Detik)
5-6-1	0,05	9,9997e-006	2488	225,2500
	0,13	9,9965e-006	1384	125,0470
	0,2	9,9966e-006	2108	189,7810
	0,35	9,9457e-006	5704	531,7340
	0,6	1,6352e-005	15000	1,4821e+003
5-9-1	0,05	9,9914e-006	844	87,9220
	0,13	9,9988e-006	983	101,3280
	0,2	9,9992e-006	6458	691,3750
	0,35	9,9965e-006	4729	499,1250
	0,6	0,1080	15000	1,7035e+003
5-12-1	0,05	9,9864e-006	349	39,2820
	0,13	9,9710e-006	216	24,2030
	0,2	9,9980e-006	873	95,3280
	0,35	9,9993e-006	5907	677,8600
	0,6	9,9711e-006	11301	1,3097e+003
5-17-1	0,05	9,9607e-006	567	80,6120
	0,13	9,9966e-006	527	76,6100
	0,2	9,9957e-006	3630	522,9380
	0,35	9,9908e-006	1190	174,5000
	0,6	-	4	0,9530

Berdasarkan Tabel 3.1 dapat diperoleh nilai MSE terkecil dari masing-masing model yakni :

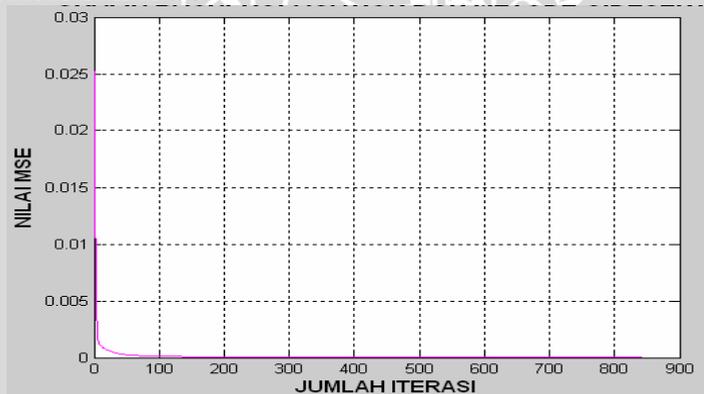
1. Model arsitektur 5-6-1.



Gambar 3.3 Hasil pembelajaran dari model 5-6-1 dengan $\alpha = 0,35$

Dari Gambar 3.3 terlihat perolehan nilai MSE terkecil untuk model arsitektur 5-6-1 dengan laju pembelajaran 0,35 yaitu $9,9457e-006$. JST berhenti melakukan pembelajaran pada iterasi ke-5704 dengan total waktu yang diperlukan selama 531,7340 detik. Artinya model arsitektur 5-6-1 mencapai semua kriteria pemberhentian yang diberikan. Selain itu, dari Gambar 3.3 terlihat bahwa pada awal iterasi grafik MSE tidak stabil namun ketika mencapai iterasi ke-3000 grafik MSE menjadi stabil.

2. Model arsitektur 5-9-1

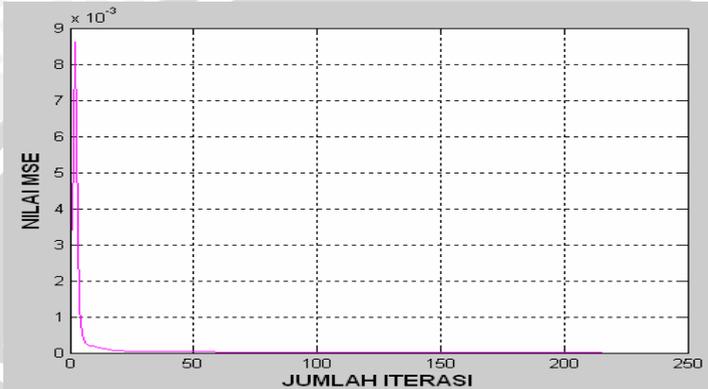


Gambar 3.4 Hasil pembelajaran dari model 5-9-1 dengan $\alpha = 0,05$

Dari Gambar 3.4 terlihat perolehan MSE terkecil untuk model arsitektur 5-9-1 dengan laju pembelajaran 0,05 adalah $9,9914e-006$. Berbeda dengan model sebelumnya pada model ini terlihat bahwa

mulai dari iterasi awal hingga akhir grafik MSE stabil dan JST berhenti melakukan pembelajaran pada iterasi ke-844 dengan total waktu yang diperlukan selama 87,9220 detik.

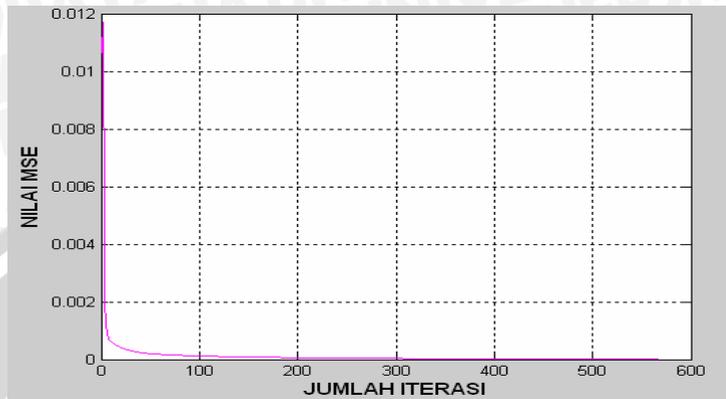
3. Model arsitektur 5-12-1



Gambar 3.5 Hasil pembelajaran dari model 5-12-1 dengan $\alpha = 0,13$

Dari Gambar 3.5 terlihat perolehan MSE terkecil untuk model arsitektur 5-12-1 dengan laju pembelajaran 0,13 yaitu $9,9710e-006$. Pada model ini, JST berhenti melakukan pembelajaran pada iterasi ke-216 dengan total waktu yang diperlukan selama 24,2030 detik. Dari Gambar 3.5 tersebut bisa dilihat pada awal iterasi MSE bergerak mencapai 9×10^{-3} namun setelah proses terus berjalan MSE menjadi stabil.

4. Model arsitektur 5-17-1



Gambar 3.6 Hasil pembelajaran dari model 5-17-1 dengan $\alpha = 0,05$

Dari Gambar 3.6 terlihat perolehan MSE terkecil untuk model arsitektur 5-17-1 dengan laju pembelajaran sebesar 0,05 yaitu 9,9607e-006. Dari gambar tersebut terlihat bahwa grafik MSE bergerak secara stabil dan berhenti pada iterasi ke-567 dengan total waktu yang diperlukan selama 80,6120 detik.

3.5.2 Hasil Pembelajaran *Backpropagation* dengan Metode *Quasi Newton*

Hasil pembelajaran *Backpropagation* dengan metode *Quasi Newton* dengan laju pembelajaran yang bersifat statis menghasilkan nilai seperti yang terdapat pada Tabel 3.2.

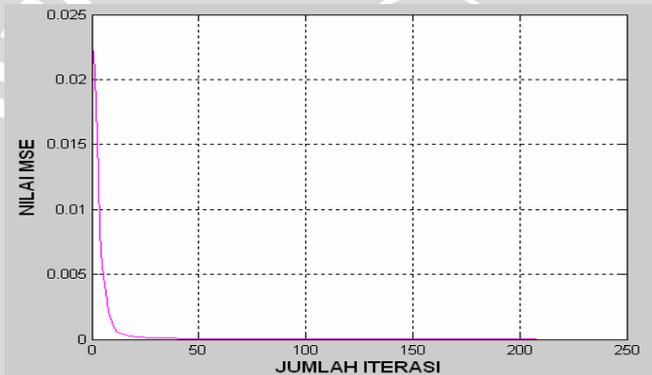
Tabel 3.2 Hasil pembelajaran *Backpropagation* dengan metode *Quasi Newton*

Model	Alpha	MSE	Iterasi ke-	Waktu(Detik)
5-6-1	0,05	9,9961e-006	1753	794,2650
	0,13	9,9653e-006	208	95
	0,2	9,9836e-006	2077	951,2810
	0,35	9,9872e-006	3974	1,8129e+003
	0,6	-	2	3,7340
5-9-1	0,05	9,9199e-006	90	56,3600
	0,13	9,9665e-006	146	120,2190
	0,2	-	28	21,8600
	0,35	9,9964e-006	4658	1,9031e+003
	0,6	-	1	4,6560
5-12-1	0,05	9,9741e-006	280	212,3440

	0,13	9,9383e-006	92	119,6560
	0,2	9,9369e-006	96	77,6090
	0,35	9,9797e-006	1291	1,0301e+003
	0,6	-	1	10,2340
5-17-1	0,05	9,9948e-006	114	125,7190
	0,13	9,5297e-006	24	27,2500
	0,2	9,5138e-006	118	128,9380
	0,35	-	1	5,4220
	0,6	-	1	19,3750

Berdasarkan Tabel 3.2 dapat ditelaah lebih jauh perolehan nilai MSE terkecil dari masing-masing model yakni:

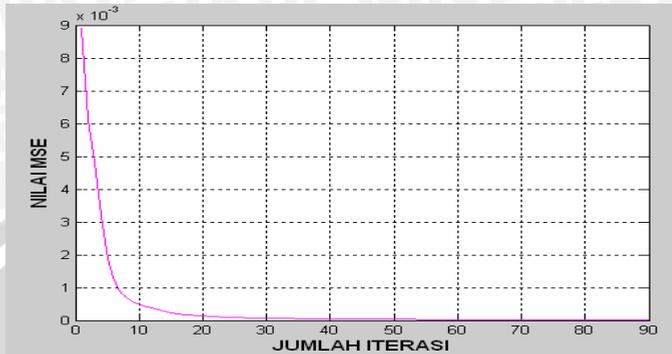
1. Model arsitektur 5-6-1



Gambar 3.7 Hasil pembelajaran dari model 5-6-1 dengan $\alpha = 0,13$

Dari Gambar 3.7 terlihat perolehan MSE terkecil untuk model arsitektur 5-6-1 dengan laju pembelajaran 0,13 yaitu 9,9653e-006. Pada model ini terlihat bahwa grafik MSE bergerak stabil mulai dari awal iterasi hingga akhir iterasi. JST berhenti melakukan pembelajaran pada iterasi ke-208 dengan total waktu yang diperlukan selama 95 detik.

2. Model arsitektur 5-9-1



Gambar 3.8 Hasil pembelajaran dari model 5-9-1 dengan $\alpha = 0,05$

Dari Gambar 3.8 dapat dilihat perolehan MSE terkecil untuk model arsitektur 5-9-1 dengan laju pembelajaran yang diberikan 0,05 yaitu 9,9199e-006. Pada model ini terlihat bahwa nilai MSE bergerak stabil dan sangat cepat karena nilai MSE tercapai pada iterasi ke-90 dengan total waktu yang diperlukan selama 56,3600 detik.

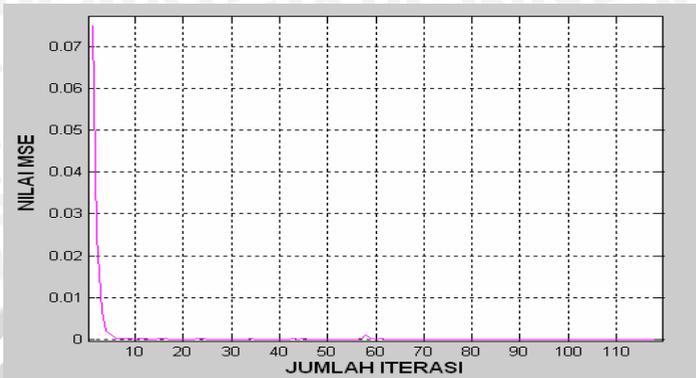
3. Model arsitektur 5-12-1



Gambar 3.9 Hasil pembelajaran dari model 5-12-1 dengan $\alpha = 0,2$

Dari Gambar 3.9 terlihat perolehan MSE terkecil untuk model arsitektur 5-12-1 dengan laju pembelajaran yang diberikan 0,2 yaitu 9,9369e-006. Dari gambar tersebut terlihat bahwa nilai MSE bergerak dengan cepat dan berhenti pada iterasi ke-96 dengan total waktu yang diperlukan selama 77,6090 detik.

4. Model arsitektur 5-17-1



Gambar 3.10 Hasil pembelajaran dari model 5-17-1 dengan $\alpha = 0,2$

Dari Gambar 3.10 terlihat perolehan MSE terkecil untuk model arsitektur 5-17-1 dengan laju pembelajaran 0,2 yakni $9,5138e-006$. Dari gambar tersebut terlihat bahwa nilai MSE dari model 5-17-1 bergerak stabil hingga berhenti pada iterasi ke-118 dengan total waktu yang diperlukan selama 128,9380 detik.

3.6 Analisa Perbandingan Hasil Pembelajaran

Berdasarkan Tabel 3.1 dan Tabel 3.2 diperoleh suatu perbandingan dari kedua metode, antara lain :

1. Nilai MSE

Dari seluruh model yang diberikan yakni 5-6-1, 5-9-1, 5-12-1 dan 5-17-1 menghasilkan nilai MSE yang berbeda-beda. Selanjutnya jika ditelaah lebih jauh nilai MSE terkecil terdapat pada metode *Quasi Newton* (Tabel 3.2) jika dibandingkan dengan metode *Gradient Descent* (Tabel 3.1). Namun dari kedua tabel tersebut dapat dilihat bahwa pada metode *Quasi Newton* JST sering gagal melakukan pembelajaran. Hal ini justru berbeda dengan metode *Gradient Descent* yang hanya gagal pada model arsitektur 5-17-1 dengan laju pembelajaran 0,6.

2. Jumlah iterasi

Pada metode *Gradient Descent* (Tabel 3.1) terlihat bahwa untuk seluruh model yang diberikan rata-rata JST mampu mengenali pola pembelajaran dengan jumlah iterasi yang mencapai ribuan. Rata-rata iterasi terbesar terjadi pada laju pembelajaran 0,6 yang mencapai 15.000 iterasi kecuali untuk model 5-17-1 karena JST gagal melakukan pembelajaran.

Selanjutnya pada metode *Quasi Newton* (Tabel 3.2) JST membutuhkan sedikit iterasi untuk mencapai target yang diberikan jika dibandingkan dengan metode *Gradient Descent*. Namun dari Tabel 3.2 dapat dilihat bahwa dengan laju pembelajaran 0,6 untuk semua model, JST gagal melakukan pembelajaran.

3. Laju pembelajaran

Pemilihan laju pembelajaran yang bersifat statis menghasilkan nilai yang berbeda untuk kedua metode. Untuk model arsitektur 5-6-1, 5-9-1, 5-12-1 dan 5-17-1 pada Tabel 3.1 hampir seluruh laju pembelajaran yang diberikan mampu mencapai kriteria yang diberikan. Kecuali pada laju pembelajaran 0,6 untuk model arsitektur 5-17-1 yang tidak mampu mencapai target MSE yang diberikan atau dengan kata lain jaringan gagal mengenali pola pembelajaran.

Hal ini justru berlainan untuk metode *Quasi Newton* yang disajikan dalam Tabel 3.2. Di mana laju pembelajaran harus dipilih secara hati-hati karena dapat dilihat untuk Tabel 3.2 masih banyak terjadi kegagalan karena JST tidak mampu mencapai target yang diberikan.

4. Efisiensi waktu

Dari hasil pembelajaran dengan metode *Gradient Descent* (Tabel 3.1) dapat dilihat bahwa untuk seluruh model yang diberikan membutuhkan waktu yang relatif lama jika dibandingkan dengan metode *Quasi Newton* (Tabel 3.2). Pada metode *Gradient Descent* total waktu terlama terjadi pada laju pembelajaran 0,6 kecuali untuk model arsitektur 5-17-1 yang hanya membutuhkan waktu 0,9530 detik. Hal ini terjadi karena JST gagal melakukan pembelajaran. Sebaliknya untuk metode *Quasi Newton* dengan laju pembelajaran 0,6 JST gagal melakukan pembelajaran untuk seluruh model yang diberikan.

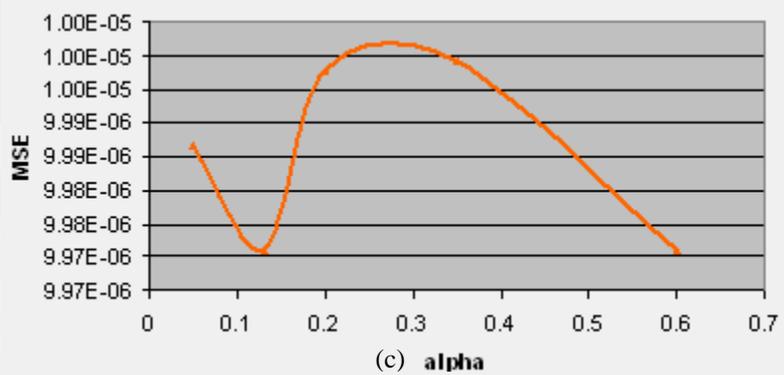
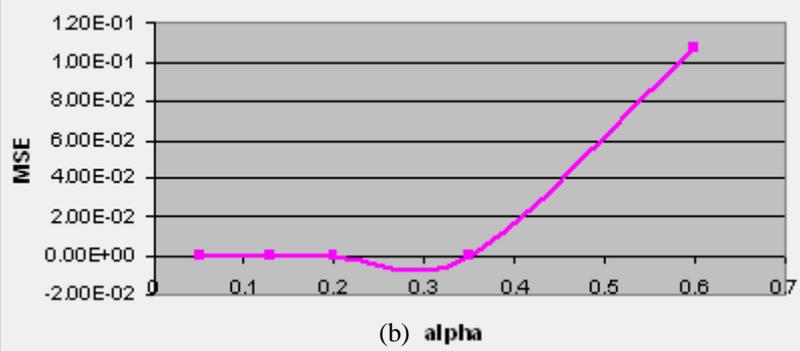
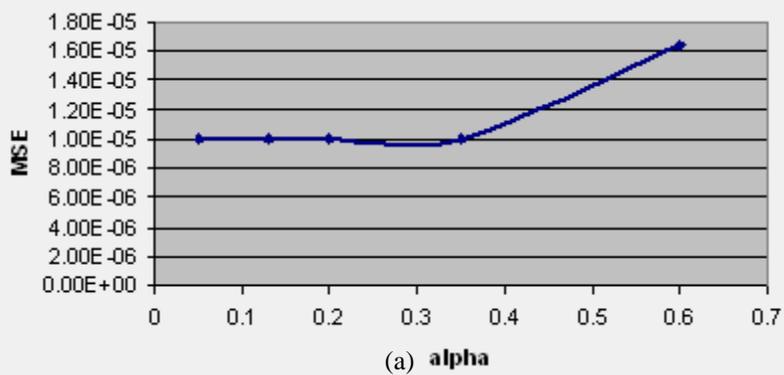
3.7 Pemilihan Model Arsitektur Terbaik

Pemilihan model arsitektur terbaik dilakukan untuk menentukan model arsitektur yang paling sesuai untuk mengenali pola data pembelajaran dari data saham harian Bank Danamon, Tbk (Lampiran 4). Dalam hal ini digunakan beberapa kriteria pemilihan arsitektur terbaik antara lain :

3.7.1 Nilai MSE

- 1). Metode *Gradient Descent*

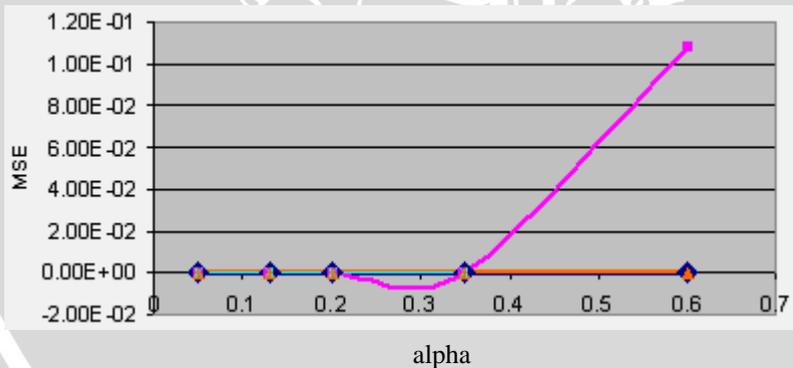
Berdasarkan Tabel 3.1 dapat dilihat perolehan MSE berdasarkan nilai laju pembelajaran (alpha) yang berbeda-beda untuk masing-masing model seperti yang ditunjukkan oleh Gambar 3.11.



(d)

Gambar 3.11 Hasil pembelajaran dari metode *Gradient Descent* dengan laju pembelajaran (alpha) yang berbeda
(a). Model 5-6-1 (b). Model 5-9-1 (c). Model 5-12-1 (d). Model 5-17-1

Dari Gambar 3.11 (a) dan (b) ditunjukkan bahwa semakin besar laju pembelajaran (alpha) membuat grafik MSE semakin tinggi. Namun hal ini berbeda untuk model 5-12-1 yang ditunjukkan oleh Gambar 3.11 (c) dimana pada saat laju pembelajaran 0,13 dan 0,6 grafik MSE menurun menjadi $9,9710e-006$ dan $9,9711e-006$. Selanjutnya Gambar 3.11 (d) menunjukkan bahwa model 5-17-1 dengan laju pembelajaran 0,6 JST gagal melakukan pembelajaran karena grafik MSE berhenti pada laju pembelajaran 0,35. Secara keseluruhan grafik MSE berdasarkan laju pembelajaran dari seluruh model ditunjukkan pada Gambar 3.12.



Gambar 3.12 MSE hasil pembelajaran dari seluruh model pada metode *Gradient Descent*

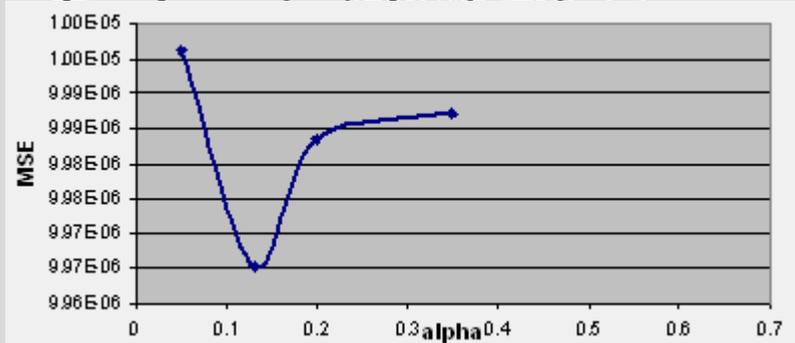
dimana

- Model 5-6-1
- Model 5-9-1
- Model 5-12-1
- Model 5-17-1

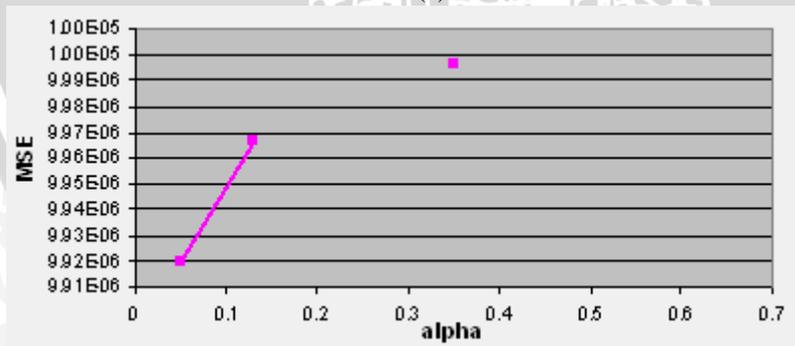
Dari Gambar 3.12 dapat dilihat bahwa grafik MSE dari seluruh model yang diberikan hanya model arsitektur 5-9-1 yang terlihat bergerak paling tinggi mencapai 0,1080. Selanjutnya untuk model 5-17-1 grafik MSE berhenti pada laju pembelajaran 0,35. Grafik MSE untuk model 5-6-1 dan 5-12-1 terlihat berhimpitan. Namun jika dipandang kembali Gambar 3.11 (a) dan (c) jelas terlihat model 5-12-1 lebih baik jika dibanding model 5-6-1.

2). Metode *Quasi Newton*

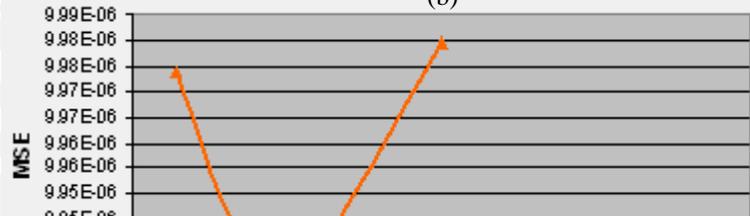
Berdasarkan Tabel 3.2 dapat dilihat perolehan MSE berdasarkan nilai laju pembelajaran (α) yang berbeda-beda untuk masing-masing model seperti yang ditunjukkan pada Gambar 3.13

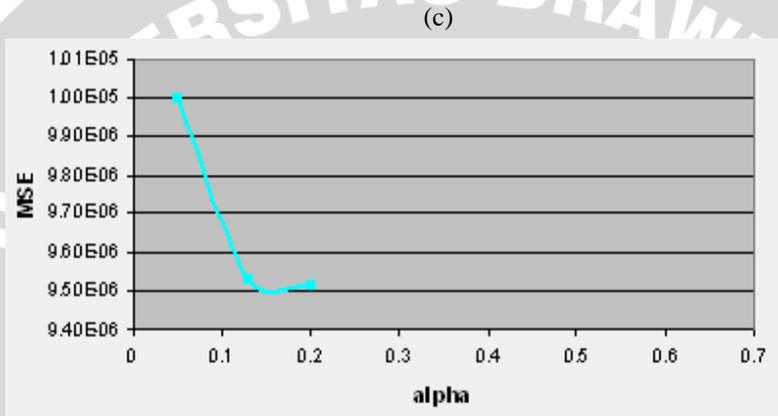


(a)



(b)



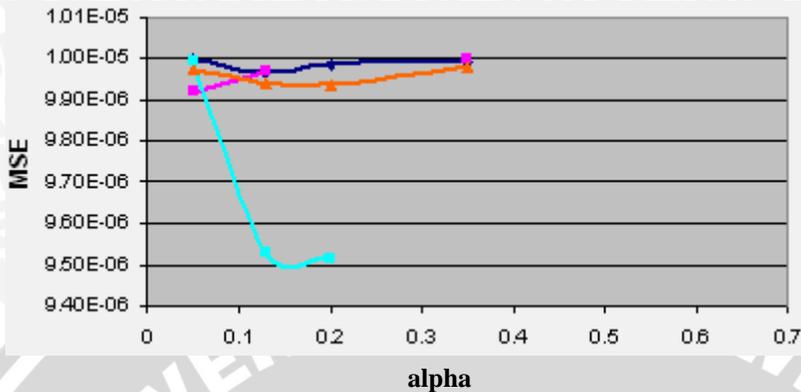


(d)

Gambar 3.13 Hasil pembelajaran dari metode *Quasi Newton* dengan laju pembelajaran (alpha) yang berbeda

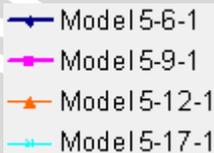
(a). Model 5-6-1 (b). Model 5-9-1 (c). Model 5-12-1 (d). Model 5-17-1

Dari Gambar 3.13 (a) dan (c) ditunjukkan bahwa dengan nilai laju pembelajaran yang berbeda menghasilkan grafik MSE yang berbeda pula. Selanjutnya untuk model arsitektur 5-9-1 grafik MSE terputus pada laju pembelajaran 0,13 karena pada laju pembelajaran 0,2 JST gagal melakukan pembelajaran seperti yang ditunjukkan pada Gambar 3.13 (b). Pada model arsitektur 5-17-1 dapat lihat bahwa grafik MSE berhenti pada laju pembelajaran 0,2 karena untuk laju pembelajaran 0,35 dan 0,6 JST gagal melakukan pembelajaran sehingga tidak diperoleh MSE pada saat pembelajaran, lihat Gambar 3.13 (d). Nilai MSE secara keseluruhan dapat dilihat pada Gambar 3.14.



Gambar 3.14 MSE hasil pembelajaran dari seluruh model pada metode *Quasi Newton*

dimana

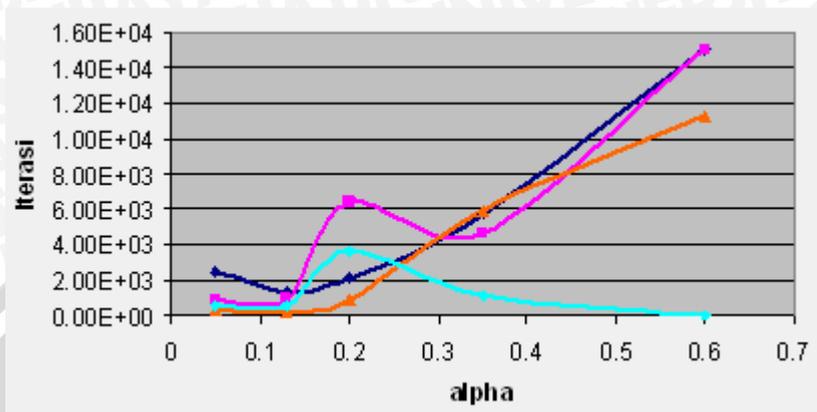


Dari Gambar 3.14 terlihat bahwa nilai MSE yang mengalami penurunan yakni pada model 5-17-1 namun model ini gagal melakukan pembelajaran pada alpha/laju pembelajaran 0,35 dan 0,6. Grafik MSE yang lain yang mampu melakukan pembelajaran dengan baik yakni model 5-12-1 karena pada alpha 0,2 model ini menghasilkan MSE paling kecil yakni $9,9369e-006$ jika dibandingkan dengan model 5-6-1 dan 5-9-1.

3.7.2 Jumlah iterasi

1). Metode *Gradient Descent*

Berdasarkan Tabel 3.1 dapat dilihat perolehan jumlah iterasi pada masing-masing model seperti yang ditunjukkan pada Gambar 3.15.



Gambar 3.15 Jumlah iterasi dari seluruh model dengan metode *Gradient Descent*

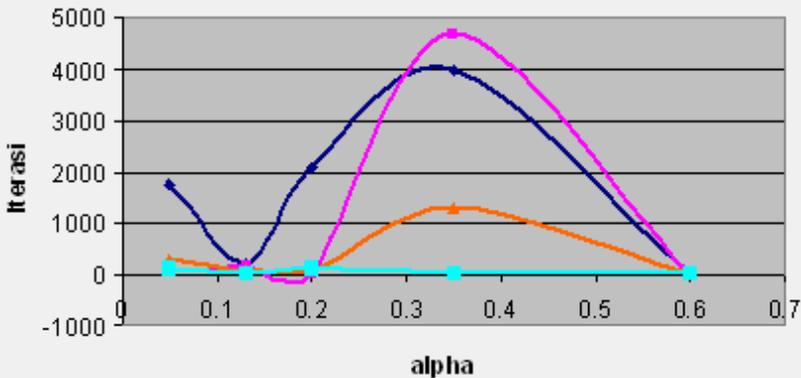
dimana



Dari Gambar 3.15 dapat dilihat bahwa untuk model 5-6-1, 5-9-1 dan 5-12-1 jumlah iterasi untuk laju pembelajaran yang semakin besar menghasilkan jumlah iterasi yang semakin besar pula. Namun berbeda dengan model 5-17-1 yang justru semakin kecil. Dari model arsitektur 5-6-1, 5-9-1 dan 5-12-1 dengan laju pembelajaran 0,6 terlihat bahwa iterasi paling sedikit terjadi pada model 5-12-1 yakni 11.301 iterasi.

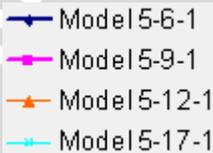
2). Metode *Quasi Newton*

Berdasarkan Tabel 3.2 diperoleh jumlah iterasi pada masing-masing model seperti yang ditunjukkan pada Gambar 3.16.



Gambar 3.16 Jumlah iterasi dari seluruh model pada metode *Quasi Newton*

dimana



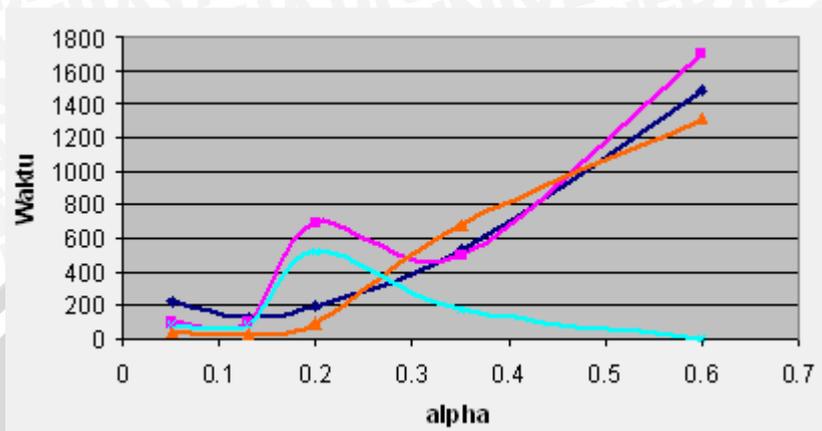
Dari Gambar 3.16 terlihat bahwa jumlah iterasi terbesar terjadi pada model 5-9-1 yang mencapai 4658 iterasi dengan laju pembelajaran 0,35. Selanjutnya untuk model 5-17-1 terlihat bahwa pada laju pembelajaran 0,35 dan 0,6 tampak tidak terjadi iterasi karena jumlah iterasi yang diperoleh berhimpit dengan garis nol. Hal ini terjadi karena pada saat ini nilai laju pembelajaran tersebut JST gagal melakukan pembelajaran.

Selanjutnya dari model 5-6-1, 5-9-1 dan 5-12-1 seperti yang tampak pada Gambar 3.16 terlihat bahwa model 5-12-1 membutuhkan sedikit iterasi untuk mencapai kriteria yang diberikan jika dibandingkan dengan model yang lain.

3.7.3 Total waktu

1). Metode *Gradient Descent*

Total waktu dari masing-masing model dapat dilihat pada Gambar 3.17.



Gambar 3.17 Total waktu dari seluruh model pada metode *Gradient Descent*

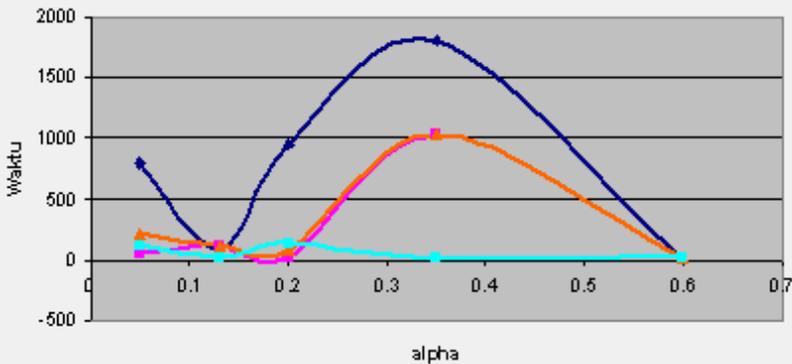
dimana



Dari Gambar 3.17 terlihat bahwa untuk model 5-6-1, 5-9-1 dan model 5-12-1 total waktu untuk alpha/laju pembelajaran yang semakin besar membutuhkan waktu yang semakin lama. Namun hal ini berbeda dengan model 5-17-1 yang justru membutuhkan waktu yang singkat karena JST gagal melakukan pembelajaran. Selanjutnya untuk model 5-6-1, 5-9-1 dan 5-12-1 terlihat bahwa dengan alpha/laju pembelajaran yang semakin besar dan waktu yang dibutuhkan tidak terlalu lama terjadi pada model 5-12-1 dimana total waktu terlamanya adalah $1,3097e+003$ detik.

2). Metode *Quasi Newton*

Total waktu yang diperlukan dari seluruh model yang ada ditunjukkan oleh Gambar 3.18.



Gambar 3.18 Total waktu dari seluruh model pada metode *Quasi Newton*

dimana



Gambar 3.18 menunjukkan bahwa total waktu terlama terjadi pada model arsitektur 5-6-1 dengan laju pembelajaran 0,35 yakni $1,8129e+003$ detik. Sedangkan pada model 5-17-1 dengan laju pembelajaran 0,35 dan 0,6 grafik seakan menyatu dengan garis nol. Hal ini terjadi karena JST gagal melakukan pembelajaran. Dari gambar tersebut juga terlihat untuk model 5-9-1 dan 5-12-1 grafik saling berhimpit. Namun jika memandang kembali Tabel 3.2 untuk laju pembelajaran 0,35 jelas waktu paling kecil terjadi pada model 5-12-1.

3.8 Hasil Proses Simulasi

Dari hasil pembelajaran yang telah dilakukan maka secara keseluruhan nilai MSE terkecil, jumlah iterasi yang sedikit dan waktu yang singkat terdapat pada model arsitektur 5-12-1. Tahap selanjutnya yaitu menguji kemampuan (Testing/Simulasi) JST untuk meramalkan harga penutupan dari data saham harian milik Bank Danamon, Tbk tahun 2004 dengan menggunakan model 5-12-1. Pada tahap ini JST menggunakan bobot awal (Lampiran 1) yang diperoleh dari pembelajaran. Berikut hasil simulasi yang diperoleh untuk masing-masing metode :

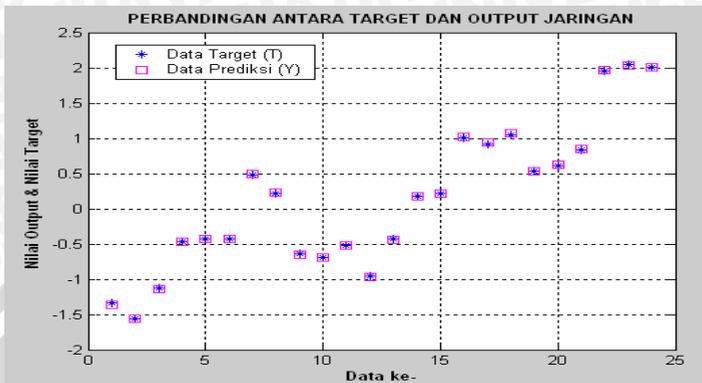
1. Metode *Gradient Descent*

Dengan menggunakan nilai awal (Lampiran 1) yang diperoleh dari proses pembelajaran dengan metode *Gradient Descent* diperoleh hasil simulasi yang disajikan dalam Tabel 3.3.

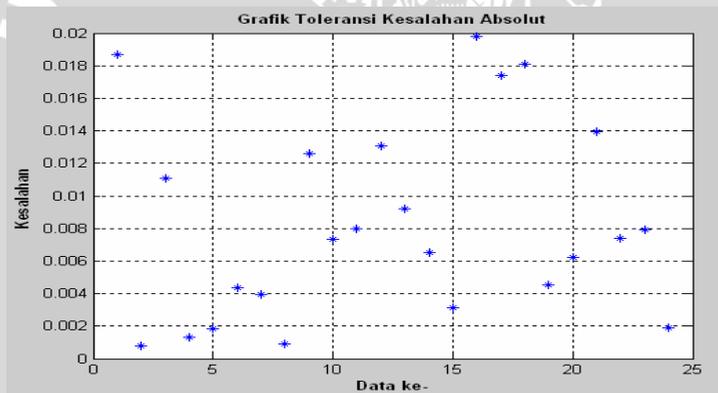
Tabel 3.3 Hasil simulasi menggunakan metode *Gradient Descent*

Data ke-	Harga Penutupan (T)	Harga Penutupan dari JST (Y)	Selisih
1.	-1,3390	-1,3577	0,0187
2.	-1,5560	-1,5568	7,7662e-004
3.	-1,1220	-1,1331	0,0111
4.	-0,4710	-0,4723	0,0013
5.	-0,4270	-0,4289	0,0019
6.	-0,4270	-0,4313	0,0043
7.	0,4840	0,4880	-0,0040
8.	0,2230	0,2239	-9,0986e-004
9.	-0,6440	-0,6566	0,0126
10.	-0,6880	-0,6953	0,0073
11.	-0,5140	-0,5220	0,0080
12.	-0,9480	-0,9611	0,0131
13.	-0,4270	-0,4362	0,0092
14.	0,1800	0,1735	0,0065
15.	0,2230	0,2198	0,0032
16.	1,0040	1,0238	-0,0198
17.	0,9180	0,9354	-0,0174
18.	1,0480	1,0661	-0,0181
19.	0,5270	0,5315	-0,0045
20.	0,6140	0,6202	-0,0062
21.	0,8310	0,8450	-0,0140
22.	1,9590	1,9664	-0,0074
23.	2,0460	2,0381	0,0079
24.	2,0020	2,0039	-0,0019
MSE			5,1987e-005

Berdasarkan Tabel 3.3 dapat dilihat selisih antara nilai target dan *output* dari jaringan terbesar adalah 0,0187 dan menghasilkan MSE 5,1987e-005. Selanjutnya dari Tabel 3.3 dapat digambarkan nilai dari target dan *output* jaringan yang ditunjukkan oleh Gambar 3.19 dan kesalahan yang terjadi antara nilai target dan *output* dari jaringan yang ditunjukkan oleh Gambar 3.20.



Gambar 3.19 Perbandingan target dengan *output* jaringan dengan metode *Gradient Descent* pada model arsitektur 5-12-1



Gambar 3.20 Selisih antara target dengan *output* jaringan

Berdasarkan Gambar 3.19 terlihat bahwa hasil prediksi yang dilakukan oleh JST cukup bagus karena mampu mendekati nilai sebenarnya. Hal ini terlihat dari bergabungnya simbol data prediksi dan data target menjadi satu. Selanjutnya kesalahan terbesar mencapai 0,0187 yang ditunjukkan oleh Gambar 3.20.

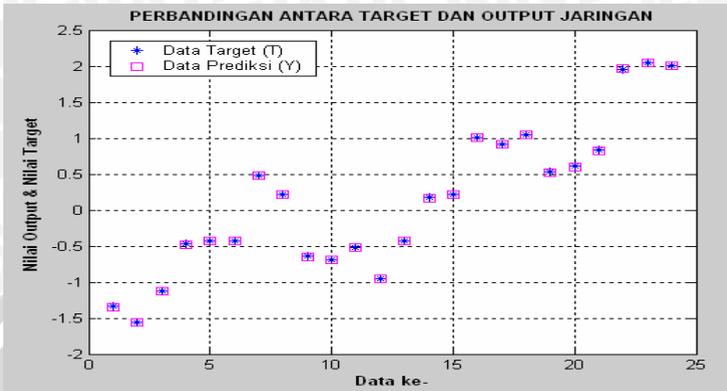
2. Metode *Quasi Newton*

Dengan menggunakan model arsitektur 5-12-1 dan nilai awal (Lampiran 1) yang diperoleh dari proses pembelajaran metode *Quasi Newton*. Selanjutnya dilakukan simulasi data yang baru dan diperoleh hasil yang disajikan dalam Tabel 3.4.

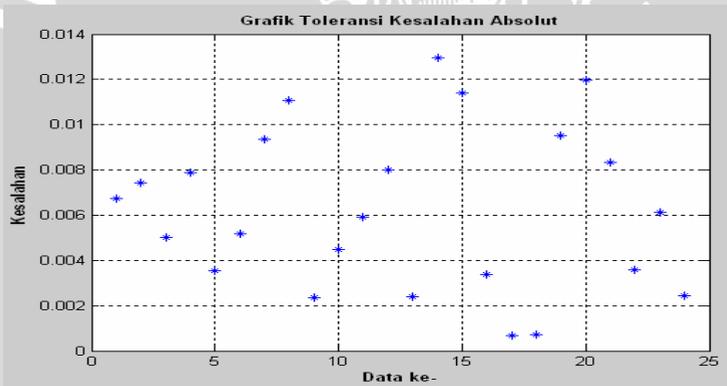
Tabel 3.4 Hasil simulasi menggunakan metode *Quasi Newton*

Data ke-	Harga Penutupan (T)	Harga Penutupan dari JST (Y)	Selisih
1.	-1,3390	-1,3457	0,0067
2.	-1,5560	-1,5634	0,0074
3.	-1,1220	-1,1270	0,0050
4.	-0,4710	-0,4789	0,0079
5.	-0,4270	-0,4305	0,0035
6.	-0,4270	-0,4322	0,0052
7.	0,4840	0,4747	0,0093
8.	0,2230	0,2119	0,0111
9.	-0,6440	-0,6464	0,0024
10.	-0,6880	-0,6925	0,0045
11.	-0,5140	-0,5199	0,0059
12.	-0,9480	-0,9560	0,0080
13.	-0,4270	-0,4294	0,0024
14.	0,1800	0,1671	0,0129
15.	0,2230	0,2116	0,0114
16.	1,0040	1,0074	-0,0034
17.	0,9180	0,9173	6,9273e-004
18.	1,0480	1,0473	7,1899e-004
19.	0,5270	0,5175	0,0095
20.	0,6140	0,6020	0,0120
21.	0,8310	0,8227	0,0083
22.	1,9590	1,9626	-0,0036
23.	2,0460	2,0399	0,0061
24.	2,0020	2,0044	-0,0024
	MSE		2,5757e-005

Dari Tabel 3.4 dapat dilihat selisih antara nilai target dan *output* dari jaringan terbesar adalah 0,0129 dan menghasilkan MSE 2,5757e-005. Selanjutnya dari Tabel 3.4 dapat digambarkan nilai dari target dan *output* jaringan yang ditunjukkan oleh Gambar 3.21 dan kesalahan yang terjadi antara nilai target dan *output* dari jaringan yang ditunjukkan oleh Gambar 3.22.



Gambar 3.21 Perbandingan target dengan *output* jaringan dengan Metode *Quasi Newton* pada model arsitektur 5-12-1



Gambar 3.22 Selisih antara target dengan *output* jaringan

Berdasarkan Gambar 3.21 dapat dikatakan bahwa hasil prediksi harga penutupan oleh JST bagus karena mampu mendekati nilai sebenarnya. Hal ini terlihat dari menyatunya grafik data target dan data prediksi. Selisih tertinggi yang terjadi antara data target dengan data prediksi yakni 0,0129 yang ditunjukkan oleh Gambar 3.22.

BAB IV PENUTUP

4.1 Kesimpulan

Berdasarkan hasil dan pembahasan, dapat disimpulkan:

1. Penerapkan metode yang berbeda pada algoritma pembelajaran *Backpropagation* yaitu:

a. Metode *Gradient Descent*

Perubahan bobot antara *output layer* dengan *hidden layer*

$$w_{jk}(t+1) = w_{jk}(t) + \alpha \delta_k z_j \cdot$$

Perubahan bobot bias antara *output layer* dengan *hidden layer*

$$w_{0k}(t+1) = w_{0k}(t) + \alpha \delta_k \cdot$$

Perubahan bobot antara *hidden layer* dengan *input layer*

$$v_{ij}(t+1) = v_{ij}(t) + \alpha \gamma_j x_i \cdot$$

Perubahan bobot bias antara *hidden layer* dengan *input layer*

$$v_{0j}(t+1) = v_{0j}(t) + \alpha \gamma_j \cdot$$

b. Metode *Quasi Newton*

Perubahan bobot antara *output layer* dengan *hidden layer*

$$w_{jk}(t+1) = w_{jk}(t) + \alpha B_k(t) (\delta_k z_j(t))$$

dengan

$$B_k(t+1) = B_k(t) + \frac{\Delta(-\delta_k z_j(t)) \Delta(-\delta_k z_j(t))^T}{\Delta(-\delta_k z_j(t))^T \Delta w_{jk}(t)} + \frac{(-\delta_k z_j(t)) (-\delta_k z_j(t))^T}{(-\delta_k z_j(t))^T \left(\frac{\Delta w_{jk}(t)}{\alpha} \right)}$$

Perubahan bobot antara *hidden layer* dengan *input layer*

$$v_{ij}(t+1) = v_{ij}(t) + \alpha A_j(t) (\gamma_j x_i(t))$$

dengan

$$A_j(t+1) = A_j(t) + \frac{\Delta(-\gamma_j x_i(t)) \Delta(-\gamma_j x_i(t))^T}{\Delta(-\gamma_j x_i(t))^T \Delta v_{ij}(t)} + \frac{(-\gamma_j x_i(t)) (-\gamma_j x_i(t))^T}{((-\gamma_j x_i(t))^T \left(\frac{\Delta v_{ij}(t)}{\alpha} \right))}$$

2. Perbandingan model *Backpropagation*

a). Metode *Gradient Descent*

Metode ini rata-rata menghasilkan nilai MSE yang cukup kecil namun membutuhkan waktu yang relatif lama dengan jumlah iterasi yang besar. Untuk tahap simulasi metode ini mampu memprediksi harga penutupan pada data saham harian milik Bank Danamon, Tbk tahun 2004 dengan MSE yakni $5,1987e-005$.

b). Metode *Quasi Newton*

Metode ini mampu mencapai kriteria yang diberikan dengan nilai MSE yang kecil, jumlah iterasi yang sedikit dan waktu pembelajaran yang cepat. Namun metode ini sangat sensitif terhadap laju pembelajaran yang bersifat statis karena jika terlalu besar, metode ini akan gagal dalam melatih jaringan. Dari hasil simulasi, metode ini mampu memprediksi harga penutupan data saham harian milik Bank Danamon, Tbk tahun 2004 dengan MSE yakni $2,5757e-005$.

4.2 Saran

Pada penulisan tugas akhir banyak terdapat kekurangan pada metode *Quasi Newton*. Hal ini dapat terlihat pada pemberian laju pelatihan yang bersifat statis sehingga banyak membuat pelatihan kurang optimal. Oleh karena itu, untuk pengembangan yang lebih baik laju pelatihan dapat dicari dengan menggunakan metode *line search* seperti *Backtracking*.

DAFTAR PUSTAKA

- Anton, H. 1987. *Aljabar Linear Elementer*. Erlangga. Jakarta.
- Boyd, S dan Lieven, V. 2004. *Convex Optimization*. Cambridge University Press. United Kingdom.
- Bronshstein, I.N. dkk. 2007. *Handbook of Mathematics*. Springer. Berlin.
- Buchanan J.L. dan Peter, R.T. 1992. *Numerical Methods and Analysis*. Mc. Graw-Hill. Singapore.
- Demuth, H. dkk. 1998. *Neural Network Toolbox User's Guide*. The Mathwork, Inc.
- Fallgren, M. 2006. *On The Robustness of Conjugate-Gradient methods and Quasi-Newton methods*. Department of Mathematics Royal Institute of Technology.
- Freund, R. 2004. *The Steepest Descent Algorithm for Unconstrained Optimization*. Massachusetts Institute of Technology.
- Gill, P.E. dkk. 1981. *Practical Optimization*. Academic Press. New York.
- Koivo, N.H. 2006. *Neural Network: Basics using Matlab Neural Network Toolbox*.
- Kusumadewi, S. 2004. *Membangun Jaringan Syaraf Tiruan Menggunakan Matlab dan Excel Link*. Graha Ilmu. Yogyakarta.
- Leader, J.J. 2004. *International Edition Numerical Analysis and Scientific Computation*. Pearson Education, Inc. USA.
- Lewin, J dan Lewin, M. 1993. *An Introduction to Mathematical Analysis*. McGraw-Hill, Inc. Singapore.
- Mathews, J.H dan Kurtis D.F. 2004. *International edition Numerical Methods Using Matlab Fourth Edition*. Pearson Education International. USA.
- Nocedal, J. dan Stephen, J. W. 1999. *Numerical Optimization*. Springer-Verlag New York, Inc. USA.
- Patterson, D.W. 1996. *Artificial Neural Networks Theory and Applications*. Prentice Hall Simon and Schusler Pte Ltd. Singapore.
- Setiawan, K. 2003. *Paradigma Sistem Cerdas*. Bayumedia Publishing. Malang.
- Siang, J. H. 2005. *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab*. Andi. Yogyakarta.

Stanevski, N dan Dimiter, T. 2004. *On The Quasi Newton Training Method for Feed-Forward Neural Networks*. International Conference on Computer System and Technologies.

Walpole, R.E. 1995. *Pengantar Statistika Edisi ketiga*. Gramedia Pustaka Utama. Jakarta.

UNIVERSITAS BRAWIJAYA

