

**PENERAPAN JARINGAN SYARAF TIRUAN
BACKPROPAGATION DALAM PREDIKSI TREND
HARGA SAHAM**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar
Sarjana dalam bidang Ilmu Komputer

oleh:

MUHAMMAD ALKAFF

0410963038-96



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2009**

UNIVERSITAS BRAWIJAYA



LEMBAR PENGESAHAN SKRIPSI
PENERAPAN JARINGAN SYARAF TIRUAN
***BACKPROPAGATION* DALAM PREDIKSI *TREND* HARGA**
SAHAM.

Oleh:
MUHAMMAD ALKAFF
0410963038-96

Telah dipertahankan di depan Majelis Penguji
Pada tanggal 11 November 2009
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Komputer pada Program S-1 Ilmu Komputer

Pembimbing I

Pembimbing II

Drs. Marji, MT
NIP. 196708011992031001

Edy Santoso, S.Si.,M.Kom
NIP. 197404142003121004

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Agus Suryanto, MSc
NIP. 196908071994121001

UNIVERSITAS BRAWIJAYA



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Alkaff
NIM : 0410963038
Jurusan : Matematika
Penulis Skripsi berjudul : Penerapan Jaringan Syaraf Tiruan
Backpropagation Dalam Prediksi
Trend Harga Saham

Dengan ini menyatakan bahwa:

1. Isi dari 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 dikemudian hari ternyata Skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 11 November 2009

Yang Menyatakan,

(Muhammad Alkaff)

NIM. 0410963038

UNIVERSITAS BRAWIJAYA



PENERAPAN JARINGAN SYARAF TIRUAN *BACKPROPAGATION* DALAM PREDIKSI *TREND* HARGA SAHAM

ABSTRAK

Data saham merupakan data yang diukur terurut berdasarkan waktu. Jika data tersebut dimodelkan, maka dapat terlihat bahwa data tersebut memiliki *trend*. Dengan cara tersebut maka kemunculan data berikutnya dapat diprediksi. Pada penelitian ini disajikan suatu metode dengan menggunakan algoritma jaringan syaraf tiruan (JST) *backpropagation* untuk melakukan prediksi terhadap *trend* harga saham sehingga dapat memprediksi arah pergerakan harga saham. Metode yang digunakan pada penelitian ini yaitu dengan menggunakan data harga saham selama seminggu ditambah nilai faktor siklus (*trend* harga saham setahun sebelumnya) untuk memprediksi *trend* harga saham seminggu berikutnya. Secara umum sistem ini memiliki 2 bagian pemrosesan. Proses pertama adalah proses normalisasi terhadap data harga saham sehingga range data input menjadi 0.1-0.9. Data inilah yang dijadikan sebagai *input* kedalam JST. Proses kedua adalah proses pelatihan terhadap data *training* dan peramalan terhadap *trend* yang terjadi pada minggu berikutnya dengan menggunakan harga saham selama seminggu yang diinputkan oleh *user*. Untuk mendapatkan struktur JST yang terbaik, dilakukan pelatihan dengan beberapa parameter diantaranya jumlah *neuron* pada *hidden layer* dan laju pembelajaran (*learning rate*). Tolak ukur keberhasilan sistem ini adalah dengan menghitung nilai *mean squared error* (MSE). Data yang digunakan adalah data harga penutupan saham Astra, Sampoerna, dan Indofood periode Januari – Desember 2008. Dari pengujian didapatkan MSE untuk saham Astra 0.03268, untuk saham Sampoerna 0.06366 dan saham Indofood 0.8703. Sedangkan keakuratan hasil prediksi untuk data uji adalah 25% dari data uji Astra, 50% dari data uji Sampoerna dan 33.33 % dari data uji Indofood.

UNIVERSITAS BRAWIJAYA



IMPLEMENTATION OF BACKPROPAGATION NEURAL NETWORK FOR FORECASTING STOCK PRICE TREND

ABSTRACT

Stock data are measured data sorted by time. If the data is modeled, it can be seen that the data has a trend. In this way the emergence of the next data can be predicted. In this research presented a method using artificial neural network algorithm backpropagation to perform prediction of stock price trends that can predict the direction of stock price movement. The method used in this study is to use stock price data for the week plus a cycle factor value (stock price trends a year before) to predict the trend of stock prices next week. In general, this system has 2 parts processing. The first process is the normalization process of the stock price data so that the input data range to be 0.1-0.9. This data was used as input into neural network. The second process is the training and forecasting data for trends that occur in the next week using the stock price during the week as an input from user. To get the best neural network structure, conducted training with several parameters including the number of neurons in the hidden layer and learning rate. The success measuring rod of this system is by calculating the mean squared error. The data used is the stock's closing price data Astra, Sampoerna, and Indofood period January to December 2008. MSE obtained from the test for 0.03268 shares of Astra, for 0.06366 shares of Sampoerna and 0.8703 shares of Indofood. Whereas the accuracy of prediction for test data is 25% of the Astra test data, 50% of Sampoerna test data and 33.33% of Indofood test data.

UNIVERSITAS BRAWIJAYA



Kata Pengantar

Alhamdulillah rabbil 'alamin. Puji syukur penulis panjatkan kehadiran Allah SWT, karena atas segala rahmat dan limpahan hidayahNya, Tugas Akhir yang berjudul “Penerapan Jaringan Syaraf Tiruan Dalam Prediksi Trend Saham Menggunakan Metode *Backpropagation*” ini dapat diselesaikan. Tugas Akhir ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada program studi Ilmu Komputer, jurusan Matematika, fakultas MIPA, universitas Brawijaya.

Semoga Allah melimpahkan rahmat atas Nabi Muhammad SAW, makhluk paling mulia yang senantiasa memberikan cahaya petunjuk, dan atas keluarganya dan sahabat-sahabatnya..

Dalam penyelesaian tugas akhir ini, penulis telah mendapat begitu banyak bantuan baik moral maupun materiil dari banyak pihak. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

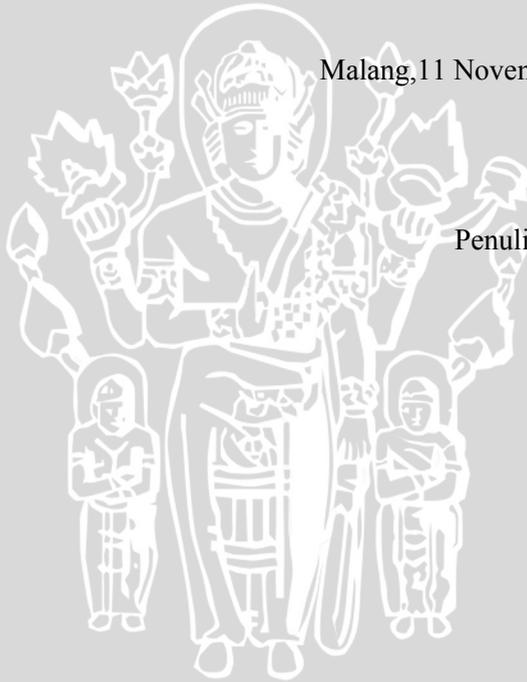
1. Drs, Marji M.T selaku Pembimbing I dan Ketua Program Studi Ilmu Komputer. Terima kasih atas semua waktu dan bimbingan yang telah diberikan.
2. Edy Santoso, S.Si, M.Kom selaku Pembimbing II. Terima kasih atas semua waktu dan bimbingan yang telah diberikan.
3. Bondan Sapta Prakoso, ST selaku Penasehat Akademik.
4. Dr. Agus Suryanto, M.Sc selaku Ketua Jurusan Matematika.
5. Segenap bapak dan ibu dosen yang telah mendidik dan mengamalkan ilmunya kepada penulis.
6. Segenap staf dan karyawan di Jurusan Matematika FMIPA Universitas Brawijaya.
7. *Abi, Ummi* dan adik-adik tercinta. Terima kasih atas cinta, kasih sayang, doa, dukungan dan semangat yang tiada henti.
8. *Freeze game center community*. Terima kasih atas hiburan dan kesenangannya.
9. Sahabat- sahabat Ilkom `04. Terima kasih atas senyuman, semangat dan hari-hari kita.
10. Warga dan Eks-Warga Sumber Sari 93. Terima kasih telah menjadi bagian perjalanan ini.

11. Pihak lain yang telah membantu terselesaikannya Tugas Akhir ini yang tidak bisa penulis sebutkan satu-persatu.

Semoga penulisan laporan tugas akhir ini bermanfaat bagi pembaca sekalian. Dengan tidak lupa kodratnya sebagai manusia, penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan, dan mengandung banyak kekurangan, sehingga dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang membangun dari pembaca.

Malang, 11 November 2009

Penulis



DAFTAR ISI

Halaman

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR LAMPIRAN	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Metodologi Pemecahan Masalah	3
1.7 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	5
2.1 SAHAM	5
2.1.1 Pengertian Saham	5
2.1.2 Jenis Jenis Saham	6
2.1.3 Faktor Yang Mempengaruhi Pergerakan Saham	6
2.1.4 <i>Trend</i> Pasar	7
2.2 <i>Time Series Forecasting</i>	8
2.3 Jaringan Syaraf Tiruan	8
2.3.1 Pengertian Jaringan Syaraf Tiruan	9
2.3.2 Aplikasi Jaringan Syaraf Tiruan	9
2.3.3 Model Dasar Jaringan Syaraf Tiruan	11
2.3.4 Lapisan Pada Jaringan Syaraf Tiruan	13
2.3.5 Proses Pembelajaran	14
2.3.6 Fungsi Transfer	14
2.3.7 Jaringan Syaraf Backpropagation	15
2.3.7.1 Algoritma Jaringan Syaraf Tiruan	16
2.3.7.2 Karakteristik Jaringan Syaraf Backpropagation	17

2.4 Normalisasi	19
BAB III METODOLOGI DAN PERANCANGAN SISTEM	21
3.1 Studi Literatur	22
3.2 Data Yang Digunakan	22
3.3 Deskripsi Umum Sistem	22
3.4 Perancangan Proses	23
3.4.1 Proses Normalisasi Data	24
3.4.2 Pelatihan Jaringan Syaraf Tiruan	25
3.4.3 Proses Prediksi <i>Trend</i> Saham	27
3.5 Menentukan Input dan Output Data	28
3.6 Pertimbangan Faktor Siklus	28
3.7 Perancangan Uji Coba	29
3.8 Contoh Perhitungan	30
3.9 Perancangan Antar Muka	41
BAB IV IMPLEMENTASI DAN PEMBAHASAN	45
4.1 Lingkungan Implementasi	45
4.1.1 Lingkungan Perangkat Keras	45
4.1.2 Lingkungan Perangkat Lunak	45
4.2 Implementasi Program	45
4.2.1 Struktur Data	45
4.2.2 Proses Mencari Nilai Minimum	46
4.2.3 Proses Mencari Nilai Maksimum	47
4.2.4 Proses Normalisasi	48
4.2.5 Proses Menghitung Faktor Skala (β)	49
4.2.6 Proses Inialisasi Bobot	49
4.2.7 Proses Menghitung Nilai Aktivasi (<i>Sigmoid</i>)	51
4.2.8 Proses Menghitung Turunan Fungsi Aktivasi	51
4.2.9 Proses <i>Feedforward</i>	52
4.2.9.1 Fungsi Mencari Nilai Z_{in}	52
4.2.9.2 Fungsi Mencari Nilai Y_{in}	52
4.2.9.3 Prosedur <i>Feedforward</i>	53
4.2.9 Fungsi <i>Stop Training</i>	54
4.2.10 Proses <i>Backpropagation</i>	56
4.2.10.1 Prosedur Hitung <i>Error Output</i>	56
4.2.10.2 Prosedur Mencari Nilai Delta Wjk	56
4.2.10.3 Prosedur Mencari Nilai Koreksi Wok	57
4.2.10.4 Prosedur Hitung <i>Error Hidden</i>	58
4.2.10.5 Prosedur Mencari Nilai Delta Vij	58

4.2.10.6	Prosedur Mencari Nilai Koreksi Voj	59
4.2.10.7	Prosedur Update Wjk	60
4.2.10.8	Prosedur Update Vij	60
4.2.10.9	Prosedur <i>Backpropagation</i>	61
4.3	Implementasi Antar Muka	61
4.4	Hasil Pengujian	64
4.4.1	Hasil Percobaan	64
4.4.1.1	Pengaruh Jumlah <i>Neuron</i> Pada <i>Hidden Layer</i>	64
4.4.1.2	Pengaruh Nilai <i>Learning Rate</i>	69
4.5	Analisa Hasil	73
BAB V KESIMPULAN DAN SARAN		83
5.1	Kesimpulan	83
5.2	Saran	84
DAFTAR PUSTAKA		85
LAMPIRAN		



UNIVERSITAS BRAWIJAYA



DAFTAR GAMBAR

	Halaman
Gambar 2.1 <i>Trend</i>	7
Gambar 2.2 Fungsi Aktivasi	11
Gambar 2.3 Jaringan Syaraf Tiruan <i>Feedforward</i>	12
Gambar 2.4 Jaringan Syaraf Tiruan <i>Feedback</i>	13
Gambar 3.1 Langkah-langkah Penelitian	21
Gambar 3.2 <i>Flowchart</i> Sistem	23
Gambar 3.3 <i>Flowchart</i> Normalisasi Data	25
Gambar 3.4 <i>Flowchart</i> Pelatihan JST	26
Gambar 3.5 <i>Flowchart</i> Prediksi <i>Trend</i> Saham	27
Gambar 3.6 <i>Form</i> Utama	41
Gambar 3.7 <i>Form Training</i>	43
Gambar 3.8 <i>Form Forecasting</i>	44
Gambar 4.1 Struktur data	46
Gambar 4.2 Fungsi cari nilai minimum	46
Gambar 4.3 Fungsi cari nilai maksimum	47
Gambar 4.4 Fungsi normalisasi	47
Gambar 4.5 Fungsi hitung Beta	48
Gambar 4.6 Proses inialisasi bobot	49
Gambar 4.7 Fungsi aktivasi sigmoid	50
Gambar 4.8 Turunan fungsi aktivasi	50
Gambar 4.9 Proses cari nilai Z_{in}	51
Gambar 4.10 Proses cari nilai \bar{Y}_{in}	52
Gambar 4.11 Prosedur <i>feedforward</i>	52
Gambar 4.12 Fungsi <i>StopTraining</i>	54
Gambar 4.13 Prosedur <i>ErrorOutput</i>	55
Gambar 4.14 Prosedur <i>DeltaWjk</i>	56
Gambar 4.15 Prosedur koreksi Wok	56
Gambar 4.16 Prosedur <i>ErrorHidden</i>	57
Gambar 4.17 Prosedur <i>DeltaVij</i>	58
Gambar 4.18 Prosedur koreksi Voj	58
Gambar 4.19 Prosedur <i>Update Wjk</i>	59
Gambar 4.20 Prosedur <i>Update Vij</i>	59
Gambar 4.21 Prosedur <i>Backpropagation</i>	60
Gambar 4.22 sub menu Perusahaan	61

Gambar 4.23 sub menu Saham	61
Gambar 4.24 sub menu <i>Training</i>	62
Gambar 4.25 sub menu <i>Forecasting</i>	63
Gambar 4.26 MSE pelatihan saham Astra	73
Gambar 4.27 MSE pelatihan saham Sampoerna	76
Gambar 4.28 MSE pelatihan saham Indofood	79

UNIVERSITAS BRAWIJAYA



DAFTAR TABEL

Halaman

Tabel 3.1 Input dan target kelas	28
Tabel 3.2 Definisi kelas output	28
Tabel 3.3 Faktor siklus	29
Tabel 3.4 Pengaruh jumlah <i>neuron hidden</i>	29
Tabel 3.5 Pengaruh <i>learning rate</i>	30
Tabel 3.6 Hasil pengujian	30
Tabel 3.7 Sampel data saham	30
Tabel 3.8 Hasil normalisasi	31
Tabel 3.9 Bobot dari <i>input</i> ke <i>hidden</i> (V_{ij})	31
Tabel 3.10 Inisialisasi nilai V_{ij}	32
Tabel 3.11 Nilai V_{ij} baru	32
Tabel 3.12 Nilai bias (V_{oj})	33
Tabel 3.13 Nilai W_{jk}	33
Tabel 3.14 Nilai bias (W_{ok})	33
Tabel 3.15 Tabel operasi pada <i>hidden</i> (Z_{in})	34
Tabel 3.16 Hasil aktivasi Z_{in}	35
Tabel 3.17 Tabel operasi pada <i>output</i> (Y_{in})	36
Tabel 3.18 Hasil aktivasi Y_{in}	36
Tabel 3.19 Hasil aktivasi target	36
Tabel 3.20 Nilai kemelesetan Y_k	37
Tabel 3.21 Nilai ΔW_{jk}	37
Tabel 3.22 Koreksi nilai bias <i>output</i> (W_{ok})	38
Tabel 3.23 Nilai kemelesetan Z_j	38
Tabel 3.24 Faktor kesalahan (∂_j) unit <i>hidden</i>	38
Tabel 3.25 Nilai ΔV_{ij}	39
Tabel 3.26 Nilai ΔV_{oj}	39
Tabel 3.27 Nilai V_{ij} baru	40
Tabel 3.28 Nilai V_{oj}	40
Tabel 3.29 Nilai W_{jk} baru	40
Tabel 3.30 Nilai W_{oj} baru	41
Tabel 4.1 Astra <i>Hidden neuron</i> = 10	64
Tabel 4.2 Astra <i>Hidden neuron</i> = 20	64
Tabel 4.3 Astra <i>Hidden neuron</i> = 30	64
Tabel 4.4 Astra <i>Hidden neuron</i> = 40	65

Tabel 4.5 Astra <i>Hidden neuron</i> = 50	65
Tabel 4.6 Sampoerna <i>Hidden neuron</i> = 10	65
Tabel 4.7 Sampoerna <i>Hidden neuron</i> = 20	65
Tabel 4.8 Sampoerna <i>Hidden neuron</i> = 30	66
Tabel 4.9 Sampoerna <i>Hidden neuron</i> = 40	66
Tabel 4.10 Sampoerna <i>Hidden neuron</i> = 50	66
Tabel 4.11 Indofood <i>Hidden neuron</i> = 10	66
Tabel 4.12 Indofood <i>Hidden neuron</i> = 20	67
Tabel 4.13 Indofood <i>Hidden neuron</i> = 30	67
Tabel 4.14 Indofood <i>Hidden neuron</i> = 40	67
Tabel 4.15 Indofood <i>Hidden neuron</i> = 50	67
Tabel 4.16 MSE terkecil berdasarkan jumlah <i>hidden neuron</i>	68
Tabel 4.17 Astra <i>Learning Rate</i> = 0.2	68
Tabel 4.18 Astra <i>Learning Rate</i> = 0.4	69
Tabel 4.19 Astra <i>Learning Rate</i> = 0.6	69
Tabel 4.20 Astra <i>Learning Rate</i> = 0.8	69
Tabel 4.21 Sampoerna <i>Learning Rate</i> = 0.2	69
Tabel 4.22 Sampoerna <i>Learning Rate</i> = 0.4	70
Tabel 4.23 Sampoerna <i>Learning Rate</i> = 0.6	70
Tabel 4.25 Sampoerna <i>Learning Rate</i> = 0.8	70
Tabel 4.25 Indofood <i>Learning Rate</i> = 0.2	70
Tabel 4.26 Indofood <i>Learning Rate</i> = 0.4	71
Tabel 4.27 Indofood <i>Learning Rate</i> = 0.6	71
Tabel 4.28 Indofood <i>Learning Rate</i> = 0.8	71
Tabel 4.29 MSE terkecil berdasarkan <i>learning rate</i>	72
Tabel 4.30 Hasil Peramalan Data Training Saham Astra	74
Tabel 4.31 Hasil peramalan terhadap data uji saham Astra	75
Tabel 4.32 Hasil Peramalan Data Training Saham Sampoerna	78
Tabel 4.33 Hasil peramalan terhadap data uji saham Sampoerna	79
Tabel 4.34 Hasil Peramalan Data Training Saham Indofood	81
Tabel 4.35 Hasil peramalan terhadap data uji saham Indofood	82

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Data uji saham Astra Januari-Juni 2009	87
Lampiran 2. Data uji saham Sampoerna Januari-Juni 2009	88
Lampiran 3. Data uji saham Indofood Januari-Juni 2009	89



UNIVERSITAS BRAWIJAYA



BAB I

PENDAHULUAN

1.1 Latar Belakang

Dunia investasi tidak lagi didominasi oleh jenis investasi konvensional seperti tabungan, deposito di bank atau di lembaga keuangan lain yang memberikan bunga. Para investor lebih tertarik untuk menanamkan modalnya melalui pembagian kepemilikan perusahaan yang ditandai dengan surat berharga yang disebut saham. Proses investasi menggunakan saham dilakukan dengan cara jual beli sejumlah saham yang akan menentukan persentase kepemilikan seorang investor terhadap perusahaan yang bersangkutan.

Secara umum berlaku hukum bahwa media investasi yang mempunyai tingkat kemungkinan memperoleh keuntungan yang tinggi akan diikuti dengan resiko yang tinggi pula. Resiko yang dimaksud adalah potensi terhadap kesalahan dalam membuat prediksi atas nilai atau harga di masa mendatang. Melakukan prediksi atas harga saham tidaklah mudah, dikarenakan harga saham di pasar selalu berfluktuasi. Karena itu seorang investor akan melakukan analisis data serta faktor eksternal yang mempengaruhi saham agar terhindar dari kerugian yang tinggi.

Untuk meminimalisir kesalahan dalam analisa data saham diperlukan sebuah sistem yang mampu menganalisa dan memprediksi data saham secara akurat. Data saham yang dianalisis merupakan data *time-series* yaitu data yang diukur terurut berdasarkan waktu. Jika data tersebut dimodelkan, maka dapat terlihat bahwa data tersebut memiliki *trend*. Dengan memodelkan *trend* data maka kemunculan data berikutnya dapat diketahui.

Jaringan syaraf tiruan (*artificial neural networks*) atau disingkat JST adalah sistem komputasi dimana arsitektur dan operasi diilhami dari pengetahuan tentang sel syaraf biologi di dalam otak. JST dapat digambarkan sebagai model matematis dan komputasi untuk fungsi aproksimasi non linear, klasifikasi data, cluster dan regresi non linear atau sebuah simulasi dari koleksi model syaraf biologi (Kristanto A, 2004).

Jaringan syaraf tiruan mulai diterapkan pada beberapa bidang, salah satunya adalah untuk membantu manusia dalam memprediksi

suatu kejadian atau peristiwa yang akan terjadi berdasarkan data dari masa lalu. Data masa lalu akan dipelajari oleh jaringan syaraf tiruan sehingga mempunyai kemampuan untuk memberikan keputusan terhadap data yang belum pernah dipelajari.

Terdapat beberapa model arsitektur jaringan syaraf tiruan, diantaranya adalah model *Single Layer Perceptron*, *Multi Layer Perceptron*, dan *Hopfield Network*. Dalam skripsi ini, akan digunakan model *Multi Layer Perceptron* dengan pembelajaran dengan menggunakan algoritma delta yang disebut *error backpropagation training algorithm*.

Penerapan jaringan syaraf tiruan dalam peramalan telah banyak dilakukan antara lain dalam prediksi harga saham (Bambang Budi. dkk, 1999), peramalan valuta asing (Halim S, 2000), peramalan indeks pasar modal (Wu Shang-Yu, 1999), serta penelitian lainnya. Perbedaan penelitian ini dengan penelitian sebelumnya yaitu penelitian ini lebih berfokus pada peramalan arah *trend* harga saham untuk membantu investasi saham jangka menengah.

Berdasarkan latar belakang yang telah dipaparkan, maka judul yang diambil dalam skripsi ini adalah “**Penerapan Jaringan Syaraf Tiruan *Backpropagation* dalam Prediksi *Trend* Harga Saham**”.

1.2 Perumusan Masalah

Rumusan masalah dalam skripsi ini adalah bagaimana implementasi jaringan syaraf tiruan dengan menggunakan metode *Backpropagation* dalam memprediksi *trend* harga saham.

1.3 Batasan Masalah

Batasan masalah dalam skripsi ini adalah sebagai berikut :

1. Prediksi harga saham hanya berasal dari data-data teknis dan tidak memperhatikan faktor-faktor fundamental.
2. Metodologi *learning* yang digunakan hanya satu macam yaitu *Backpropagation*.
3. Hasil yang didapatkan bukan berupa harga saham akan tetapi suatu keputusan mengenai *trend* yang akan terjadi.
4. Tidak dilakukan perbandingan dengan sistem lainnya yang sudah ada.

1.4 Tujuan Penelitian

Beberapa tujuan yang ingin dicapai dalam pelaksanaan skripsi ini adalah:

1. Pemahaman mengenai implementasi metode Jaringan Syaraf Tiruan untuk kasus berbasis *time series*, khususnya dalam masalah prediksi *trend* pergerakan harga saham.
2. Pembangunan aplikasi yang dapat memprediksi *trend* pergerakan harga saham dengan akurat.

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari pelaksanaan penelitian ini adalah:

1. Pembuatan aplikasi komputer yang mampu memberikan prediksi *trend* harga saham dengan akurat.
2. Pemahaman mengenai penerapan Jaringan Syaraf Tiruan dalam prediksi *trend* harga saham.

1.6 Metodologi Pemecahan Masalah

Untuk mencapai tujuan yang dirumuskan sebelumnya, maka metodologi yang digunakan dalam penulisan tugas akhir ini adalah:

1. Studi Literatur
Mempelajari literatur-literatur baik yang berupa buku (*textbook*), jurnal dan artikel ilmiah, maupun *website* yang berhubungan dengan Jaringan Syaraf Tiruan dan juga referensi mengenai saham
2. Pendefinisian dan analisis masalah
Mendefinisikan dan menganalisis masalah untuk mencari solusi yang optimal.
3. Perancangan dan implementasi sistem
Membuat perancangan perangkat lunak dengan analisis terstruktur dan mengimplementasikan hasil rancangan tersebut.
4. Uji coba dan analisa hasil implementasi
Menguji perangkat lunak, dan menganalisa hasil dari implementasi tersebut apakah sudah sesuai dengan tujuan yang dirumuskan sebelumnya, untuk kemudian dievaluasi dan disempurnakan.

1.7 Sistematika Penulisan

Skripsi ini disusun berdasarkan sistematika penulisan sebagai berikut:

1. BAB I PENDAHULUAN

Berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi pemecahan masalah, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Menguraikan teori-teori yang berhubungan dengan jaringan syaraf tiruan dan saham.

3. BAB III METODOLOGI DAN PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai metode-metode yang digunakan dalam prediksi *trend* harga saham.

4. BAB IV IMPLEMENTASI DAN UJI COBA SISTEM

Pada bab ini akan dilakukan implementasi sistem, pengujian dan analisa sistem perangkat lunak yang dibangun, yaitu apakah hasil dari jaringan syaraf tiruan dapat memberikan prediksi *trend* harga saham secara akurat.

5. BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan dari seluruh rangkaian penelitian serta saran kemungkinan pengembangannya.

BAB II TINJAUAN PUSTAKA

2.1 Saham

2.1.1 Pengertian Saham

Saham adalah modal yang dilepaskan oleh suatu perusahaan kepada masyarakat agar masyarakat dapat memiliki sebagian hak dari perusahaan tersebut. Hal ini dilakukan karena pemilik perusahaan membutuhkan modal untuk investasi dalam perusahaannya. Dengan menjual sahamnya, maka perusahaan harus berbagi kepemilikan perusahaan tersebut dengan pemegang saham (*stockholder*), yang berarti pula berbagi keuntungan yang diperoleh oleh perusahaan tersebut.

Saham sendiri dibagi menjadi dua jenis saham, yaitu saham biasa (*common stock*) dan saham preferen (*preferred stock*). Saham dikenal memiliki karakteristik *high risk-high return*. Artinya saham merupakan surat berharga yang memberikan peluang keuntungan yang tinggi namun juga berpotensi risiko tinggi. Saham memungkinkan pemodal mendapatkan keuntungan (*capital gain*) dalam jumlah besar dalam waktu singkat. Namun seiring dengan berfluktuasinya harga saham, saham juga dapat membuat investor mengalami kerugian besar dalam waktu singkat.

Beberapa istilah mengenai harga yang sering digunakan dalam perdagangan saham yaitu:

- a. *Open*, yaitu harga awal saham pada saat pembukaan bursa dalam suatu periode. Harga awal ini tidak selalu sama pada setiap pembukaan bursa.
- b. *High*, yaitu harga saham tertinggi pada suatu periode perdagangan.
- c. *Low*, yaitu harga saham terendah pada suatu periode perdagangan.
- d. *Close*, yaitu harga akhir saham pada saat penutupan bursa dalam suatu periode perdagangan.

(Anwar S, 2008)

2.1.2 Jenis-Jenis Saham

Saham-saham yang diperdagangkan di bursa dapat dikelompokkan ke dalam berbagai saham yaitu:

a. *Growth Stocks* adalah saham-saham yang memiliki pertumbuhan di atas pertumbuhan rata-rata. Saham ini mempunyai kecenderungan mempunyai rasio harga saham dengan pendapatan per saham cukup tinggi dibandingkan dengan pasarnya. Investor yang mempunyai informasi yang biasanya melakukan investasi pada saham ini.

b. *Cyclical Stocks* adalah saham-saham yang pertumbuhan pendapatannya sama atau di bawah pertumbuhan ekonomi secara keseluruhan. Operasi perusahaan yang sahamnya dianggap *cyclical stocks* sangat sensitif terhadap kondisi bisnis secara agregat.

c. *Defensive Stocks* adalah saham-saham yang secara umum bertumbuh pendapatannya pada rata-rata atau di bawah rata-rata tetapi tidak sangat sensitif atau sangat sedikit dipengaruhi oleh kondisi bisnis.

d. *Value Stocks* adalah saham-saham yang mempunyai harga pasar saham lebih kecil dari nilai bukunya. Artinya, rasio harga saham terhadap pendapatan sangat kecil dibandingkan dengan rasio pasar. Investor yang mempunyai informasi sangat menyukai saham ini karena kemungkinan mendapatkan kapital gain cukup besar di masa mendatang.

e. *Aggressive Stocks* adalah saham-saham yang mempunyai risiko tinggi dimana saham ini mempunyai beta di atas satu. Artinya, saham ini akan mengalami kenaikan yang lebih tinggi dari kenaikan pasar bila pasar naik dan mengalami penurunan yang tajam bila pasar turun. Saham ini biasanya disukai oleh investor yang menyukai risiko atau berspekulasi

(<http://arsasi.wordpress.com/2008/03/16/jenis-jenis-saham>, 2008)

2.1.3 Faktor yang Mempengaruhi Pergerakan Saham

Pada dasarnya harga saham dipengaruhi oleh permintaan dan penawaran, namun untuk melakukan penilaian harga saham dengan baik diperlukan data operasional perusahaan seperti laporan

keuangan yang telah diaudit, *performance* perusahaan di masa yang akan datang dan kondisi ekonomi.

Secara fundamental, faktor-faktor yang mempengaruhi pergerakan harga saham adalah faktor ekonomi, besarnya suku bunga dan politik. Harga saham cenderung terpengaruh oleh berita-berita yang memiliki dampak terhadap perekonomian.

Secara teknis, harga saham dipengaruhi oleh sentimen pasar. Harga sangat bergantung dari jumlah persediaan dan permintaan. Jika kondisi pasar sudah jenuh dalam pembelian suatu instrument, maka dapat dipastikan harga instrument tersebut akan mengalami penurunan. Harga pasar juga dapat dilihat dari arah pergerakan pasar (*market trend*) atau dari harga-harga pada hari sebelumnya. (Anwar S, 2008)

2.1.4 Trend Pasar

Trend pasar merefleksikan arah pergerakan harga saham secara umum. *Trend* merupakan hal esensial dalam dalam pendekatan teknikal. Teknik – teknik yang digunakan dalam pendekatan teknikal semuanya bertujuan untuk melakukan perdagangan sesuai dengan trend tersebut. Dalam perdagangan saham dikenal tiga jenis trend utama yaitu *uptrend* (trend kenaikan), *downtrend* (trend penurunan), dan *sideways* (kondisi biasa). (Anwar S, 2008). Ilustrasinya dapat dilihat pada gambar 2.1.



Gambar 2.1. Trend

Sumber : Ilustrasi jenis-jenis trend (Anwar S, 2008)

2.2 Time Series Forecasting

Time Series merupakan sekumpulan pengamatan yang terurut secara periodik ($x_1, x_2, x_3, \dots, x_n$), yang muncul pada *domain* yang luas. Analisis pada kasus berbasis *time series* merupakan bagian dari ilmu statistik. Sedangkan *forecasting* merupakan proses yang menghasilkan sekumpulan keluaran dengan diberikan masukan berupa sekumpulan variabel.

Time series forecasting merupakan solusi dalam bidang statistik dalam menganalisis data *spatio-temporal* (data yang memiliki dimensi spasial dan waktu). Kasus peramalan dengan menggunakan data *time series* telah banyak dikembangkan dan diaplikasikan. Salah satu penerapan *time series forecasting* yang paling umum adalah penerapan dalam masalah prediksi harga saham. Pergerakan harga saham sesuai dengan karakteristik data *time series*, mengandung data yang memiliki dimensi waktu. Prediksi pada harga saham dapat dilakukan dalam kurun waktu tertentu dengan memperhatikan data dari waktu sebelumnya. (Anwar S, 2008)

2.3 Jaringan Syaraf Tiruan

Saat ini bidang kecerdasan buatan dalam usahanya menirukan intelegensi manusia, belum mengadakan pendekatan dalam bentuk fisiknya melainkan dari sisi yang lain. Pertama-tama diadakan studi mengenai teori dasar mekanisme proses terjadinya intelegensi. Bidang ini disebut '*Cognitive Science*'. Dari teori dasar ini dibuatlah suatu model untuk disimulasikan pada komputer, dan dalam perkembangannya yang lebih lanjut dikenal berbagai sistem kecerdasan buatan yang salah satunya adalah jaringan saraf tiruan.

Dibandingkan dengan bidang ilmu yang lain, jaringan saraf tiruan relatif masih baru. Sejumlah literatur menganggap bahwa konsep jaringan saraf tiruan bermula pada makalah Waffen McCulloch dan Walter Pitts pada tahun 1943. Dalam makalah tersebut mereka mencoba untuk memformulasikan model matematis sel-sel otak. Metode yang dikembangkan berdasarkan sistem saraf biologi ini, merupakan suatu langkah maju dalam industri komputer. (Eliyani, 2005)

2.3.1 Pengertian Jaringan Syaraf Tiruan

Suatu jaringan saraf tiruan memproses sejumlah besar informasi secara paralel dan terdistribusi, hal ini terinspirasi oleh model kerja otak biologis. Beberapa definisi tentang jaringan saraf tiruan adalah sebagai berikut di bawah ini.

Menurut *Haykin, S. (1994), Neural Networks: A Comprehensive Foundation, NY, Macmillan*, mendefinisikan jaringan saraf sebagai berikut:

“Sebuah jaringan saraf adalah sebuah prosesor yang terdistribusi paralel dan mempunyai kecenderungan untuk menyimpan pengetahuan yang didapatkannya dari pengalaman dan membuatnya tetap tersedia untuk digunakan. Hal ini menyerupai kerja otak dalam dua hal yaitu: 1. Pengetahuan diperoleh oleh jaringan melalui suatu proses belajar. 2. Kekuatan hubungan antar sel saraf yang dikenal dengan bobot sinapsis digunakan untuk menyimpan pengetahuan.

Dan menurut *Zurada, J.M. (1992), Introduction To Artificial Neural Systems, Boston: PWS Publishing Company*, mendefinisikan sebagai berikut:

“Sistem saraf tiruan atau jaringan saraf tiruan adalah sistem selular fisik yang dapat memperoleh, menyimpan dan menggunakan pengetahuan yang didapatkan dari pengalaman”.

DARPA Neural Network Study (1988, AFCEA International Press, p. 60) mendefinisikan jaringan syaraf buatan sebagai berikut :

Sebuah jaringan syaraf adalah sebuah sistem yang dibentuk dari sejumlah elemen pemroses sederhana yang bekerja secara paralel dimana fungsinya ditentukan oleh stuktur jaringan, kekuatan hubungan, dan pegolahan dilakukan pada komputasi elemen atau nodes.

(Suyanto, 2008)

2.3.2 Aplikasi Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan mampu menggambarkan setiap situasi adanya sebuah hubungan antara variabel *predictor (independents, inputs)* dan variabel *predicted (dependents, outputs)*, ketika hubungan tersebut sangat kompleks dan tidak mudah untuk menjelaskan kedalam istilah yang umum dari “*correlations*” atau

“*differences between groups*”. Beberapa contoh permasalahan yang dapat dipecahkan secara baik oleh Jaringan Syaraf Tiruan antara lain:

a. Deteksi Fenomena Kedokteran.

Berbagai indikasi yang berhubungan dengan kesehatan (kombinasi dari denyut jantung, tingkatan dan berbagai substansi dalam darah, dll) dapat dimonitoring. Serangan pada kondisi kesehatan tertentu dapat dihubungkan dengan perubahan kombinasi yang sangat kompleks (nonlinear dan interaktif) pada *subset* dari variabel, dapat dimonitoring. Jaringan Syaraf Tiruan telah digunakan untuk mengenali pola yang diperkirakan sehingga perlakuan yang tepat dapat dilakukan.

b. Untuk mendeteksi golongan darah manusia

Dengan menggunakan pengolahan citra. Manusia berusaha keras dengan segala kemampuannya untuk menirukan kehebatan yang mereka miliki, misalnya seorang dokter dengan keahliannya dapat membedakan golongan darah manusia antara A, B, AB, dan O. Dengan pendekatan kecerdasan buatan, manusia berusaha menirukan bagaimana pola-pola dibentuk. Jaringan Syaraf Tiruan telah dikembangkan sebagai generalisasi model matematik dari pembelajaran manusia.

c. Prediksi Pasar Saham.

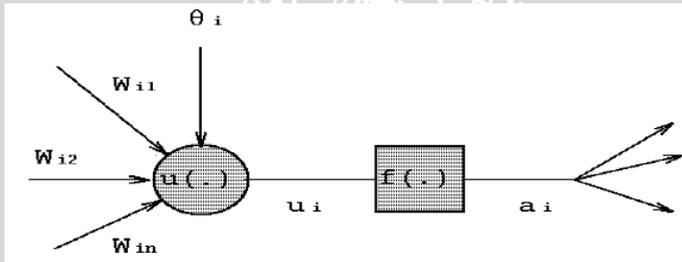
Fluktuasi dari harga saham dan index saham adalah contoh lain yang kompleks, multidimesi tetapi dalam beberapa kondisi tertentu merupakan phenomena yang dapat prediksi. Jaringan Syaraf Tiruan telah digunakan oleh analis teknik untuk membuat prediksi tentang pasar saham yang didasarkan atas sejumlah faktor seperti keadaan masa lalu bursa yang lain dan berbagai indikator ekonomi.

d. Perjanjian Kredit.

Berbagai informasi biasanya didapat dari seorang peminjam seperti umur, pendidikan, pekerjaan dan berbagai data lain. Setelah pembelajaran dari Jaringan Syaraf Tiruan tentang data peminjam, analisis Jaringan Syaraf Tiruan dapat mengidentifikasi karakteristik peminjam sehingga dapat digunakan untuk mengklasifikasikan peminjam terhadap resiko peminjam dalam kategori baik atau buruk. (Eliyani, 2005)

2.3.2 Model Dasar Jaringan Syaraf Tiruan

Mengadopsi esensi dasar dari system syaraf biologi, syaraf tiruan digambarkan sebagai berikut : Menerima input atau masukan (baik dari data yang dimasukkan atau dari output sel syaraf pada jaringan syaraf. Setiap input datang melalui suatu koneksi atau hubungan yang mempunyai sebuah bobot (*weight*). Setiap sel syaraf mempunyai sebuah nilai ambang. Jumlah bobot dari input dan dikurangi dengan nilai ambang kemudian akan mendapatkan suatu aktivasi dari sel syaraf (*post synaptic potential*, PSP, dari sel syaraf). Signal aktivasi kemudian menjadi fungsi aktivasi / fungsi transfer untuk menghasilkan output dari sel syaraf.(Eliyani, 2005). Fungsi aktivasi ($f(.)$) dapat dilihat pada Gambar 2.2.



Gambar 2.2. Fungsi Aktivasi
Sumber : Fungsi Aktifasi (Eliyani,2005)

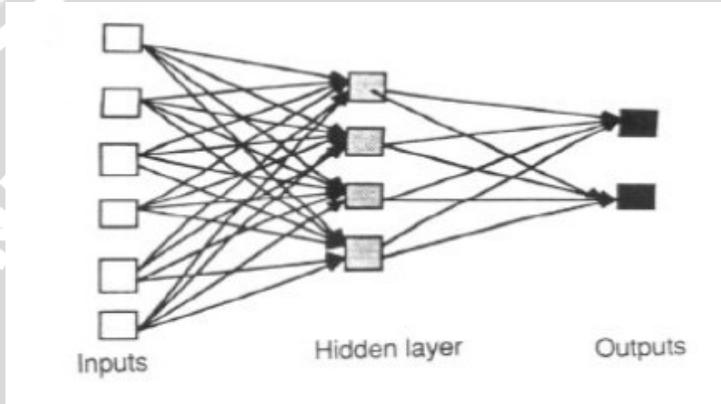
Berdasarkan dari arsitektur (pola koneksi), Jaringan Syaraf Tiruan dapat dibagi kedalam dua kategori:

a. Struktur *feedforward*

Sebuah jaringan yang sederhana mempunyai struktur *feedforward* dimana signal bergerak dari input kemudian melewati lapisan tersembunyi dan akhirnya mencapai unit output (mempunyai struktur perilaku yang stabil).

Tipe jaringan *feedforward* mempunyai sel syaraf yang tersusun dari beberapa lapisan. Lapisan input bukan merupakan sel syaraf. Lapisan ini hanya memberi pelayanan dengan mengenalkan suatu

nilai dari suatu variabel. Lapisan tersembunyi dan lapisan output sel syaraf terhubung satu sama lain dengan lapisan sebelumnya. Kemungkinan yang timbul adalah adanya hubungan dengan beberapa unit dari lapisan sebelumnya atau terhubung semuanya (lebih baik). Struktur *feedforward* dapat dilihat pada gambar 2.3.



Gambar 2.3. Jaringan Syaraf Tiruan *Feedforward*
Sumber : *Jaringan Syaraf Tiruan Feedforward* (Eliyani, 2005)

Yang termasuk dalam struktur *feedforward* :

- *Single-layer perceptron*
- *Multilayer perceptron*
- *Radial-basis function networks*
- *Higher-order networks*
- *Polynomial learning networks*

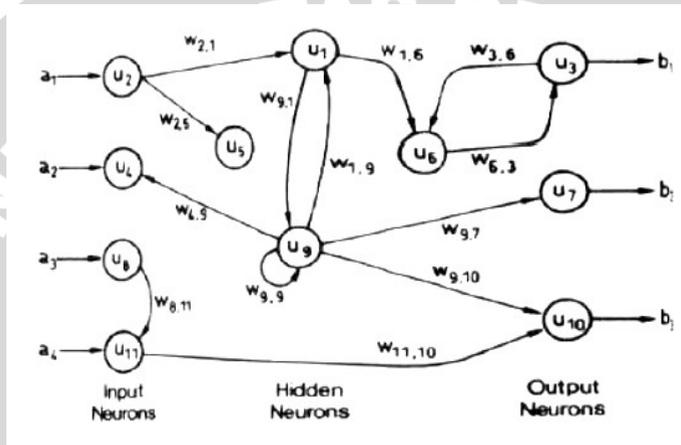
b. Struktur *recurrent (feedback)*

Jika suatu jaringan berulang (mempunyai koneksi kembali dari output ke input) akan menimbulkan ketidakstabilan dan akan menghasilkan dinamika yang sangat kompleks. Jaringan yang berulang sangat menarik untuk diteliti dalam Jaringan Syaraf Tiruan, namun sejauh ini structure *feedforward* sangat berguna untuk memecahkan masalah. Yang termasuk dalam stuktur *recurrent (feedback)* :

- *Competitive networks*
- *Self-organizing maps*
- *Hopfield networks*
- *Adaptive-resonance theory models*

(Eliyani, 2005)

Struktur *feedback* dapat dilihat pada gambar 2.4.



Gambar 2.4. Jaringan Syaraf Tiruan *FeedBack*

Sumber : *Jaringan Syaraf Tiruan FeedBack* (Eliyani, 2005)

2.3.3 Lapisan pada Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan biasanya mempunyai 3 group atau lapisan yaitu unit-unit : lapisan **input** yang terhubung dengan lapisan **hidden** yang selanjutnya terhubung dengan lapisan **output**. Aktifitas unit-unit lapisan input menunjukkan informasi dasar yang kemudian digunakan dalam Jaringan Syaraf Tiruan. Aktifitas setiap unit-unit lapisan tersembunyi ditentukan oleh aktifitas dari unit-unit input dan bobot dari koneksi antara unit-unit input dan unit-unit lapisan tersembunyi. Karakteristik dari unit-unit output tergantung dari aktifitas unit-unit lapisan tersembunyi dan bobot antara unit-unit lapisan tersembunyi dan unit-unit output.

(Eliyani, 2005)

2.3.4 Proses Pembelajaran

Umumnya, jika menggunakan Jaringan Syaraf Tiruan, hubungan antara input dan output harus diketahui secara pasti dan jika hubungan tersebut telah diketahui maka dapat dibuat suatu model. Hal lain yang penting adalah proses belajar hubungan input/output dilakukan dengan pembelajaran. Ada dua tipe pembelajaran yang dikenal yaitu : pembelajaran terawasi dan pembelajaran tak terawasi.

Pada pembelajaran terawasi, metode ini digunakan jika output yang diharapkan telah diketahui sebelumnya. Biasanya pembelajaran dilakukan dengan menggunakan data yang telah ada. Pada contoh diatas misalnya data pasar saham yang ada pada DOW, NASDAQ atau FTSE, data yang ada sebelumnya mengenai aplikasi kredit yang berhasil termasuk daftar pertanyaan atau posisi sebuah robot dan reaksi yang benar.

Pada metode pembelajaran yang tidak terawasi, tidak memerlukan target output. Pada metode ini tidak dapat ditentukan hasil seperti apa yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu range tertentu tergantung pada nilai input yang diberikan. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran seperti ini biasanya sangat cocok untuk pengelompokkan (klasifikasi) pola. (Eliyani, 2005)

2.3.5 Fungsi Transfer

Karakter dari Jaringan Syaraf Tiruan tergantung atas bobot dan fungsi input output (fungsi transfer) yang mempunyai ciri tertentu untuk setiap unit. Fungsi ini terdiri dari 3 kategori yaitu :

- Untuk *linear units*, Aktifitas output adalah sebanding dengan jumlah bobot output.

$$f(x) = x, \text{ untuk semua } x \quad (2.1)$$

- Untuk *threshold units*, Output diatur satu dari dua tingkatan tergantung dari apakah jumlah input adalah lebih besar atau lebih kecil dari nilai ambang.

$$f(x) = \begin{cases} 1 & \text{jika } x \geq \theta \\ 0 & \text{jika } x < \theta \end{cases} \quad (2.2)$$

- Untuk *sigmoid units*, Output terus menerus berubah-ubah tetapi tidak berbentuk linear.

Fungsi biner sigmoid

$$f(x) = 1/1+e^{-x} \quad (2.3)$$

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

Unit ini mengandung kesamaan yang lebih besar dari sel syaraf sebenarnya dibandingkan dengan linear dan *threshold* unit, namun ketiganya harus dipertimbangkan dengan perkiraan kasar.

2.3.6 Jaringan Syaraf *Backpropagation*

Jaringan syaraf *backpropagation* merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Algoritma *backpropagation* menggunakan error output untuk mengubah nilai bobot-bobotnya dalam arah mundur. Untuk mendapatkan error, tahap perambatan maju harus dikerjakan terlebih dahulu. Pelatihan sebuah jaringan yang menggunakan *backpropagation* terdiri dari 3 langkah, yaitu : pelatihan pola input secara *feedforward*, perhitungan dan *backpropagation* dari kumpulan kesalahan dan penyesuaian bobot. Sesudah pelatihan, aplikasi dalam jaringan hanya terdiri dari fase *feedforward*. Bahkan, jika pelatihan menjadi lambat, sebuah jaringan yang dilatih dapat menghasilkan outputnya sendiri secara cepat. Banyak variasi dari *backpropagation* yang dapat dibangun untuk meningkatkan kecepatan proses pelatihan.

2.3.6.1 Algoritma Jaringan Syaraf *Backpropagation*

1. Inisialisasi bobot awal dengan nilai acak yang cukup kecil
2. Bila test kondisi berhenti bernilai salah (belum mencapai epoch maksimum atau belum mencapai batas nilai keluaran yang diharapkan), maka:

Feedforward:

- a. Tiap masukan sinyal x_i diteruskan ke lapisan di atasnya (lapisan tersembunyi)
- b. Tiap unit tersembunyi y_i menjumlahkan sinyal – sinyal masukan berbobot :

$$y_in_j = v_{oj} + \sum_{i=1}^n x_i v_{ij}$$

gunakan fungsi aktivasi untuk hitung sinyal keluaran:

$$y_j = f(y_in_j)$$

kemudian salurkan sinyal keluaran ke lapisan di atasnya.

- c. Tiap unit keluaran (z_k) menjumlahkan sinyal-sinyal masukan berbobot:

$$z_in_k = w_{ok} + \sum_{i=1}^n y_i w_{ik}$$

gunakan fungsi aktivasi untuk hitung sinyal keluaran:

$$z_k = f(z_in_k)$$

kemudian salurkan sinyal keluaran ke lapisan di atasnya.

Backpropagation:

- a. Tiap unit keluaran (z_k) menerima target pola yang berkaitan dengan pola masukan pembelajaran, hitung nilai kemelesetan:

$$\delta_k = (t_k - z_k) f'(z_in_k)$$

Pakai koreksi bobot untuk memperbaharui nilai w_{jk} :

$$\Delta w_{jk} = c \partial_k y_j$$

Pakai koreksi bias untuk memperbaharui nilai w_{0k} :

$$W_{0k} = c \partial_k$$

Kirimkan ∂_k ke lapisan dibawahnya.

- b. Tiap unit tersembunyi (y_i) menjumlahkan delta masukan dari unit-unit yang berada diatasnya.

$$\partial_{in_j} = \sum_{k=1}^m \partial_k w_{jk}$$

nilai ini dikalikan dengan turunan dari fungsi aktivasi untuk mendapatkan nilai kemelesetan:

$$\partial = \partial_{in_j} f'(y_{in_j})$$

Pakai nilai koreksi kemelesetan untuk memperbaharui nilai v_{ij} :

$$\Delta v_{ij} = c \partial_j x_i$$

Pakai juga nilai koreksi bias untuk memperbaharui nilai v_{0j} :

$$\Delta v_{0j} = c \partial_j$$

- c. Tiap unit keluaran (z_k) memperbaharui bias dan bobotnya :

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

Tiap unit tersembunyi (y_j) memperbaharui bias dan bobotnya :

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

3. Apabila $E < E_{\max}$ atau max epoch telah terpacai maka lakukan tes kondisi berhenti.
(Muis S, 2006)

2.3.6.2 Karakteristik Jaringan Syaraf Backpropagation

Beberapa hal yang perlu diperhatikan untuk pemanfaatan algoritma pelatihan *Error Backpropagation*, antara lain :

1. Bobot (w atau v)

Nilai bobot yang terlalu kecil akan menghasilkan nilai keluaran jaringan syaraf baik ke neuron tersembunyi maupun ke neuron

keluaran mendekati nol, ini menyebabkan waktu panjang untuk proses pelatihan/pembelajaran.

Nilai inialisasi bobot awal pada umumnya antara -0.5 sampai 0.5 disamping itu inialisasi bobot secara dinamis selama proses pelatihan akan sangat menentukan kecepatan belajar neuron tersembunyi.

n = banyaknya *neuron input*

p = banyaknya *neuron hidden*

β = faktor skala

$\beta = 0.7(p)^{1/n}$

v_{ij} (lama) = antara -0.5 sampai 0.5 (inialisasi bobot awal)

proses inialisasi bobot secara dinamis untuk neuron j ($j=1,2,..,p$) pada lapisan tersembunyi sebagai berikut:

- Hitung $\| v_{ij}$ (lama) $\|$

- Hitung bobot v_{ij} (baru)

$$v_{ij} = \beta v_{ij} \text{ (lama)} / \| v_{ij} \text{ (lama)} \|\|$$

- Nilai bias (v_{0j}) ditentukan antara $-\beta$ dan β

2.Lama proses pelatihan

Pelatihan jaringan syaraf *Error Backpropagation* ditujukan untuk memperoleh hasil keluaran (untuk tiap pola masukan) sama dengan pola asli yang dijadikan sebagai acuan pelatihan.

3.Banyaknya pola masukan

Ada hubungan antara banyakan pola masukan P yang tersedia untuk pelatihan dengan banyaknya bobot W yang hendak dilatih dan ketelitian e yang diharapkan.

$$P = W/e$$

Misalnya tingkat ketelitian diharapkan $e = 0.1$ (kesalahan 10%) dan banyaknya bobot yang hendak dilatih 80, maka banyaknya data pola masukan yang diperlukan adalah 800.

4.Presentasi data

Pada umumnya jaringan syaraf tiruan dapat dilatih dengan data masukan berupa kontinu atau diskrit. Jaringan syaraf tiruan lebih mudah dilatih dengan data diskrit daripada data kontinu.

5.Neuron lapisan tersembunyi.

Banyaknya neuron lapisan tersembunyi berkaitan dengan “jumlah” lapisan neuron tersembunyi. Pada umumnya jaringan

syaraf dengan tiga lapisan neuron yaitu lapisan masukan, lapisan tersembunyi, dan lapisan keluaran sudah mampu memberikan tingkat ketelitian yang cukup memadai.
(Muis S, 2006)

2.4 Normalisasi

Bila jaringan syaraf tiruan dilatih dengan keluaran diskrit (diwakili 0/1) berupa pengelompokan m keluaran, variabel masukan harus dinormalisasi terlebih dahulu. (Muis, 2006). Data- data yang ada dilakukan normalisasi dengan membagi nilai data tersebut dengan nilai range data (nilai data maksimum-nilai data minimum). (Bambang Budi. dkk, 1999). Fungsi aktivasi yang akan digunakan dalam sistem ini adalah fungsi sigmoid biner karena fungsi ini memiliki sifat non-linier sehingga sangat baik untuk menyelesaikan permasalahan dunia nyata yang kompleks dan biasanya bersifat non-linier. Data ditransformasi pada interval [0.1, 0.9] karena fungsi ini merupakan fungsi asimtotik yang nilainya tidak pernah mencapai 0 atau 1. Rumus untuk mendapatkan nilai normalisasi data dapat dilihat pada persamaan 2.1. Dimana a adalah batas bawah (nilai minimum dari data) dan b adalah batas atas (nilai maksimum dari data)

$$x' = \frac{0.8(x - a)}{b - a} + 0.1 \quad (2.5)$$

UNIVERSITAS BRAWIJAYA



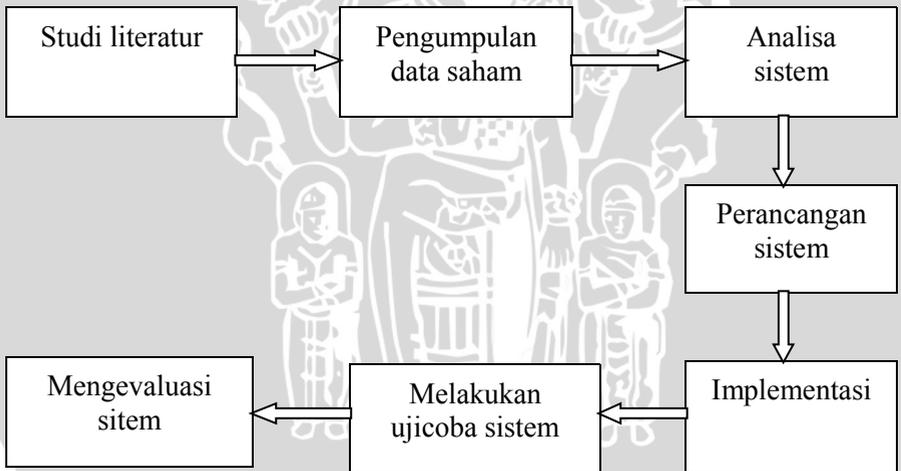
BAB III
METODOLOGI DAN PERANCANGAN SISTEM

Dalam bab ini akan dijelaskan mengenai analisis sistem dan perancangan sistem untuk prediksi *trend* saham menggunakan jaringan syaraf tiruan dengan metode *backpropagation*.

Penelitian dilakukan dengan tahapan-tahapan sebagai berikut:

1. Mempelajari literatur yang terkait dengan masalah saham dan jaringan syaraf tiruan *backpropagation*.
2. Mengumpulkan data – data historis saham dari situs *Yahoo! Finance*.
3. Menganalisa sistem dan melakukan perancangan sistem menggunakan jaringan syaraf tiruan *backpropagation*.
4. Mengimplementasikan sistem.
5. Melakukan uji coba sistem dengan memasukan data – data historis saham ke dalam sistem.
6. Mengevaluasi hasil analisa sistem.

Langkah-langkah yang dilakukan dapat dijelaskan kembali dalam bentuk alur yang ditunjukkan dalam Gambar 3.1.



Gambar 3.1 Langkah-langkah penelitian

3.1 Studi Literatur

Dalam merealisasikan tujuan dan pemecahan masalah, penelitian ini dilakukan dengan studi literatur. Teori-teori mengenai konsep

jaringan syaraf tiruan, dan prediksi pergerakan *trend* saham yang digunakan sebagai dasar penelitian diperoleh dari buku dan internet. Kemudian data-data yang telah diperoleh diubah ke bentuk yang dapat diproses oleh program sehingga dapat digunakan untuk melakukan analisa.

3.2 Data Yang Digunakan

Data yang digunakan untuk penelitian ini adalah data – data historis saham

1. Astra International Tbk
2. H M Sampoerna Tbk
3. Indofood Sukses Makmur Tbk

tahun 2008 yang diambil dari situs *Yahoo!*. Data yang digunakan adalah data harga penutupan karena paling mencerminkan pergerakan saham. Untuk proses training, hanya data yang lengkap yang akan dijadikan sebagai data training. Sebagai data uji akan digunakan data saham dari bulan Januari sampai Juni tahun 2009.

3.3 Deskripsi Umum Sistem

Secara umum sistem yang dibangun adalah perangkat lunak untuk memberikan rekomendasi yang mengimplementasikan jaringan syaraf tiruan *backpropagation*. Jaringan syaraf tiruan berfungsi sebagai pengidentifikasi kategori *output*. Sistem bertujuan memberikan prediksi mengenai *trend* harga saham. Prediksi yang di *output*-kan oleh sistem yaitu:

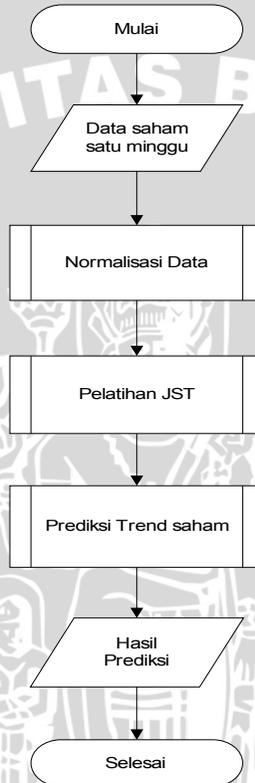
1. *Trend* naik (*Bullish*)
2. *Trend* turun (*Bearish*)
3. *Trend* naik berubah ke *trend* turun
4. *Trend* turun berubah ke *trend* naik

Dari informasi yang dihasilkan tersebut diharapkan dapat memberikan rekomendasi kepada investor dalam membeli atau menjual saham.

3.4 Perancangan Proses

Sistem ini terdiri atas beberapa tahapan yaitu pengambilan data historis saham yang kemudian akan dinormalisasi terlebih dahulu.

Setelah dinormalisasi, data siap untuk dilatihkan dalam jaringan syaraf tiruan. Apabila error minimum telah didapatkan maka sistem siap untuk melakukan testing atau melakukan prediksi yang akan menghasilkan *output* berupa *trend* saham. Tahapan yang dilakukan oleh sistem secara umum dapat dilihat pada gambar 3.2.



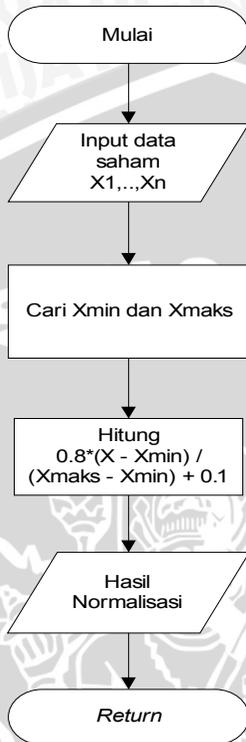
Gambar 3.2 Flowchart Sistem

3.4.1 Proses Normalisasi Data

Karena jaringan syaraf tiruan dilatih dengan keluaran diskrit (diwakili 0/1) berupa pengelompokan m keluaran, maka variabel masukan harus dinormalisasi terlebih dahulu. Data harga saham ditransformasikan dengan menggunakan persamaan 2.10. Normalisasi dilakukan dengan cara mencari batas atas yaitu nilai maksimum dari data dan batas bawah yaitu nilai minimum dari data. Kemudian nilai x dikurangi nilai minimum dibagi dengan nilai range data. Agar data berada pada interval (0.1-0.9) maka data dikalikan 0.8 dan ditambahkan dengan 0.1.

Flowchart proses normalisasi dapat dilihat pada gambar 3.3 dengan langkah-langkah sebagai berikut:

1. Mulai
2. Input data saham dari X_1 sampai dengan X_n
3. Kemudian cari batas atas data yaitu data maksimum dan batas bawah data yaitu data minimum.
4. Hitung nilai X dikurangi batas bawah kemudian dibagi dengan range data (X maksimum dikurangi X minimum), hasilnya dikalikan dengan 0.8 kemudian ditambahkan dengan 0.1 agar data berada pada interval 0.1 - 0.9
5. Didapatkan *output* berupa hasil normalisasi
6. Return



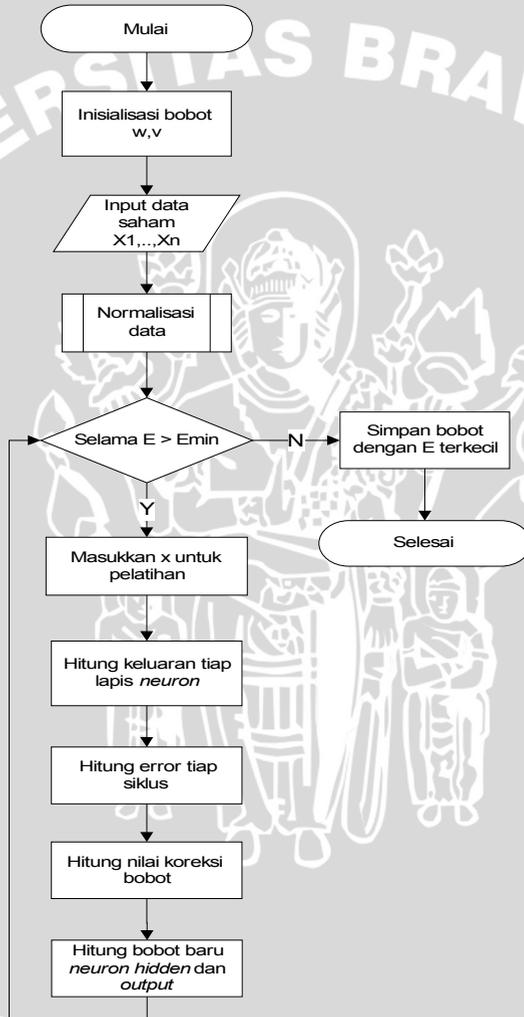
Gambar 3.3 *Flowchart* Normalisasi Data

3.4.2 Pelatihan Jaringan Syaraf Tiruan

Flowchart pelatihan jaringan syaraf tiruan dapat dilihat pada gambar 3.4 dengan langkah-langkah sebagai berikut :

1. Mulai
2. Inisialisasi bobot awal untuk lapisan *hidden* v dan lapisan *output* w dengan nilai random dari -0.5 sampai 0.5
3. Masukkan data saham selama 1 minggu ditambah faktor siklus
4. Lakukan proses normalisasi data
5. Selama $error > error$ minimum yang ditetapkan, lakukan langkah 6-10
6. Masukkan data saham x sebagai data pelatihan

7. Hitung *output* pada lapisan *hidden* dan lapisan *output*
8. Hitung *error* pada lapisan *hidden* dan lapisan *output*
9. Hitung nilai koreksi bobot pada *hidden* dan *output*
10. Hitung nilai bobot baru pada *hidden* dan *output*
11. Simpan nilai bobot baru dengan E terkecil.
12. Pelatihan selesai.

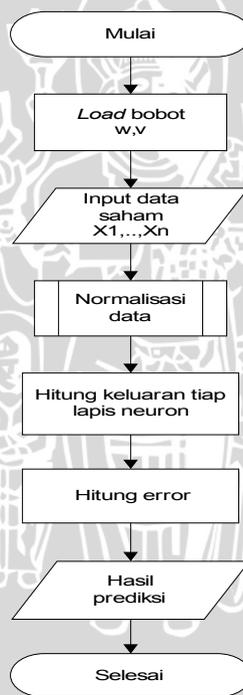


Gambar 3.4 Flowchart Pelatihan JST

3.4.3 Proses Prediksi *Trend* Saham

Flowchart prediksi *trend* saham dapat dilihat pada gambar 3.5 dengan langkah-langkah sebagai berikut :

1. Mulai
2. *Load* bobot w dan v yang telah didapatkan dari proses pelatihan
3. Masukkan data saham
4. Lakukan proses normalisasi data.
5. Hitung *output* pada lapisan *hidden* dan lapisan *output*.
6. Hitung *error* pada lapisan *hidden* dan lapisan *output*.
7. Hasil prediksi dan error didapatkan sebagai *output*
8. Selesai.



Gambar 3.5 Flowchart Prediksi *Trend* Saham

3.5 Menentukan *Input* dan *Output* Data

Input data yang digunakan dalam sistem adalah data harga penutupan saham selama n hari sebelumnya. Dari data tersebut diberikan pasangan target output kategorial yang merupakan bilangan biner. Setelah dilakukan iterasi diharapkan nilai output yang diberikan oleh sistem akan sama atau mendekati dari target output yang diberikan. Tabel input dan target kelas dapat dilihat pada tabel 3.1.

Tabel 3.1 Input dan target kelas

X_1	X_2	X_3	X_4	X_5	X_f	Kejadian(x_{t+1})	Output			
X_1	X_2	X_3	X_4	X_5	X_f	<i>Trend</i> naik	0	0	0	0
X_1	X_2	X_3	X_4	X_5	X_f	<i>Trend</i> naik berubah turun	0	1	1	0
X_1	X_2	X_3	X_4	X_5	X_f	<i>Trend</i> turun	1	1	1	1
X_1	X_2	X_3	X_4	X_5	X_f	<i>Trend</i> turun berubah naik	1	0	0	1

Data output akan didefinisikan agar lebih jelas dalam pengklasifikasiannya. Dari ciri-ciri yang didapatkan didapatkan definisi sebagai berikut. Definisi dari kelas output dapat dilihat pada tabel 3.2.

Kejadian(x_{t+1})	Output				Definisi
<i>Trend</i> naik	0	0	0	0	$X_1 < X_{rata-rata} < X_5$ dan $X_1 < X_2$
<i>Trend</i> naik berubah turun	0	1	1	0	$X_1 > X_{rata-rata} > X_5$ dan $X_1 > X_2$
<i>Trend</i> turun	1	1	1	1	$X_1 > X_{rata-rata} > X_5$ dan $X_1 > X_2$
<i>Trend</i> turun berubah naik	1	0	0	1	$X_1 < X_{rata-rata} < X_5$ dan $X_1 < X_2$

3.7 Pertimbangan Faktor Siklus

Faktor siklus dijadikan sebagai input tambahan kedalam sistem. Diharapkan dengan menambahkan faktor siklus hasil prediksi yang didapatkan menjadi lebih akurat. Faktor siklus didapatkan dari *trend* harga saham setahun sebelumnya. Faktor siklus dikelompokkan berdasarkan *trend* yang terjadi tahun sebelumnya, apabila tidak terdapat data faktor siklus pada tahun sebelumnya maka faktor siklus dianggap 0. Tabel faktor siklus akan disajikan pada tabel 3.3.

Tabel 3.3 Faktor siklus

Faktor siklus	Trend yang terjadi
0.8	<i>Trend naik</i>
0.6	<i>Trend naik berubah turun</i>
0.4	<i>Trend turun berubah naik</i>
0.2	<i>Trend turun</i>

3.8 Perancangan Uji Coba

Pada subbab perancangan uji coba, akan dijelaskan mengenai pengujian datanya. Data yang digunakan dalam penelitian ini adalah harga penutupan saham, seperti yang telah dijelaskan pada subbab 3.2.

- Pengaruh Jumlah *Neuron Hidden*

Untuk mengetahui pengaruh *neuron hidden* terhadap banyaknya iterasi, dilakukan pengujian dengan menentukan jumlah *neuron* yang diujikan. Jumlah iterasi maksimal ditetapkan 20000 iterasi. Bentuk dari hasil percobaan tersebut akan disajikan pada tabel 3.4.

Tabel 3.4 Pengaruh jumlah *neuron hidden* terhadap banyaknya iterasi.

Jumlah <i>hidden</i>	MSE	Waktu

- Pengaruh *Learning Rate*

Tabel pengaruh *Learning rate* pada kesalahan minimal dapat dilihat pada tabel 3.5.

Tabel 3.5. Pengaruh *learning rate* pada kesalahan minimal

<i>Learning rate</i>	MSE	Waktu

Dari jaringan yang terbentuk nantinya akan dilakukan proses forecasting atau peramalan. Tabel hasil peramalan jaringan syaraf tiruan dapat dilihat pada tabel 3.6

Tabel 3.6 Hasil Pembelajaran.

Data	Hasil Prediksi	Real

3.6 Contoh Perhitungan

Dari data saham Astra didapatkan data sampel yang dapat dilihat pada tabel 3.7.

Tahap 1 Input Data

Tabel 3.7 Sampel data saham

Senin	Selasa	Rabu	Kamis	Jumat	F	T ₁	T ₂	T ₃	T ₄
10500	10550	10500	11000	11500	0.8	0	0	0	0

Data tersebut harus ditransformasi terlebih dahulu pada interval [0.1, 0.9] dengan menggunakan persamaan 3.1, karena fungsi aktivasi yang digunakan adalah fungsi sigmoid, dimana fungsi ini merupakan fungsi asimtotik yang nilainya tidak pernah mencapai 0 atau 1.

$$x' = \frac{0.8(x - a)}{b - a} + 0.1$$

Misalnya diambil kejadian *trend* naik sebagai sampel. Data hasil normalisasi dapat dilihat pada tabel 3.6.

Tahap 2 Normalisasi Data

Tabel 3.8 Hasil normalisasi

Senin	Selasa	Rabu	Kamis	Jumat	F	T ₁	T ₂	T ₃	T ₄
0,1	0,14	0,14	0,5	0,9	0.8	0	0	0	0

Inisialisasi bobot dari *input* ke *hidden* dapat dilihat pada tabel 3.9.

Tahap 3 Inisialisasi Bobot

Tabel 3.9 Bobot dari *input* ke *hidden* (v_{ij})

V_{ij}	Z_1	Z_2	Z_3	Z_4	Z_5
X_1	-0.2808	-0.1597	0.2304	0.2204	0.0537
X_2	-0.0811	-0.3790	-0.0672	-0.0881	0.2182
X_3	-0.4293	0.3543	-0.4072	-0.4889	0.0270
X_4	0.3802	0.1489	0.3325	-0.0544	0.3355
X_5	-0.0837	0.0875	-0.4343	0.0902	-0.1316
X_6	0.0513	0.2596	0.4606	0.4792	0.4062

Proses inisialisasi V_{ij} (baru):

$$n \text{ (jumlah unit input)} = 6$$

$$p \text{ (jumlah unit hidden)} = 5$$

$$\beta \text{ (faktor skala)} = 0.7 * (5)^{1/6} = 0.915$$

Nilai V_j :

$$V_1 = \sqrt{[(-0.2808)^2 + (-0.0811)^2 + (-0.4293)^2 + (0.3802)^2 + (-0.0837)^2 + (0.0513)^2]} = 0.6508$$

⋮

$$V_5 = \sqrt{[(0.0537)^2 + (0.2182)^2 + (0.0270)^2 + (0.3355)^2 + (-0.1316)^2 + (0.4062)^2]} = 0.5880$$

Nilai V_j dapat dilihat pada tabel 3.10

Tabel 3.10 Inisialiasi nilai V_j

J	V_j
1	0.6508
2	0.6257
3	0.8570
4	0.7319
5	0.5880

$$V_{11} \text{ (baru)} = \frac{\beta * V_{11} \text{ (lama)}}{\|V_1\|} = \frac{0.915 * -0.2808}{\|0.6508\|} = -0.3947$$

⋮

$$V_{65} \text{ (baru)} = \frac{\beta * V_{65} \text{ (lama)}}{\|V_5\|} = \frac{0.915 * 0.4062}{\|0.5880\|} = 0.5056$$

Nilai V_{ij} baru dapat dilihat pada tabel 3.11

Tabel 3.11 Nilai V_{ij} baru

V_{ij}	Z_1	Z_2	Z_3	Z_4	Z_5
X_1	-0.3947	-0.2329	0.2460	0.2857	0.0836
X_2	-0.1049	-0.5348	-0.0714	-0.1106	0.3376
X_3	-0.5528	0.4322	-0.4324	-0.5612	0.0383
X_4	0.4396	0.1725	0.3482	-0.0618	0.4754
X_5	-0.0932	0.1008	-0.4516	0.1023	-0.1653
X_6	0.0571	0.2984	0.4743	0.5429	0.5056

Nilai bias V_{0j} adalah bilangan acak yang bernilai antara -0.915 dan 0.915

Beta = 0.915

Random(-0.915-0.915)

Nilai bias V_{0j} dapat dilihat

Dimensi V_{0j} :

Baris = jumlah neuron hidden

Kolom = satu

Tabel 3.12 Nilai bias (V_{0j})

J	V_{0j}
1	-0.2800
2	-0.4132
3	0.3461
4	-0.5914
5	0.1700

Nilai W_{jk} diperoleh dari proses acak dengan range antara -0,05 sampai 0,05 :

Inisialisasi nilai W_{jk} dapat dilihat pada tabel 3.13

Tabel 3.13 Nilai W_{jk}

W_{jk}	W_1	W_2	W_3	W_4
Z_1	0.2800	-0.4235	-0.0125	-0.1410
Z_2	0.4110	-0.4294	-0.2334	-0.2590
Z_3	-0.3044	-0.2438	0.1847	-0.3577
Z_4	0.3793	0.3714	-0.4522	-0.0696
Z_5	-0.0780	-0.0945	-0.0233	0.1388

Nilai W_{0k} diperoleh dari proses acak dengan range antara -0,05 sampai 0,05 :

Inisialisasi nilai bias W_{0k} dapat dilihat pada tabel 3.14

Tabel 3.14 Nilai bias W_{0k}

K	W_{0k}
1	0.1692
2	0.2495
3	-0.2369
4	-0.1680

Tahap 4 Proses *Feedforward*

Hitung keluaran unit *hidden* (z_j)

Nilai z_{in_j} :

$$z_{in_j} = v_{oj} + \sum_{i=1}^n x_i v_{ij}$$

$$\begin{aligned} Z_{in_1} &= -0.28 + \{ (0.1 * -0.3947) + (0.14 * -0.1049) + \\ & (0.14 * -0.5528) + (0.5 * 0.4396) + (0.9 * -0.0932) + (0.8 * 0.0571) \} \\ &= -0.2079 \end{aligned}$$



$$\begin{aligned} Z_{in_5} &= 0.17 + \{ (0.1 * 0.0836) + (0.0.4754) + (0.14 * 0.0383) + \\ & (0.5 * 0.4754) + (0.9 * -0.1653) + (0.8 * 0.5056) \} = 0.7228 \end{aligned}$$

Hasil operasi pada *hidden neuron* dapat dilihat pada tabel 3.15.

Tabel 3.15 Tabel operasi pada *hidden* (Z_{in})

J	Z in
1	-0.0279
2	-0.0524
3	0.4646
4	-0.1389
5	0.7228

Kemudian dihitung aktivasinya dengan menggunakan persamaan

$$f(x) = 1/1+e^{-z}$$

Hasil aktivasi unit *hidden* dapat dilihat pada tabel 3.16.

Tabel 3.16 Hasil aktivasi Z_{in}

J	Z
1	0.4482
2	0.4868
3	0.6141
4	0.4653
5	0.6732

Hitung keluaran unit *output* (y_k)

Nilai y_{in_k}

$$y_{in_k} = w_{ok} + \sum_{i=1}^n y_i w_{jk}$$

$$y_{in_1} = 0.1692 + \{ (0.4482 * 0.2800) + (0.4868 * 0.4110) + (0.6141 * -0.3044) + (0.4653 * 0.3793) + (0.6732 * -0.0780) \} = 0.4319$$

Hasil operasi pada *output neuron* dapat dilihat pada tabel 3.17.

Tabel 3.17 Tabel operasi pada *output* (Y_{in})

K	Y_{in}
1	0.4319
2	-0.1899
3	-0.3834
4	-0.5160

Kemudian dihitung aktivasinya dengan menggunakan persamaan

$$f(x) = 1/1+e^{-y}$$

Hasil aktivasinya dapat dilihat pada tabel 3.18

Tabel 3.18 Hasil aktivasi Y_{in}

J	Y
1	0.6063
2	0.4526
3	0.4053
4	0.3737

Tahap 5 Proses *Backpropagation*:

Hitung aktivasi target dengan fungsi sigmoid

$$f(x) = 1/1+e^{-x}$$

Hasil aktivasi target dapat dilihat pada tabel 3.19.

Tabel 3.19 Hasil aktivasi target

K	T
1	0.5
2	0.5
3	0.5
4	0.5

Hitung nilai kemelesetan y_k :

$$\partial_k = (t_k - y_k) f'(y_{in_k}) = \partial_{in_k} y_k (1 - y_k)$$

$$\partial_1 = (0.5 - 0.6063) * 0.6063 * (1 - 0.6063) = -0.0254$$

Nilai kemelesetan y_k dapat dilihat pada tabel 3.20

Tabel 3.20 Nilai kemelesetan y_k

K	∂
1	-0.0254
2	0.0117
3	0.0228
4	0.0295

Hitung nilai koreksi bobotnya dengan $\alpha = 0.1$

$$\Delta W_{jk} = \alpha \partial_k z_j$$

$$\Delta W_{11} = 0.1 * -0.0254 * 0.4482 = -0.00114$$

.

.

$$\Delta W_{54} = 0.1 * 0.0295 * 0.6732 = 0.00199$$

Nilai koreksi bobot pada unit *output* dapat dilihat pada tabel 3.21.

Tabel 3.21 Tabel ΔW_{jk}

ΔW_{jk}	ΔW_1	ΔW_2	ΔW_3	ΔW_4
Z_1	-0.00114	0.00053	0.00102	0.00132
Z_2	-0.00124	0.00057	0.00111	0.00144
Z_3	-0.00156	0.00072	0.00140	0.00181
Z_4	-0.00118	0.00055	0.00106	0.00137
Z_5	-0.00171	0.00079	0.00154	0.00199

Koreksi bias bobot ΔW_{0k}

$$\Delta W_{01} = \alpha * \delta_1 = 0.1 * -0.0254 = -0.00254$$

Nilai koreksi bias pada unit *output* dapat dilihat pada tabel 3.18

Tabel 3.22 Koreksi nilai bias *output* (W_{0k})

K	W_0
1	-0.00254
2	0.00117
3	0.00228
4	0.00295

Hitung nilai kemelesetan z_j :

$$\partial_{in_j} = \sum_{k=1}^m \partial_k w_{jk}$$

Tabel nilai kemelesetan y_j dapat dilihat pada tabel 3.23.

Tabel 3.23 Tabel ∂_{in_j}

J	∂_{in_j}
1	-0.01653
2	-0.01779
3	-0.00149
4	-0.01765
5	-0.00037

Hitung faktor kesalahan ∂ di unit *hidden*

$$\partial_j = \partial_{in_j} f'(z_{in_j}) = \partial_{in_j} z_j (1 - z_j)$$

Faktor kesalahan di unit *hidden* dapat dilihat pada tabel 3.24.

Tabel 3.24 Tabel ∂_j

J	∂
1	-0.00409
2	-0.00444
3	-0.00035
4	-0.00439
5	0.00003

Pakai nilai koreksi kemelesetan untuk memperbaharui nilai v_{ij} :

$$\Delta v_{ij} = \alpha \delta_j x_i$$

Nilai koreksi bobot pada unit *hidden* dapat dilihat pada tabel 3.25.

Tabel 3.25 Tabel ΔV_{ij}

ΔV_{ij}	ΔV_1	ΔV_2	ΔV_3	ΔV_4	ΔV_5
X_1	-0.00004	-0.00004	0	-0.00004	0
X_2	-0.00006	-0.00006	0	-0.00006	0
X_3	-0.00004	-0.00004	0	-0.00004	0
X_4	-0.0002	-0.00022	-0.00002	-0.00022	0
X_5	-0.00037	-0.0004	-0.00003	-0.0004	0.00001
X_6	-0.00033	0.00036	-0.00008	-0.00035	0.00001

Koreksi bias bobot ΔV_{0j}

$$\Delta v_{0j} = \alpha \delta_j$$

Nilai koreksi bobot bias dapat dilihat pada tabel 3.26

Tabel 3.26 Tabel ΔV_{0j}

J	ΔV_{0j}
1	-0.000409
2	-0.000444
3	-0.000035
4	-0.000439
5	0.000003

Perbaharui bobot di unit *hidden*

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

Bobot baru di unit *hidden* dapat dilihat pada tabel 3.27.

Tabel 3.27 Tabel V_{ij} baru

V_{ij}	Z_1	Z_2	Z_3	Z_4	Z_5
X_1	-0.39477	-0.23293	0.2460	0.28566	0.0836
X_2	-0.10497	-0.53492	-0.0714	-0.11066	0.3376

X_3	-0.55291	0.43219	-0.4324	-0.56124	0.0383
X_4	0.43942	0.17231	0.34822	-0.06208	0.4754
X_5	-0.0936	0.10042	-0.45172	0.10194	-0.16539
X_6	0.05677	0.29806	0.47434	0.54263	0.50558

Kemudian perbaiki nilai bias V_{0j}

Nilai bias V_{0j} baru dapat dilihat pada tabel 3.28

Tabel 3.28 Tabel V_{0j} baru

J	V_{0j}
1	-0.28046
2	-0.41371
3	0.34612
4	-0.59192
5	0.17

Perbaharui bobot di unit *output*

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

Bobot baru di unit *output* dapat dilihat pada tabel 3.29.

Tabel 3.29 Tabel W_{jk} baru

W_{jk}	W_1	W_2	W_3	W_4
Z_1	0.2789	-0.423	-0.0115	-0.1397
Z_2	0.4098	-0.4289	-0.2346	-0.2576
Z_3	-0.306	-0.2431	0.1861	-0.356
Z_4	0.3782	0.372	-0.4512	-0.0683
Z_5	-0.0797	-0.0937	-0.02324	0.1409

Kemudian perbaiki nilai bias W_{0j}

Nilai bias W_{0j} baru dapat dilihat pada tabel 3.30

Tabel 3.30 Tabel W_{0j} baru

K	W_{0k}
1	0.16666

2	0.25067
3	-0.2346
4	-0.1651

Setelah dilakukan perbaikan bobot sampai didapatkan error minimal atau max epoch, maka hasil perubahan bobot akan disimpan untuk digunakan pada saat ada data baru yang akan dilakukan peramalan.

3.7 Perancangan Antar Muka

Untuk antar muka dari perangkat lunak prediksi *trend* saham ini terdiri dari 3 form, yaitu form saham, form *training*, dan form *forecast*. Form utama dapat dilihat pada gambar 3.1, form *training* dapat dilihat pada gambar 3.2, form *forecast* dapat dilihat pada gambar 3.3.



Form Saham

→ A

→ B

Gambar 3.1 Form Utama

Pada Form utama terdiri dari 2 bagian yaitu:

A : Menu navigasi yang berisi *option* yang dapat dipilih oleh user. Pada menu terdapat pilihan *training*, *forecast*, dan *exit*.

B : Halaman untuk menampilkan data saham

Form *Training*

→ C
→ A
→ B

Gambar 3.2 Form *Training*

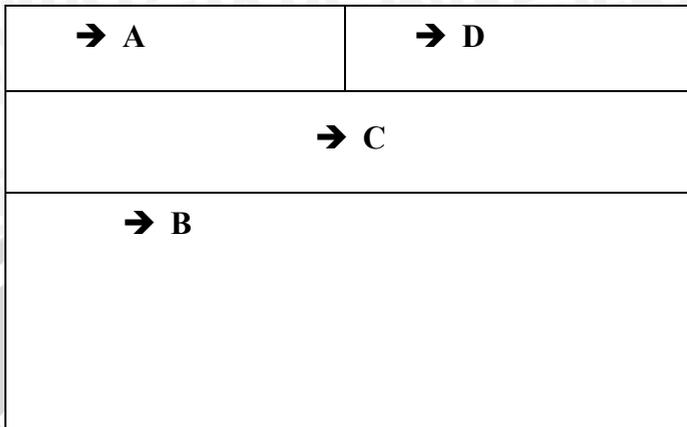
Pada Form *training* terdiri dari 3 bagian yaitu:

A : Menu yang berisi pilihan untuk mengatur arsitektur jaringan syaraf tiruan dan mengatur nilai parameter yang akan digunakan dalam proses *training*.

B : Menampilkan proses *training*

C : Menu navigasi

Form *Forecast*



Gambar 3.3 Form *Forecast*

Pada Form *forecast* terdiri dari 4 bagian yaitu:

- A : Menu data input dan parameter yang digunakan
- B : Menampilkan proses peramalan
- C : Menu navigasi
- D : Hasil *forecast*

UNIVERSITAS BRAWIJAYA



BAB IV IMPLEMENTASI DAN PEMBAHASAN

4.1 Lingkungan Implementasi

Lingkungan implementasi meliputi lingkungan perangkat keras serta lingkungan perangkat lunak.

4.1.1 Lingkungan perangkat keras

Perangkat keras yang digunakan dalam pembangunan sistem peramalan *trend* harga saham ini adalah:

1. Prosesor Intel(R) Centrino Duo(R) 2.20 GHz.
2. RAM 4 GB
3. *Harddisk* dengan kapasitas 320 GB
4. Monitor
5. *Keyboard*
6. *Mouse*

4.1.2 Lingkungan perangkat lunak

Perangkat lunak yang digunakan dalam pembangunan sistem peramalan *trend* harga saham ini adalah :

1. Sistem Operasi *Microsoft Windows Vista*
2. *Borland Delphi 7*
3. *Microsof Access 2007*

4.2 Implementasi Program

Pada sub bab ini akan dibahas mengenai implementasi dari sistem peramalan *trend* harga saham.

4.2.1. Struktur Data

Struktur data yang digunakan untuk menyimpan data harga saham, bobot dan output sistem adalah *array*. Struktur data tersebut seperti pada gambar 4.1

```
const
    SUM_OUTPUT = 4;
    SUM_INPUT = 6;
type
    ar_saham= Array [0..SUM_INPUT - 1] of Real;
    ar_target = Array [0..SUM_OUTPUT - 1] of Real;
    ar_dua_dimensi = Array of Array of Real;
    ar_satu_dimensi = Array of Real;
```

Gambar 4.1 Struktur data

Nilai SUM_OUTPUT ditetapkan nilainya 4 dikarenakan jumlah output yang akan dihasilkan oleh sistem berjumlah 4. Sedangkan nilai SUM_INPUT ditetapkan nilainya 6 karena jumlah input yang akan diinputkan ke dalam sistem berjumlah 6. Dari struktur data tersebut dapat dijelaskan bahwa penyimpanan setiap data harga saham merupakan *array* dari tipe data *real*. Penyimpanan data target juga merupakan *array* dari tipe data *real*. Untuk menyimpan data bobot bias dan *error* digunakan data *array* 1 dimensi dan untuk bobot *neuron hidden* dan *output* digunakan *array* 2 dimensi.

4.2.2 Proses Mencari Nilai Minimum

Fungsi untuk mencari nilai minimum dapat dilihat pada gambar 4.2.

```
function getMinimum(ar_data: ar_saham) : Real;
var
  r_min : Real;
  i : Integer;
begin
  r_min := ar_data[0];
  for i := 1 to Length(ar_data) - 2 do
  begin
    if (ar_data[i] < r_min) then
      r_min := ar_data[i];
    end;
  Result := r_min;
end;
```

Gambar 4.2 Fungsi cari nilai minimum

Fungsi tersebut digunakan untuk mencari nilai minimum harga saham yang nantinya akan digunakan untuk proses normalisasi.

4.2.3 Proses Mencari Nilai Maksimum

Fungsi untuk mencari nilai maksimum dapat dilihat pada gambar 4.3.



```

function getMaximum(ar_data: ar_saham) : Real;
var
  r_max : Real;
  i : Integer;
begin
  r_max := ar_data[0];

  for i := 1 to Length(ar_data) - 2 do
  begin
    if (ar_data[i] > r_max) then
      r_max := ar_data[i];
    end;
  end;
  Result := r_max;
end;

```

saham yang nantinya akan digunakan untuk proses normalisasi.

4.2.4 Proses Normalisasi

Fungsi untuk melakukan proses normalisasi dapat dilihat pada gambar 4.4

```

function Normalisasi(r_data : Real; r_min :
Real; r_max : Real) : Real;
begin
  Result := RoundTo(((0.8*(r_data - r_min)) /
(r_max - r_min)) + 0.1, -2);
end;

```

Gambar 4.4 Fungsi normalisasi

Fungsi tersebut digunakan untuk melakukan normalisasi terhadap data harga saham agar *range* data yang didapatkan sebagai hasil berkisar dari 0.1 - 0.9.

4.2.5 Proses Menghitung Faktor Skala (β)

Fungsi untuk menghitung nilai faktor skala (β) dapat dilihat pada gambar 4.5.

```

function Beta(i_neuron_input : Integer;
i_neuron_hidden : Integer) : Real;
begin
  Result := RoundTo((0.7 *
Power(i_neuron_hidden,1/i_neuron_input)),
-5);
end;

```

Gambar 4.5 Fungsi hitung β

```
procedure InisialisasiBobot ();
var
    i , j : Integer;
    r_beta : Real;
begin
    r_beta:=Beta(i_neuron_input,i_neuron_hidden);
    Randomize;
    for i :=0 to i_neuron_input - 1 do
    begin
        for j := 0 to i_neuron_hidden - 1 do
        begin
            ar_bobot_Vij[i,j]:=RoundTo(RandomValue(0.5),-
            5);
            end;
        end;
        for i := 0 to i_neuron_hidden - 1 do
        begin
            ar_bias_Voj[i]:=RoundTo(RandomValue(r_beta),-
            5);
            end;
            for i := 0 to i_neuron_input - 1 do
            begin
                for j := 0 to i_neuron_hidden - 1 do
                begin
                    ar_bobot_Vij[i,j] := RoundTo(((r_beta *
                    ar_bobot_Vij[i,j]) / getVj(j)), -5);
                    end;
                end;
                for i := 0 to i_neuron_hidden - 1 do
                begin
                    for j := 0 to SUM_OUTPUT - 1 do
                    begin
                        ar_bobot_Wjk[i,j]:=RoundTo(RandomValue(0.5),-
                        5);
                        end;
                    end;
                    for i := 0 to SUM_OUTPUT - 1 do
                    begin
                        ar_bias_Wok[i]:= RoundTo(RandomValue(0.5),-5);
                        end;
                    end;
                end;
```

UNIVERSITAS BRAWIJAYA



Gambar 4.6 Proses inialisasi bobot

Fungsi diatas digunakan untuk inialisasi bobot. Tahap pertama yang dilakukan adalah mengambil nilai faktor skala (β) kemudian melakukan inialisasi bobot bias *neuron hidden* dan bobot pada *neuron hidden* secara acak. Setelah itu nilai bias dan bobot *hidden* akan dikalikan dengan faktor skala. Fungsi tersebut juga digunakan untuk inialisasi bobot bias *neuron output* dan bobot pada *neuron output*.

4.2.7 Proses Menghitung Nilai Aktivasi (*Sigmoid*)

Fungsi untuk menghitung nilai aktivasi dapat dilihat pada gambar 4.7.

```
function Sigmoid(r_data : Real) : Real;  
begin  
Result := RoundTo(1/(1 + exp(-r_data)), -5);  
end;
```

Gambar 4.7 Fungsi aktivasi sigmoid

Fungsi tersebut digunakan untuk menghitung nilai aktivasi dari *neuron hidden* dan *neuron output*.

4.2.8 Proses Menghitung Turunan Fungsi Aktivasi

Fungsi untuk menghitung turunan fungsi aktivasi *sigmoid* (*Sigmoid Derivative*) dapat dilihat pada gambar 4.8.

```
function SigmoidDerivative(r_data : Real) : Real;  
begin  
Result := RoundTo((Sigmoid(r_data)) *  
(1 - Sigmoid(r_data)), -5);  
end;
```

Gambar 4.8 Turunan fungsi aktivasi

Fungsi *SigmoidDerivative* digunakan untuk melakukan proses perhitungan nilai *error* pada *neuron hidden* dan *neuron output*.

4.2.9 Proses *Feedforward*

Prosedur untuk melakukan proses *feedforward* terdiri dari prosedur *Z_in*, *Y_in*, dan *Feedforward*.

4.2.9.1 Fungsi Mencari Nilai *Z_in*

Fungsi mencari nilai Z_{in} dapat dilihat pada gambar 4.9.

```
function getZ_in(i_index : Integer) : Real;
var
    i : Integer;
    r_Z_in : Real;
begin
    r_Z_in := 0;
    for i := 0 to i_neuron_input - 1 do
    begin
        r_Z_in := r_Z_in + (ar_data_saham[i] *
        ar_bobot_Vij[i,i_index]);
    end;
    Result := RoundTo((ar_bias_Voj[i_index]+
    r_Z_in),-5);
end;
```

Gambar 4.9 Proses cari nilai Z_{in}

4.2.9.2 Fungsi Mencari Nilai Y_{in}

Fungsi untuk mencari nilai Y_{in} dapat dilihat pada gambar 4.10.

```
function getY_in(i_index : Integer) : Real;
var
    i : Integer;
    r_Y_in : Real;
begin
    r_Y_in := 0;
    for i := 0 to i_neuron_hidden - 1 do
    begin
        r_Y_in := r_Y_in + (ar_Z[i] *
        ar_bobot_Wjk[i,i_index]);
    end;
    Result := RoundTo((ar_bias_Wok[i_index] +
    r_Y_in),-5);
end;
```

Gambar 4.10 Proses cari nilai Y_{in}

4.2.9.3 Prosedur *Feedforward*

Prosedur untuk melakukan proses *feedforward* dapat dilihat pada gambar 4.11.

```
procedure FeedForward(i_iterasi : Integer);
var
    i : Integer;
begin
    for i := 0 to i_neuron_hidden - 1 do
    begin
        ar_Zin[i] := getZ_in(i);
        ar_Z[i] := Sigmoid(ar_Zin[i]);
    end;

    for i := 0 to SUM_OUTPUT - 1 do
    begin
        ar_Y_in[i] := RoundTo(getY_in(i), -5);
        ar_Y[i] := RoundTo(Sigmoid(ar_Y_in[i]), -5);
        ar_Y_hit[i_iterasi-1, i] := ar_Y_in[i];
        ar_T_hit[i_iterasi-1, i] := ar_target_saham[i];
    end;
end;
```

Gambar 4.11 Prosedur *feedforward*

Prosedur *feedforward* digunakan untuk mencari nilai *output* dari *neuron hidden* dan *neuron output*. Nilai tersebut kemudian akan diaktivasi dengan menggunakan fungsi *sigmoid*.

4.2.9 Fungsi StopTraining

Fungsi ini digunakan untuk menentukan apakah proses training dilanjutkan atau dihentikan. Dalam fungsi ini dihitung nilai MSE

(*Mean Square Error*) dari proses training. Apabila MSE yang didapat dari proses training lebih dari error yang diinginkan maka training dilanjutkan, akan tetapi apabila MSE yang didapat kurang dari error yang diinginkan maka iterasi training dihentikan dan bobot-bobot terakhir yang didapat akan disimpan. Fungsi ini dapat dilihat pada gambar 4.12.

```
function StopTraining(r_mse : Real): Boolean;
var
  r_error : Real;
  r_tot_error : Real;

  if r_min_error > r_tot_error then
  begin
    for i := 0 to i_neuron_input - 1 do
    begin
      for j := 0 to i_neuron_hidden - 1 do
      begin
        ar_bobot_Vij_min[i,j] := ar_bobot_Vij[i,j];
      end;
    end;

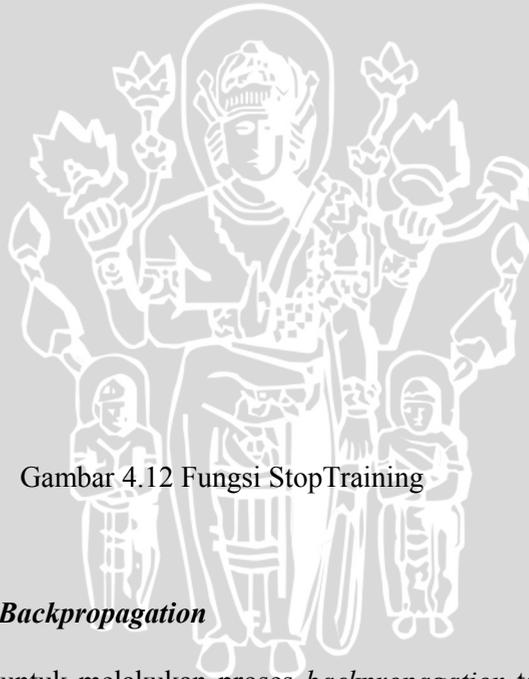
    for i := 0 to i_neuron_hidden - 1 do
    begin
      ar_bias_Voj_min[i] := ar_bias_Voj[i];
    end;

    for i := 0 to i_neuron_hidden - 1 do
    begin
      for j := 0 to SUM_OUTPUT - 1 do
      begin
        ar_bobot_Wjk_min[i,j] := ar_bobot_Wjk[i,j];
      end;
    end;

    for i := 0 to SUM_OUTPUT - 1 do
    begin
      ar_bias_Wok_min[i] := ar_bias_Wok[i];
    end;

    r_min_error := r_tot_error;
  end;
  if r_tot_error < r_mse then
  Result := True
  else
  Result := False;
  end;
```

UNIVERSITAS BRAWIJAYA



Gambar 4.12 Fungsi StopTraining

4.2.10 Proses *Backpropagation*

Prosedur untuk melakukan proses *backpropagation* terdiri dari prosedur *ErrorOutput*, *DeltaWjk*, *KoreksiWok*, *ErrorHidden*, *DeltaVij*, *KoreksiVoj*, *UpdateVij*, *UpdateWjk*.

4.2.10.1 Prosedur Hitung *Error Output*

Prosedur *ErrorOutput* dapat dilihat pada gambar 4.13.

```
procedure ErrorOutput();
var
    i : Integer;
begin
    for i := 0 to SUM_OUTPUT - 1 do
    begin
        ar_t[i] := Sigmoid(ar_target_saham[i]);
        ar_error_output[i] := RoundTo((ar_t[i] - ar_Y[i])
        *SigmoidDerivative(ar_Y_in[i]), -5);
    end;
end;
```

Gambar 4.13 Prosedur *ErrorOutput*

Prosedur *ErrorOutput* digunakan untuk menghitung nilai kemelesetan target dan keluaran dari *neuron output* yang akan dikalikan dengan turunan fungsi aktivasi untuk mendapatkan nilai dari *ErrorOutput*.

4.2.10.2 Prosedur Mencari Nilai Delta Wjk.

Prosedur *DeltaWjk* dapat dilihat pada gambar 4.14.

```
procedure DeltaWjk();
var
    i, j : Integer;
begin
    for i := 0 to i_neuron_hidden - 1 do
    begin
        for j := 0 to SUM_OUTPUT - 1 do
        begin
            ar_delta_bobot_Wjk[i,j] :=
            RoundTo(ar_error_output[j] * r_learning_rate *
            ar_Z[i], -5);
        end;
    end;
end;
```

Gambar 4.14 Prosedur *DeltaWjk*

Prosedur *DeltaWjk* adalah prosedur untuk mencari nilai koreksi bobot dari *neuron hidden* ke *neuron output*.

4.2.10.3 Prosedur Mencari Nilai Koreksi Wok

Prosedur *KoreksiWok* dapat dilihat dari gambar 4.15

```
procedure KoreksiWok();
var
    i : Integer;
begin
    for i := 0 to SUM_OUTPUT - 1 do
    begin
        ar_bias_Wok[i] := RoundTo(ar_bias_Wok[i] +
            (r_learning_rate * ar_error_output[i]), -5);
    end;
end;
```

Gambar 4.15 Prosedur *KoreksiWok*

```
procedure ErrorHidden();
var
    i, j : Integer;
    temp : Real;
begin
    for i := 0 to i_neuron_hidden - 1 do
    begin
        temp := 0;
        for j := 0 to SUM_OUTPUT - 1 do
        begin
            temp := temp + ar_error_output[j] *
                ar_bobot_Wjk[i,j];
        end;
        ar_error_hidden[i] := RoundTo(temp, -5);
        ar_faktor_koreksi[i] := RoundTo(ar_error_hidden[i]
            * SigmoidDerivative(ar_Zin[i]), -5);
    end;
end;
```

UNIVERSITAS BRAWIJAYA

Gambar 4.16 Prosedur *ErrorHidden*

Prosedur *ErrorHidden* digunakan untuk menghitung nilai kemelesetan dari *neuron hidden* yang akan dikalikan dengan turunan fungsi aktivasi untuk mendapatkan nilai dari *ErrorHidden*.

4.2.10.5 Prosedur Mencari Nilai Delta Vij

Prosedur *DeltaVij* dapat dilihat pada gambar 4.17.

```
procedure DeltaVij();
var
  i, j : Integer;
begin
  for i := 0 to i_neuron_input - 1 do
  begin
    for j := 0 to i_neuron_hidden - 1 do
    begin
      ar_delta_bobot_Vij[i, j] :=
        RoundTo((r_learning_rate * ar_faktor_koreksi[j]
          * ar_data_saham[i]), -5);
    end;
  end;
end;
```

Gambar 4.17 Prosedur *DeltaVij*

Prosedur *DeltaVij* adalah prosedur untuk mencari nilai koreksi bobot dari *neuron input* ke *neuron hidden*.

4.2.10.6 Prosedur Mencari Nilai Koreksi Voj

Prosedur *KoreksiVoj* dapat dilihat dari gambar 4.18

```
procedure KoreksiVoj();
var
    i : Integer;
begin
    for i := 0 to i_neuron_hidden - 1 do
    begin
        ar_bias_Voj[i] := RoundTo(ar_bias_Voj[i] +
            (r_learning_rate * ar_faktor_koreksi[i]), -5);
    end;
end;
```

Gambar 4.18 Prosedur *KoreksiVoj*

Prosedur *KoreksiVoj* digunakan untuk mencari nilai koreksi bobot bias pada *neuron hidden* kemudian langsung menjumlahkannya untuk mendapatkan bobot baru V_{0j} .

4.2.10.7 Prosedur Melakukan *Update Wjk*

```
procedure UpdateWjk();
var
    i, j : Integer;
begin
    for i := 0 to i_neuron_hidden - 1 do
    begin
        for j := 0 to SUM_OUTPUT - 1 do
        begin
            ar_bobot_Wjk[i,j] := RoundTo((ar_bobot_Wjk[i,j]
                + ar_delta_bobot_Wjk[i,j]), -4);
        end;
    end;
end;
```

Gambar 4.19 Prosedur *UpdateWjk*

Prosedur *UpdateWjk* digunakan untuk memperbaharui nilai bobot pada *neuron output* dengan menggunakan nilai *delta Wjk*.

4.2.10.8 Prosedur *UpdateVij*

Prosedur *UpdateVij* dapat dilihat dari gambar 4.20

```
procedure UpdateVij();
var
    i, j : Integer;
begin
    for i := 0 to i_neuron_input - 1 do
    begin
        for j := 0 to i_neuron_hidden - 1 do
        begin
            ar_bobot_Vij[i,j] :=
            RoundTo((ar_bobot_Vij[i,j] +
            ar_delta_bobot_Vij[i,j]),-5);
        end;
    end;
end;
```

Gambar 4.20 Prosedur *UpdateVij*

Prosedur *UpdateVij* digunakan untuk memperbaharui nilai bobot pada *neuron hidden* dengan menggunakan nilai *delta Vij*.

```
procedure BackPropagation();
begin
    ErrorOutput();
    DeltaWjk();
    KoreksiWok();
    ErrorHidden();
    DeltaVij();
    KoreksiVoj();
    UpdateVij();
    UpdateWjk();
end;
```

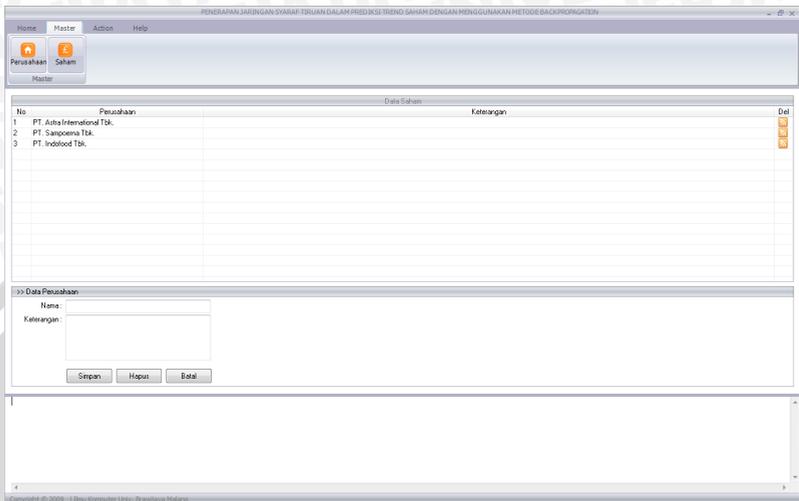
Gambar 4.21 Prosedur *backpropagation*

4.3 Implementasi Antarmuka

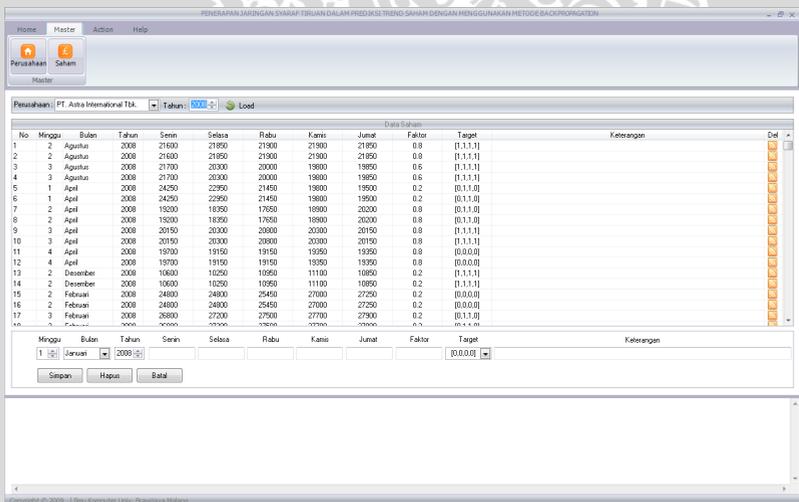
Berdasarkan rancangan antarmuka yang telah dijelaskan pada BAB III dihasilkan antarmuka yaitu:

1. *Form Master*

Form Master yaitu *form* yang digunakan untuk memasukkan data perusahaan dan data saham kedalam sistem. *Form Master* memiliki 2 sub menu yaitu sub menu perusahaan untuk memasukkan, mengubah, maupun menghapus data perusahaan. Sedangkan sub menu saham digunakan untuk memasukkan, mengubah, dan menghapus data saham. Sub menu Perusahaan dan sub menu Saham dapat dilihat pada gambar 4.21 dan 4.22



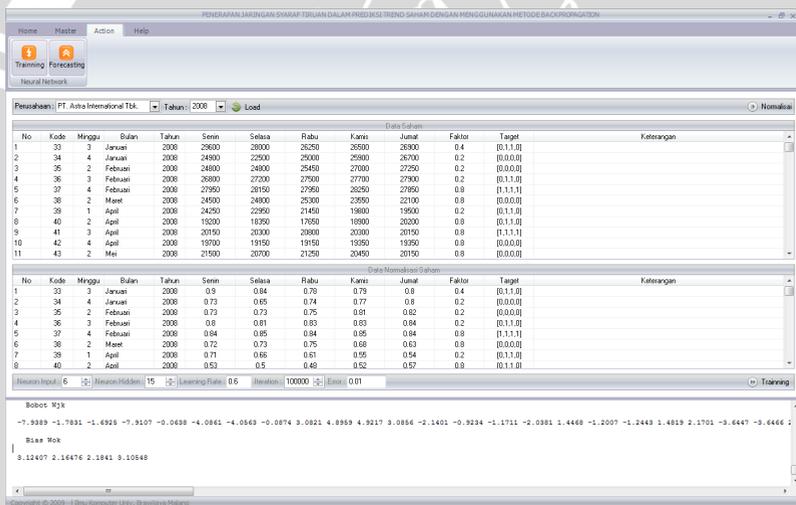
Gambar 4.22 sub menu Perusahaan



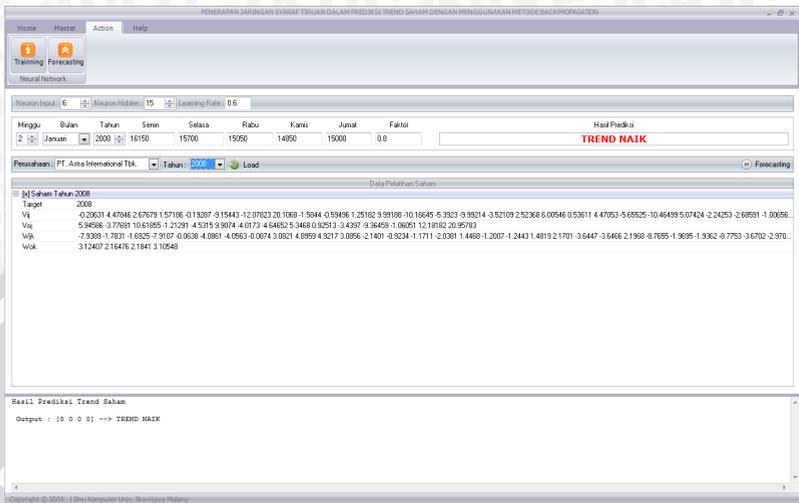
Gambar 4.23 sub menu Saham

2. Form Action

Form Action memiliki 2 sub menu yaitu sub menu *Training* dan sub menu *Forecasting*. Pada sub menu *Training*, user dapat melakukan proses *training* pada sistem dengan menentukan parameter-parameter yaitu *neuron hidden*, *learning rate*, iterasi, dan *error* yang diinginkan. Sebelum melakukan proses *training*, data-data saham yang ada harus dinormalisasi terlebih dahulu. Pada sub menu *Forecasting*, parameter-parameter yang telah ditentukan di *form Training* akan di *load* pada *form Forecasting* ini. Kemudian *user* dapat mengisikan data saham yang akan diprediksi hasilnya. Sub menu *Training* dan sub menu *Forecasting* dapat dilihat pada gambar 4.24 dan 4.25.



Gambar 4.24 sub menu *Training*



Gambar 4.25 sub menu *Forecasting*

4.4 Hasil Pengujian

Untuk memperoleh struktur jaringan syaraf tiruan yang terbaik yang digunakan untuk prediksi *trend* saham maka dilakukan pengujian terhadap sistem. Pengujian dilakukan dengan melatih jaringan syaraf tiruan dengan parameter-parameter yang berbeda, yang nantinya akan diambil satu struktur jaringan yang terbaik yang digunakan untuk melakukan prediksi *trend* saham.

4.4.1 Hasil Percobaan

Percobaan dilakukan guna menentukan jumlah *hidden neuron*, *learning rate* dan target kesalahan agar didapatkan struktur jaringan syaraf tiruan yang terbaik.

4.4.1.1 Pengaruh Jumlah *Neuron* pada *Hidden Neuron*

Untuk menentukan jumlah *hidden neuron* pada sistem dilakukan melalui proses *trial and error*. Dalam percobaan ini akan dilakukan percobaan dengan learning rate 0.5. Kemudian jumlah neuron yang diujikan antara 10 hingga 50 dengan selang 10 neuron. Jumlah iterasi

maksimal ditetapkan 20000 iterasi. Masing-masing *hidden neuron* dilakukan percobaan sebanyak 5 kali.

a. Hasil pelatihan pada saham Astra International Tbk

Tabel 4.1 *Hidden Neuron = 10*

Percobaan	MSE	Waktu
1	0.04936	68.8634s
2	0.05627	62.9780s
3	0.03635	67.5463s
4	0.08202	70.6331s
5	0.09907	74.7859s
rata-rata	0.06461	68.9613s

Tabel 4.2 *Hidden Neuron = 20*

Percobaan	MSE	Waktu
1	0.05149	104.2350s
2	0.05619	100.5880s
3	0.06341	107.6296s
4	0.06224	99.8299s
5	0.08921	99.0891s
rata-rata	0.06451	102.2743s

Tabel 4.3 *Hidden Neuron = 30*

Percobaan	MSE	Waktu
1	0.03680	126.1910s
2	0.05590	124.7095s
3	0.03760	123.4097s
4	0.03786	126.5336s
5	0.03707	120.2141s
rata-rata	0.04104	124.2116s

Tabel 4.4 *Hidden Neuron = 40*

Percobaan	MSE	Waktu
1	0.05016	150.3669s
2	0.05406	150.9803s
3	0.03626	157.8414s
4	0.08817	159.4850s
5	0.03704	147.8021s
rata-rata	0.05313	153.2951s

Tabel 4.5 *Hidden Neuron = 50*

Percobaan	MSE	Waktu
1	0.03650	145.9248s
2	0.03230	150.0347s
3	0.06500	148.4838s
4	0.05716	147.3056s
5	0.05906	147.9248s
rata-rata	0.05000	147.9347s

b. Hasil pelatihan pada saham H M Sampoerna Tbk

Tabel 4.6 *Hidden Neuron = 10*

Percobaan	MSE	Waktu
1	0.08823	60.6713s
2	0.08092	66.6204s
3	0.06941	66.3472s
4	0.08665	66.8484s
5	0.09692	66.5764s
rata-rata	0.08442	65.4127s

Tabel 4.7 *Hidden Neuron = 20*

Percobaan	MSE	Waktu
1	0.08623	89.2419s
2	0.08664	88.8854s
3	0.08071	89.4907s
4	0.06788	88.4873s
5	0.06832	88.5706s
rata-rata	0.07795	88.9351s

Tabel 4.8 *Hidden Neuron = 30*

Percobaan	MSE	Waktu
1	0.08792	110.9595s
2	0.05587	111.6019s
3	0.07681	110.6782s
4	0.08428	110.8669s
5	0.07562	109.7697s
rata-rata	0.07610	110.7752s

Tabel 4.9 *Hidden Neuron = 40*

Percobaan	MSE	Waktu
1	0.09367	135.6053s
2	0.07690	131.6516s
3	0.08336	131.6088s
4	0.06606	132.3137s
5	0.08830	132.1644s
rata-rata	0.08165	132.6688s

Tabel 4.10 *Hidden Neuron = 50*

Percobaan	MSE	Waktu
1	0.06109	154.2083s
2	0.06911	139.2269s
3	0.08258	154.6840s
4	0.07211	153.1435s
5	0.06336	152.4931s
rata-rata	0.06965	150.7512s

c. Hasil pelatihan pada saham Indofood Sukses Makmur Tbk

Tabel 4.11 *Hidden Neuron = 10*

Percobaan	MSE	Waktu
1	0.11895	66.0324s
2	0.12495	64.0972s
3	0.11022	64.5660s
4	0.13668	66.8553s
5	0.12137	66.9329s
rata-rata	0.12243	65.6967s

Tabel 4.12 *Hidden Neuron* = 20

Percobaan	MSE	Waktu
1	0.10279	85.1678s
2	0.11048	88.9653s
3	0.07223	89.5706s
4	0.11246	89.1273s
5	0.11502	89.1308s
rata-rata	0.10259	88.3923s

Tabel 4.13 *Hidden Neuron* = 30

Percobaan	MSE	Waktu
1	0.09744	111.5093s
2	0.09472	111.9155s
3	0.12127	112.1400s
4	0.10526	111.6262s
5	0.10150	111.7847s
rata-rata	0.10403	111.7951s

Tabel 4.14 *Hidden Neuron* = 40

Percobaan	MSE	Waktu
1	0.09823	134.0231s
2	0.11066	133.4734s
3	0.11413	134.1516s
4	0.09450	132.6875s
5	0.10160	132.7905s
rata-rata	0.10382	133.4252s

Tabel 4.15 *Hidden Neuron* = 50

Percobaan	MSE	Waktu
1	0.09938	152.7222s
2	0.11674	153.0532s
3	0.09783	151.2095s
4	0.06328	145.1991s
5	0.09737	152.2824s
rata-rata	0.09492	150.8933s

Berdasarkan hasil pelatihan dan hasil pengujian pada tabel 4.1 sampai dengan tabel 4.15 dapat dilihat bahwa MSE terkecil yang dihasilkan pada masing-masing unit untuk tiap pengujian selalu berubah-ubah. Berikut ini diberikan tabel MSE terkecil dari tiap unit ditunjukkan pada Tabel 4.16

Tabel 4.16 MSE terkecil

Perusahaan	MSE	Waktu	<i>neuron hidden</i>
Astra International Tbk	0.04104	124.2116s	30
H M Sampoerna Tbk	0.06965	150.7512s	50
Indofood Sukses Makmur Tbk	0.09492	150.8933s	50

4.4.1.1 Pengaruh Nilai *Learning Rate*

Pada percobaan ini akan diuji coba beberapa nilai penyesuaian yaitu 0.2, 0.4, 0.6, 0.8. Nilai *learning rate* diusahakan tidak lebih dari 1, karena dapat mengakibatkan proses *training* tidak dapat mencapai titik konvergen. Jumlah *neuron hidden* yang digunakan berdasarkan tabel 4.16. Berikut ini hasil dari pengujian dengan *learning rate* yang bervariasi dan dengan jumlah *neuron* lapisan *hidden* adalah jumlah *neuron hidden* yang menghasilkan rata-rata MSE terkecil. Jumlah iterasi maksimal ditetapkan 20000 iterasi. Masing-masing *hidden neuron* dilakukan percobaan sebanyak 5 kali.

a. Hasil pelatihan pada saham Astra International Tbk

Tabel 4.17 *Learning Rate* = 0.2

Percobaan	MSE	Waktu
1	0.12295	105.6238s
2	0.12449	105.1678s
3	0.11133	105.5428s
4	0.12310	105.8021s
5	0.12806	105.7905s
rata-rata	0.12198	105.5854s

Tabel 4.18 *Learning Rate* = 0.4

Percobaan	MSE	Waktu
1	0.06394	106.4340s
2	0.09413	105.9699s
3	0.07950	105.4062s
4	0.07416	106.9838s
5	0.08992	106.3113s
rata-rata	0.08033	106.2210s

Tabel 4.19 *Learning Rate* = 0.6

Percobaan	MSE	Waktu
1	0.03438	107.1262s
2	0.07817	107.0116s
3	0.09676	110.3183s
4	0.03450	107.2419s
5	0.03685	107.5729s
rata-rata	0.05613	107.8542s

Tabel 4.20 *Learning Rate* = 0.8

Percobaan	MSE	Waktu
1	0.02864	106.7581s
2	0.03137	107.4606s
3	0.04189	106.8252s
4	0.03261	106.9560s
5	0.02890	108.4722s
rata-rata	0.03268	107.2944s

b. Hasil pelatihan pada saham H M Sampoerna Tbk

Tabel 4.21 *Learning Rate* = 0.2

Percobaan	MSE	Waktu
1	0.13121	151.6273s
2	0.11005	145.6505s
3	0.11756	145.5521s
4	0.11543	145.8495s

5	0.12684	145.7743s
rata-rata	0.12021	146.8907s

Tabel 4.22 *Learning Rate* = 0.4

Percobaan	MSE	Waktu
1	0.07274	149.9306s
2	0.08838	150.2454s
3	0.09377	150.2593s
4	0.07667	150.3970s
5	0.07845	150.1690s
rata-rata	0.08200	150.2003s

Tabel 4.23 *Learning Rate* = 0.6

Percobaan	MSE	Waktu
1	0.09463	154.4039s
2	0.07444	149.5926s
3	0.06922	150.8102s
4	0.05187	150.2963s
5	0.08602	150.7396s
rata-rata	0.07523	151.1685s

Tabel 4.24 *Learning Rate* = 0.8

Percobaan	MSE	Waktu
1	0.06347	150.5023s
2	0.05865	150.0197s
3	0.07345	150.2222s
4	0.06252	150.3206s
5	0.06022	149.9178s
rata-rata	0.06366	150.1965s

c. Hasil pelatihan pada saham Indofood Sukses Makmur Tbk

Tabel 4.25 *Learning Rate* = 0.2

Percobaan	MSE	Waktu
1	0.12830	154.2037s
2	0.14903	154.4398s
3	0.14363	153.5729s
4	0.14437	154.3750s

5	0.15067	152.1007s
rata-rata	0.14320	153.7384s

Tabel 4.26 *Learning Rate* = 0.4

Percobaan	MSE	Waktu
1	0.10093	152.0278s
2	0.13673	151.8113s
3	0.10680	151.5220s
4	0.10762	150.9259s
5	0.08875	151.0706s
rata-rata	0.10816	151.4715s

Tabel 4.27 *Learning Rate* = 0.6

Percobaan	MSE	Waktu
1	0.08993	151.6296s
2	0.10084	152.2812s
3	0.08984	151.0162s
4	0.11173	151.8831s
5	0.09458	152.1898s
rata-rata	0.09738	151.8000s

Tabel 4.28 *Learning Rate* = 0.8

Percobaan	MSE	Waktu
1	0.08753	151.7025s
2	0.08514	151.4144s
3	0.08753	151.7025s
4	0.08687	151.9745s
5	0.08811	150.7269s
rata-rata	0.08703	151.5042s

Berdasarkan hasil pelatihan dan hasil pengujian dengan nilai penyesuaian yang berbeda-beda pada tabel 4.17 sampai dengan tabel 4.28 dapat dilihat bahwa semakin besar nilai *learning rate* maka proses training akan lebih cepat (MSE menurun tajam) akan tetapi nilai *learning rate* yang terlalu besar dapat mengakibatkan garis keputusan tidak akan pernah sampai pada posisi yang optimal. Sebaliknya, nilai *learning rate* yang kecil akan mengakibatkan

proses *training* menjadi lambat akan tetapi MSE bisa terus menurun hingga garis keputusan berada pada posisi yang optimal.

MSE terkecil yang dihasilkan pada masing-masing unit untuk tiap pengujian selalu berubah-ubah. Hal ini dikarenakan karena bobot awal yang digunakan merupakan nilai acak (random). Berikut ini diberikan tabel MSE terkecil dari tiap unit dapat dilihat pada Tabel 4.29.

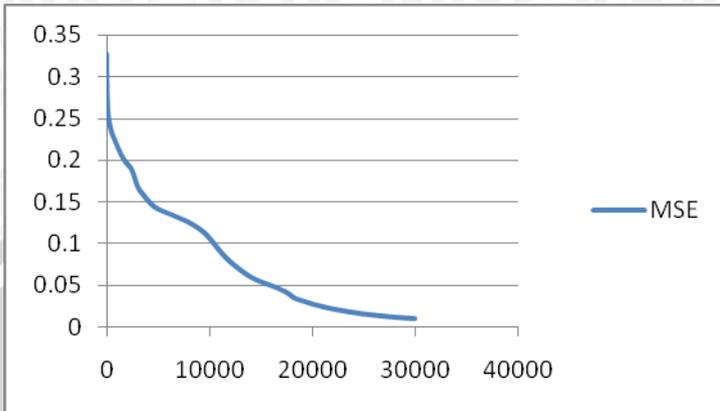
Tabel 4.29 MSE terkecil

Perusahaan	MSE	Waktu	<i>Learning rate</i>	<i>neuron hidden</i>
Astra International Tbk	0.03268	107.2944s	0.8	30
H M Sampoerna Tbk	0.06366	150.1965s	0.8	50
Indofood Sukses Makmur Tbk	0.08703	151.5042s	0.8	50

4.5 Analisa Hasil

Dari hasil pelatihan didapatkan bobot yang akan digunakan untuk menguji kemampuan sistem dalam memprediksi *trend* harga saham. Kumpulan data uji terdiri 20 data saham yang telah dilakukan pembelajaran yaitu data training saham dari bulan Januari 2008 sampai dengan bulan Juli 2008 dan 24 data yang belum pernah dilakukan pembelajaran, yaitu data saham dari bulan Januari 2009 sampai dengan bulan Juni 2009.

Berdasarkan pelatihan yang diujicobakan, untuk saham Astra, dipilih jumlah *neuron hidden* = 30, nilai *learning rate* = 0.8 dan iterasi dinaikkan menjadi 30000 iterasi agar didapatkan MSE yang lebih kecil. MSE pelatihan dapat dilihat pada Gambar 4.26.



Gambar 4.26 MSE Pelatihan Saham Astra

Berikut ini ditunjukkan hasil percobaan kebenaran prediksi *trend* saham terhadap data uji. Hasil pengujian dapat dilihat pada tabel 4.30 dan 4.31.



Tabel 4.30 Hasil Peramalan Data Training Saham Astra

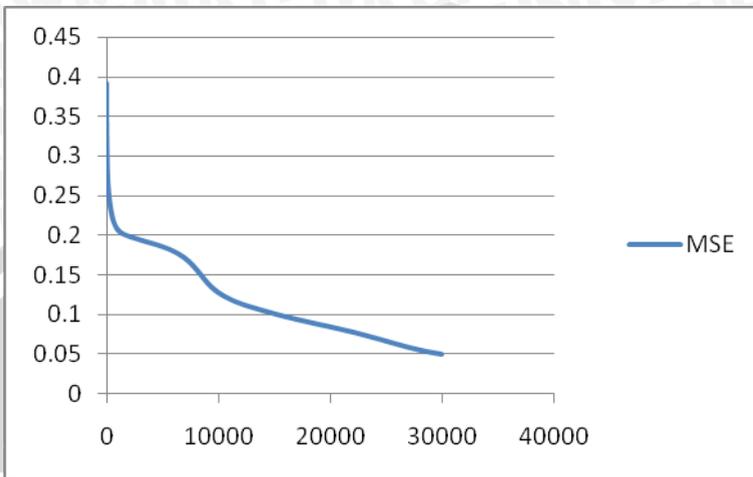
Data ke-	Hasil Prediksi	Real
1	naik berubah turun	naik berubah turun
2	naik	naik
3	naik	naik
4	naik berubah turun	naik berubah turun
5	turun	turun
6	naik	naik
7	naik berubah turun	naik berubah turun
8	naik berubah turun	naik berubah turun
9	turun	turun
10	naik	naik
11	naik	naik
12	turun	turun
13	naik berubah turun	naik berubah turun
14	naik	naik
15	turun	turun
16	naik	naik
17	naik	naik
18	turun berubah naik	turun berubah naik
19	naik	naik
20	naik	naik

Tabel 4.31 Hasil peramalan terhadap data uji saham Astra

Data ke-	Hasil Prediksi	Real
1	naik berubah turun	turun
2	naik berubah turun	naik berubah turun
3	turun berubah naik	turun
4	turun	turun
5	naik berubah turun	turun
6	turun berubah naik	turun
7	turun	naik
8	naik berubah turun	naik
9	naik berubah turun	naik
10	turun	turun berubah naik
11	turun berubah naik	naik
12	turun	naik
13	turun	turun
14	turun	naik
15	turun	turun
16	turun	naik
17	naik berubah turun	naik
18	turun	turun
19	turun	naik
20	turun berubah naik	turun berubah naik
21	turun	naik
22	turun	naik
23	naik berubah turun	turun
24	turun	turun berubah naik

Untuk data saham yang pernah dilakukan pembelajaran sebelumnya, sistem dapat mengenali dengan benar 100% dari data saham Astra. Untuk data uji yang belum pernah dilakukan pembelajaran kedalam sistem, sistem dapat memprediksi dengan benar sebanyak 25% data uji saham Astra.

Sedangkan untuk saham Sampoerna dipilih jumlah *neuron hidden* = 50, nilai *learning rate* = 0.8 dan iterasi dinaikkan menjadi 30000 iterasi agar didapatkan MSE yang lebih kecil. MSE pelatihan dapat dilihat pada Gambar 4.27.



Gambar 4.27 MSE Pelatihan Saham Sampoerna

Hasil pengujian terhadap saham Sampoerna terhadap data uji dapat dilihat pada tabel 4.32 dan 4.33.

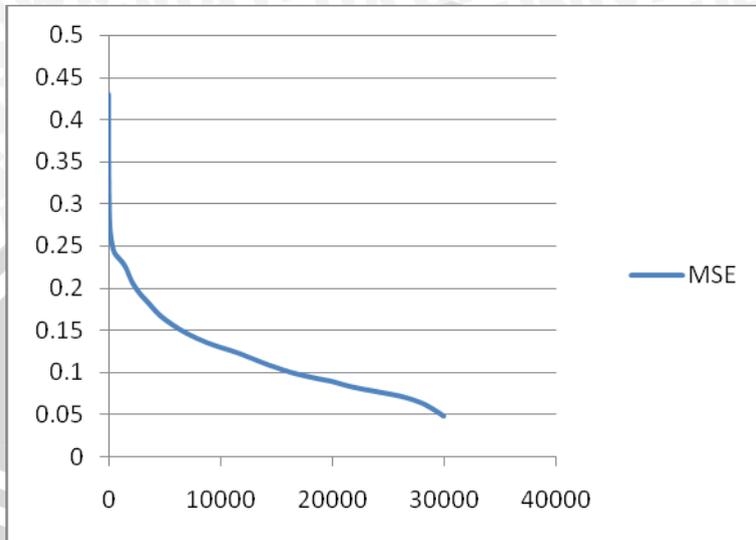
Tabel 4.32 Hasil Peramalan Data Training Saham Sampoerna

Data ke-	Hasil Prediksi	Real
1	turun	turun
2	naik berubah turun	naik berubah turun
3	turun	turun
4	turun	turun
5	naik	naik
6	turun	turun
7	naik	naik
8	naik	naik
9	turun	turun
10	naik	naik
11	turun	turun
12	turun	turun
13	naik	naik
14	turun	turun
15	naik	naik
16	turun	turun
17	turun	turun
18	turun	turun
19	turun	turun
20	turun	turun

Tabel 4.33 Hasil peramalan terhadap data uji saham Sampoerna

Data ke-	Hasil Prediksi	Real
1	turun	turun
2	turun	turun berubah naik
3	turun	naik
4	turun	turun
5	turun	naik
6	naik berubah turun	turun berubah naik
7	turun berubah naik	turun
8	naik	naik
9	turun	naik
10	turun	turun
11	turun	naik
12	turun	naik
13	turun	naik
14	tidak dikenali	turun
15	turun berubah naik	naik
16	turun	turun
17	turun	naik
18	turun	turun
19	turun	turun
20	turun	turun
21	turun	turun
22	turun	turun
23	turun	turun
24	turun	turun

Untuk data saham yang pernah dilakukan pembelajaran sebelumnya, sistem dapat mengenali dengan benar 100% dari data saham Sampoerna. Untuk data uji yang belum pernah dilakukan pembelajaran kedalam sistem, sistem dapat memprediksi dengan benar sebanyak 50% data uji saham Sampoerna. Untuk saham Indofood dipilih jumlah *neuron hidden* = 50, nilai *learning rate* = 0.8 dan iterasi dinaikkan menjadi 30000 iterasi agar didapatkan MSE yang lebih kecil. MSE pelatihan dapat dilihat pada Gambar 4.28.



Gambar 4.28 MSE Pelatihan Saham Indofood

Hasil pengujian terhadap saham Indofood terhadap data uji dapat dilihat pada tabel 4.34 dan 4.35.

Tabel 4.34 Hasil Peramalan Data Training Saham Indofood

Data ke-	Hasil Prediksi	Real
1	turun	turun
2	naik	naik
3	naik	naik
4	turun	naik
5	turun berubah naik	turun berubah naik
6	naik	naik
7	naik	naik
8	naik	naik
9	turun	turun
10	naik	naik
11	naik	naik
12	naik	naik
13	turun	turun
14	turun	turun
15	naik	naik
16	naik berubah turun	naik berubah turun
17	turun	turun
18	turun	turun
19	naik	naik
20	naik	naik

Tabel 4.35 Hasil peramalan terhadap data uji saham Indofood

Data ke-	Hasil Prediksi	Real
1	turun	turun
2	turun	turun
3	turun	turun
4	naik berubah turun	turun berubah naik
5	turun	turun
6	turun	turun
7	turun	turun
8	naik berubah turun	naik
9	turun	naik
10	turun	turun berubah naik
11	turun	naik berubah turun
12	tidak dikenali	naik
13	turun	turun
14	naik berubah turun	naik
15	turun berubah naik	naik
16	turun	turun berubah naik
17	turun	naik
18	turun	turun
19	turun	naik
20	turun berubah naik	turun
21	turun	naik berubah turun
22	turun	naik
23	naik berubah turun	turun
24	turun	turun berubah naik

Untuk data saham yang pernah dilakukan pembelajaran sebelumnya, sistem dapat mengenali dengan benar 95.8% dari data saham Sampoerna. Untuk data uji yang belum pernah dilakukan pembelajaran kedalam sistem, sistem dapat memprediksi dengan benar sebanyak 33.33% data uji saham Indofood.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang diperoleh dari tugas akhir ini:

1. Dari hasil uji coba pelatihan dengan jumlah *neuron* pada lapisan *hidden* yang bervariasi, didapatkan arsitektur JST yang berbeda-beda. Untuk data saham Astra menggunakan 30 *neuron* pada *hidden neuron*. Sedangkan data saham Sampoerna dan Indofood menggunakan 50 *neuron* pada *hidden neuron*. Perbedaan arsitektur JST pada tiap-tiap unit karena bobot awal pelatihan diperoleh secara acak.
2. Setelah dilakukan pelatihan dengan 20000 iterasi, didapatkan nilai MSE yang berbeda-beda. Dengan mengujicobakan beberapa nilai *learning rate* (0.2, 0.4, 0.6, dan 0.8) dihasilkan nilai MSE yang berbeda-beda dan dari tiap perusahaan diambil nilai MSE terkecil (Tabel 4.29), sehingga didapatkan nilai *learning rate* untuk saham Astra, Sampoerna, dan Indofood adalah 0.8.
3. Untuk data yang belum pernah dilakukan pembelajaran sebelumnya hasil prediksi sistem masih belum akurat. Untuk saham Astra sistem memprediksi dengan benar 25% dari 24 data uji. Untuk saham Sampoerna sistem memprediksi dengan benar 50% dari 24 data uji. Sedangkan untuk saham Indofood sistem memprediksi dengan benar 33.33% dari 24 data uji. Hal ini mungkin diakibatkan data training yang masih sedikit dan kurang proporsial. Dapat juga diakibatkan perbedaan karakteristik antara data training dan data uji sehingga sistem tidak dapat memprediksi data uji dengan akurat.
4. Kurangnya keakuratan prediksi arah *trend* harga saham dapat diakibatkan karena hanya faktor teknis yang dimasukkan dalam sistem. Arah *trend* pergerakan harga saham dipengaruhi oleh faktor teknis dan fundamental, sehingga faktor fundamental juga harus dimasukkan ke dalam sistem untuk hasil yang lebih optimal.

5.2 Saran

Saran yang diberikan pada pembuatan skripsi ini:

1. Dalam tugas akhir ini, sistem hanya menggunakan data-data teknis untuk melakukan prediksi. Untuk percobaan berikutnya akan lebih baik apabila faktor fundamental seperti sentimen pasar dan kestabilan perusahaan juga dimasukkan kedalam sistem.
2. Dalam sistem ini nilai output dari sistem yang sebenarnya berjumlah 2^4 atau 16 output hanya 4 yang dipakai sebagai output. Dalam penelitian berikutnya, sebaiknya semua kelas dari output dapat dipakai agar hasil prediksi dapat lebih akurat.
3. Data training masing-masing perusahaan sebaiknya diperbanyak sehingga dapat mewakili karakteristik data yang akan diujikan. Variasi data untuk pelatihan pada masing-masing perusahaan juga dapat diperbanyak sehingga dapat memprediksi *trend* harga saham dengan akurat.



DAFTAR PUSTAKA

- Anwar, S. 2008. Pembangunan Aplikasi Analisis Saham Menggunakan Metode *Rate of Change*. Teknik Informatika ITB. Bandung
- Bambang, Budi DP, dkk. 1999. Teknik Jaringan Syaraf Tiruan Feedforward Untuk Prediksi Harga Saham Pada Pasar Modal Indonesia. Jurnal Informatika Vol. 1, No. 1, Mei 1999: 11 – 22
- DARPA. Neural Network Study. 1988. AFCEA International Press, p. 60
- Eliyani. Pengantar Jaringan Syaraf Tiruan. 2005. Materikuliah.com.
- Freeman, J. Skapura, D. 1991. Neural Networks Algorithms, Applications, and Programming Techniques. Addison-Wesley. USA.
- Haykin, S. 1994, Neural Networks: A Comprehensive Foundation. Macmillan. New York
- Kristanto, Andri. 2004. Jaringan Syaraf Tiruan (Konsep Dasar, Algoritma, dan Aplikasi). GAYAMEDIA. Yogyakarta.
- Muis, S. 2006. Teknik Jaringan Syaraf Tiruan. GRAHA ILMU. Yogyakarta.
- Muis, S. 2006. Jaringan Syaraf Tiruan Sebagai Alat Bantu Peramalan Harga Saham. GRAHA ILMU. Yogyakarta.
- Suyanto. 2008. Soft Computing Membangun Mesin Ber-IQ Tinggi. Informatika Bandung. Bandung.
- Zurada, J.M. 1992. Introduction To Artificial Neural Systems. PWS Publishing Company. Boston

<http://arsasi.wordpress.com/2008/03/16/jenis-jenis-saham>. Diakses pada tanggal 10 Januari 2009

UNIVERSITAS BRAWIJAYA



Lampiran 1

Data uji saham Astra Januari-Juni 2009

minggu ke-	senin	selasa	rabu	kamis	Jumat	f. siklus	trend
m1jan	1220 0	1255 0	1350 0	1295 0	1245 0	0.2	turun
m2jan	1295 0	1275 0	1290 0	1250 0	1245 0	0.8	naik berubah turun
m3jan	1250 0	1255 0	1265 0	1255 0	1240 0	0.2	turun
m4jan	0	1310 0	1290 0	1290 0	1300 0	0.6	turun
m1feb	1275 0	1245 0	1170 0	1185 0	1190 0	0.2	turun
m2feb	1195 0	1165 0	1090 0	1105 0	1095 0	0.8	turun
m3feb	1110 0	1080 0	1095 0	1110 0	1090 0	0.8	naik
m4feb	1095 0	1095 0	1135 0	1135 0	1130 0	0.8	naik
m1mar	1085 0	1080 0	1100 0	1115 0	1125 0	0.2	naik
m2mar	0	1130 0	1190 0	1245 0	1300 0	0.4	turun berubah naik
m3mar	1310 0	1260 0	1340 0	1435 0	1370 0	0.8	naik
m4mar	1490 0	1525 0	1525 0	0	1580 0	0	naik
m1apr	1425 0	1425 0	1545 0	1570 0	1560 0	0.2	turun
m2apr	1525 0	1465 0	1450 0	0	0	0.2	naik
m3apr	1525 0	1550 0	1545 0	1585 0	1580 0	0.4	turun
m4apr	1595 0	1550 0	1530 0	1565 0	1545 0	0.2	naik
m5apr	1510 0	1520 0	1575 0	1800 0	1770 0	0.8	naik
m1mei	1855 0	1900 0	1895 0	1900 0	2155 0	0.2	turun
m2mei	1975 0	1930 0	1915 0	1835 0	1830 0	0.2	naik
m3mei	1810 0	1975 0	2000 0	0	1975 0	0.8	turun berubah naik
m4mei	1965 0	1910 0	2075 0	2175 0	2080 0	0.8	naik
m1jun	2195 0	2215 0	2210 0	2260 0	2415 0	0.4	naik

m2jun	2370 0	2470 0	2450 0	2360 0	2390 0	0.8	turun
m3jun	2370 0	2320 0	2330 0	2120 0	2205 0	0.2	turun berubah naik

Lampiran 2

Data uji saham Sampoerna Januari-Juni 2009

minggu ke-	senin	selasa	rabu	kamis	Jumat	f. siklus	trend
m1jan	9300	9500	9600	9700	9800	0.2	turun
m2jan	9850	9600	9850	9600	9800	0.2	turun berubah naik
m3jan	8800	8500	9750	1000 0	9900	0.2	naik
m4jan	9900	9900	9950	1020 0	0	0.6	turun
m1feb	9900	9900	9900	9900	9300	0.2	naik
m2feb	9500	9900	9300	9600	9850	0.2	turun berubah naik
m3feb	9900	9700	1000 0	9500	1030 0	0.8	turun
m4feb	1030 0	1000 0	9900	9900	1020 0	0.2	naik
m1mar	9500	9500	1020 0	1030 0	1030 0	0.2	naik
m2mar	0	9400	1050 0	1050 0	1050 0	0.2	turun
m3mar	1050 0	1015 0	9950	9850	9900	0.4	naik
m4mar	1050 0	1050 0	1050 0	0	1050 0	0	naik
m1apr	1050 0	1080 0	1080	1050 0	1070 0	0.2	naik
m2apr	1020 0	1070 0	1060 0	0	0	0.2	turun
m3apr	1100 0	1100 0	1010 0	1100 0	1100 0	0.8	naik
m4apr	1100 0	1150 0	1100 0	1100 0	1100 0	0.4	turun
m5apr	1100 0	1080 0	1080 0	1050 0	1080 0	0.2	naik
m1mei	1100 0	1130 0	1120 0	1120 0	1120 0	0.8	turun

m2mei	1100 0	1100 0	1100 0	1100 0	1020 0	0.2	turun
m3mei	1100 0	1080 0	1080 0	1080 0	1040 0	0.2	turun
m4mei	1050 0	1060 0	1045 0	1040 0	1040 0	0.2	turun
m1jun	1050 0	1050 0	1045 0	1045 0	1040 0	0.8	turun
m2jun	1020 0	1020 0	1020 0	1000 0	1000 0	0.2	turun
m3jun	9900	9900	9900	9600	9500	0.2	turun

Lampiran 3

Data uji saham Indofood Januari-Juni 2009

minggu ke-	senin	selasa	rabu	kamis	Jumat	f. siklus	trend
m1jan	980	1020	1020	1030	1090	0.8	turun
m2jan	1110	1100	1080	1030	1060	0.8	turun
m3jan	1050	1040	1000	1000	1000	0.2	turun
m4jan	1030	1020	1010	980	0	0.2	turun berubah naik
m1feb	960	960	970	960	1020	0.8	turun
m2feb	1000	980	970	960	960	0.8	turun
m3feb	960	940	930	930	900	0.8	turun
m4feb	920	900	880	880	880	0.8	naik
m1mar	840	850	870	860	860	0.8	naik
m2mar	0	860	870	860	880	0.6	turun berubah naik
m3mar	870	860	880	890	920	0.2	naik berubah turun
m4mar	970	990	960	0	960	0	naik
m1apr	940	940	920	960	930	0.2	turun
m2apr	970	960	960	0	0	0.8	naik
m3apr	1040	1040	1080	1080	1070	0.8	naik
m4apr	1080	1200	1180	1200	1250	0.2	turun berubah naik

m5apr	1220	1180	1250	1280	1250	0.8	naik
m1mei	1370	1370	1450	1570	1630	0.8	turun
m2mei	1600	1600	1600	1480	1450	0.8	naik
m3mei	1580	1700	1800	0	1780	0.8	turun
m4mei	1780	1720	1760	1790	1780	0.8	naik berubah turun
m1jun	1920	1930	1900	1850	1910	0.2	naik
m2jun	1860	1890	1950	1930	1890	0.2	turun
m3jun	1820	1790	1810	1770	1770	0.2	turun berubah naik

