

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini, perkembangan perekonomian dan bisnis khususnya pasar modal tidak lepas dari teknologi[MUS-09]. Banyak teknologi yang bisa dimanfaatkan untuk memudahkan penyelesaian masalah dalam perekonomian, seperti menghitung, mengolah data, hingga meramal atau memprediksi perilaku perekonomian. Teknologi sangat bermanfaat, diantaranya bisa diterapkan untuk memprediksi tren harga saham. Prediksi tren harga saham dapat dimanfaatkan para investor ketika ingin menanam saham agar memperoleh keuntungan yang sebanyak-banyaknya dengan resiko yang sekecil-kecilnya, selain itu para investor juga dapat mengetahui berbagai teknik untuk menganalisa saham.

Ada banyak cara yang dilakukan untuk meramalkan saham, salah satunya yaitu jaringan saraf tiruan. Jaringan saraf tiruan terdiri atas sebuah proses sederhana yang disebut dengan node atau simpul yang ada di otak manusia. Kumpulan dari simpul-simpul ini membentuk suatu konfigurasi yang dikenal sebagai jaringan saraf tiruan[ELI-07]. Jaringan saraf tiruan menggunakan algoritma pembelajaran yang disebut *backpropagation*. Proses pembelajaran dalam *backpropagation* dilakukan dengan penyesuaian bobot-bobot jaringan saraf tiruan dengan arah mundur berdasarkan nilai *error* dalam proses pembelajaran[KUS-09]. Algoritma ini selalu dikembangkan salah satu diantaranya adalah *Al-Alaoui Backpropagation*.

Algoritma *Al-Alaoui Backpropagation* adalah implementasi Algoritma *Al-Alaoui* kedalam sebuah jaringan saraf tiruan. Algoritma *Al-Alaoui Backpropagation* lebih sederhana dari *backpropagation* biasa karena algoritma ini hanya melakukan modifikasi terkait dengan populasi data latih dalam setiap iterasi sehingga dapat mempercepat konvergensi[ALA-07].

Penelitian lain yang menggunakan algoritma *Al-Alaoui backpropagation* pernah dilakukan oleh Toni Tegar Sahidi [SAH-12], menghasilkan akurasi sebesar 94%. Karena akurasinya lebih tinggi dan proses konvergensinya lebih cepat daripada *backpropagation* biasa maka metode *Al-Alaoui Backpropagation* dapat

digunakan untuk mengatasi data-data fluktuatif dalam memprediksi tren harga saham. Berdasarkan uraian di atas, maka skripsi ini diberi judul “**PENGUNAAN ALGORITMA AL-ALAOUİ BACKPROPAGATION UNTUK PREDIKSI TREN HARGA SAHAM**”.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, dapat diambil rumusan masalah sebagai berikut :

1. Bagaimana mengimplementasikan algoritma *Al-Alaoui Backpropagation* untuk memprediksi harga saham.
2. Berapa tingkat akurasi yang dihasilkan *Al-Alaoui Backpropagation* untuk memprediksi harga saham.

1.3 Batasan Masalah

Berdasarkan permasalahan diatas akan diberikan batasan masalah sebagai berikut :

- Input sistem berupa data saham BCA yahoo *finance* dari awal agustus tahun 2011 sampai dengan akhir tahun 2012.
- Data yang diprediksi adalah data *open* di hari berikutnya.

1.4 Tujuan Penelitian

Berdasarkan rumusan diatas, tujuan dari penelitian ini adalah sebagai berikut :

1. Mengimplementasikan sistem jaringan syaraf tiruan *Al-Alaoui backpropagation* untuk melakukan prediksi saham.
2. Mengetahui tingkat akurasi sistem prediksi saham menggunakan jaringan syaraf tiruan *Al-Alaoui backpropagation*.

1.5 Manfaat

Manfaat dari penelitian ini adalah :

1. Investor dapat memprediksi harga saham di masa depan dan memudahkan dalam investasi saham.

2. Penulis dapat mengetahui lebih dalam tentang jaringan saraf tiruan dan variasinya.

1.6 Sistematika Penulisan

Skripsi ini disusun berdasarkan sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini berisi teori-teori yang berhubungan dengan saham serta unsur-unsurnya, dan juga metode *Al-Alaoui Backpropagation*.

BAB III METODE PENELITIAN/KAJIAN

Bab ini menjelaskan mengenai metode penelitian/kajian sistem tentang prediksi saham menggunakan *Al-Alaoui Backpropagation*.

BAB IV PERANCANGAN IMPLEMENTASI

Bab ini berisi analisa kebutuhan dan perancangan sistem untuk prediksi saham menggunakan *AL-Alaoui Backpropagation*.

BAB V PENGUJIAN DAN ANALISIS

Bab ini berisi hasil implementasi perangkat lunak, pengujian, dan analisis sistem.

BAB VI PENUTUP

Bab ini berisi kesimpulan dari penelitian yang dilakukan serta saran-saran untuk pengembangan penelitian lebih lanjut.

BAB II

Kajian Pustaka dan Dasar Teori

Peneliti mengambil beberapa penelitian terdahulu untuk dijadikan dasar penelitian saat ini. Diantaranya adalah Penelitian yang dilakukan oleh Mohamad Adnan Al-Alaoui dan Rony Ferzli[ALA-07]. Penelitian ini bertujuan untuk mengimplementasikan algoritma *Al-Alaoui* kedalam *Backpropagation* untuk mempercepat konvergensi dalam melakukan *subsampling image compression*. Dan didapatkan hasil yang menunjukkan bahwa algoritma ini dapat mempercepat konvergensi, tidak ada kesalahan dalam klasifikasi pixel. Selain itu dalam penelitian yang dilakukan oleh Tony Sahidi[SAH-12]. Penelitian ini menerapkan algoritma Mohamad Adnan Al-Alaoui yaitu *Al-Alaoui Backpropagation*. Data yang digunakan adalah *Berlin database of Emotional Speech*. Data tersebut terdiri dari 10 pembicara yang mengucapkan kalimat dengan emosi yang berbeda-beda. Setelah diujikan didapatkan hasil tingkat akurasi tertinggi dengan menggunakan *hidden layer* antara 5-22 buah *neuron*. Dan akurasi rata-rata yang didapatkan adalah sebesar 94 persen. Begitu juga penelitian yang dilakukan oleh eliyani[ELI-07]. Penelitian ini menggunakan *neural network* dan *akaike information criterion* untuk melakukan peramalan harga saham. Dalam penelitian tersebut menggunakan jaringan saraf tiruan *backpropagation*, dan menghasilkan hasil yang baik, didapatkan nilai terbaik dengan menggunakan input layer 5, hidden layer 30, dan jumlah neuron pada output 1.

2.1 Prediksi

Prediksi adalah perkiraan tentang suatu kejadian pada saat ini atau masa depan. Prediksi pada saat ini digunakan untuk memperkiraan kejadian dimana faktor-faktor yang diperhatikan saling berkaitan, contohnya prediksi suatu penyakit jantung. Ketika memprediksi penyakit jantung, pertama yang akan dilihat adalah gejala-gejalanya, seperti denyut jantung tidak teratur, sesak napas dan lain sebagainya. Prediksi masa depan sama halnya dengan peramalan yaitu memprediksi dengan data historis yang diproyeksikan untuk menyimpulkan

pristiwa dimasa mendatang dengan model matematis. Prediksi yang digunakan dalam tugas akhir ini merupakan prediksi untuk masa mendatang.

2.2 Saham

2.2.1 Pengertian Saham

Saham adalah sertifikat yang menunjukkan bukti kepemilikan suatu perusahaan, dan pemegang saham memiliki hak klaim atas penghasilan dan aktiva perusahaan. Saat *investor* membeli saham di suatu perusahaan, *investor* dapat menerima pembagian keuntungan sesuai saham yang telah dibeli *investor*. Pembagian keuntungan kepada pemegang saham atau *investor* disebut *dividen*. Pembayaran dividen tidak diwajibkan, manajemen perusahaan berhak menggunakannya untuk membeli aset atau pemasaran, sehingga mendapatkan untung yang lebih besar[BAP-06].

2.2.2 Parameter Saham

Dalam saham ada 4 parameter yaitu harga buka, harga tutup, harga minimum dan harga maksimum.

2.2.3 Faktor yang Mempengaruhi Pergerakan Saham

Dalam melakukan jual beli saham, hal tersulit adalah menentukan waktu yang tepat untuk membeli dan menjualnya, ada 3 faktor utama yang memicu pergerakan saham sehingga menyebabkan fluktuasi data[THE-12]:

2.2.3.1 Faktor Fundamental

Dua dari beberapa variabel terkait menilai harga saham yaitu:

1. Earning per Share

Earning per Share adalah jumlah keuntungan dari lembar saham biasa dengan ketentuan laba bersih yang telah dikurangi dividen perusahaan dibagi dengan lembar saham yang beredar.

2. Price Earning Ratio

Price Earning Ratio adalah perbandingan antara harga perlembar saham dengan pendapatan masing-masing lembar saham pada suatu perusahaan.

2.2.3.2 Indikator Ekonomi

Indikator ekonomi terdiri dari beberapa variable, diantaranya adalah nilai tukar rupiah terhadap dolar, inflasi, dan suku bunga bank[THE-12]. Nilai tukar adalah pencerminan dari keseimbangan antara mata uang rupiah dengan mata uang dolar. Inflasi adalah meningkatnya jumlah uang yang beredar dimasyarakat pada suatu periode tertentu tanpa adanya kenaikan produksi barang. Suku bunga bank adalah kebijakan moneter yang ditetapkan oleh Bank Indonesia untuk dipublikasikan.

2.2.3.3 Resiko Sistematis

Risiko sistematis merupakan resiko yang tidak dapat didiversifikasikan atau resiko yang tidak dapat dihilangkan.

2.3 Jaringan Saraf Tiruan

Awal mula jaringan saraf tiruan ini diperkenalkan pada tahun 1943 oleh ahli saraf yang bernama Warren McCulloch[ELI-07].

2.3.1 Pengertian Jaringan Saraf Tiruan

Jaringan saraf tiruan adalah suatu metode komputasi yang menggunakan model tiruan dalam sistem jaringan saraf biologi[ELI-07]. Jaringan saraf tiruan merupakan sistem adaptif yang merubah strukturnya untuk memecahkan masalah berdasarkan informasi internal maupun external[SUH-10]. Jaringan saraf tiruan digunakan untuk mengetahui pola-pola data dengan memodelkan input dan output. Jaringan saraf tiruan dibentuk dengan menggunakan model generalisasi matematika, dengan asumsi bahwa:

- Pemrosesan informasi terjadi pada banyak elemen sederhana.
- Sinyal dikirim antara neuron melalui penghubung-penghubungnya.

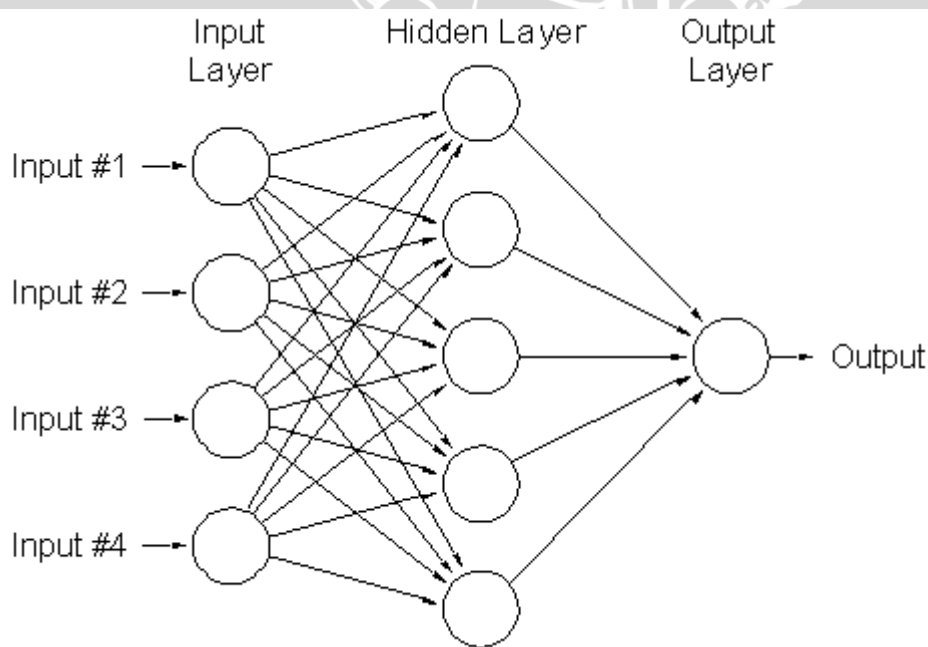
- Penghubung antar neuron mempunyai bobot masing-masing.
- Untuk menentukan *Output* setiap *neuron* menggunakan fungsi aktivasi yang dikenakan pada jumlah maksimum yang diterima. Besar *output* tersebut akan dibandingkan dengan batas ambang.

Jaringan saraf tiruan dapat ditentukan dengan 3 hal, yaitu[DED-12].

1. Pola hubungan antar *neuron*.
2. Metode Training yaitu metode untuk menentukan bobot penghubung.
3. Fungsi aktivasi yaitu fungsi untuk menentukan suatu keluaran neuron.

2.3.2 Model Dasar Jaringan Saraf Tiruan

Model dasar jaringan saraf tiruan dimodelkan dari sistem saraf biologi, model menerima masukan. Dan setiap masukan memiliki bobot tersendiri. Setiap sel saraf mempunyai nilai ambang. Jumlah bobot dari masukan dikurangi dengan nilai ambang pada sel saraf akan menghasilkan suatu aktifasi yang kemudian menjadi fungsi aktivasi untuk menentukan hasil keluaran. Gambar fungsi aktifasi dapat dilihat pada gambar 2.1



Gambar 2.1 Gambar fungsi aktifasi

Dari gambar diatas dapat diketahui jaringan saraf tiruan memiliki 3 bagan, yaitu input layer, hidden layer, dan output layer. Masing-masing bagian memiliki neuron-neuron, jumlah neuron dapat berubah-ubah sesuai data. Input memiliki bobot tersendiri yang akan diproses di hidden layer(f) untuk menghasilkan output, seperti pada jaringan saraf manusia, dari otak diterjemahkan oleh saraf untuk diteruskan ke motorik. 3 bagian penting yang harus selalu berhubungan ialah masukan, sel tersembunyi, dan keluaran.

2.3.3 Jumlah Hidden Neuron

Ada banyak cara untuk menentukan jumlah hidden neuron. Salah satunya adalah Jumlah hidden neuron berjumlah $2/3$ dari input layer ditambah jumlah output layer [HEA-08].

2.3.4 Proses Pembelajaran

Pada umumnya pada jaringan saraf tiruan, hubungan antara input dan output diketahui secara pasti untuk dijadikan model. Proses hubungan antara input dan output dilakukan dengan pembelajaran. Pembelajaran dalam jaringan saraf tiruan terbagi menjadi dua yaitu pembelajaran terawasi dan pembelajaran tidak terawasi.

Pada pembelajaran terawasi output yang diharapkan sudah diketahui. Pembelajaran disini dilakukan menggunakan data yang sudah ada, seringkali digunakan untuk memprediksi kedepan.

Pada pembelajaran tidak terawasi tidak diperlukan target output. Hasil tidak dapat ditentukan seperti yang diharapkan selama proses ini. Selama proses pembelajaran nilai bobot disusun berdasarkan input yang diberikan. Tujuan pembelajaran ini adalah mengelompokan unit-unit yang sama sehingga cocok untuk dijadikan klasifikasi[ELI-05].

2.3.5 Algoritma *Backpropagation*

Algoritma *Backpropagation* merupakan salah satu variasi dalam jaringan saraf tiruan. Perbedaannya dengan jaringan saraf tiruan yang lain ialah proses pembelajarannya dilakukan dengan penyesuaian bobot-bobot jaringan saraf tiruan dengan arah mundur dan didasarkan oleh *error* ketika proses pembelajaran [KUS-09]. *Backpropagation* bekerja secara iterative dengan menggunakan sekumpulan data, dari tiap data dalam setiap proses yang dilakukan dicari bobot relasinya, lalu dimodifikasi agar nilai *Mean Squared Error (MSE)* antara jaringan dengan nilai sesungguhnya seminimal mungkin.

Algoritma pelatihan untuk *backpropagation* dengan satu *layer* tersembunyi (dengan fungsi aktivasi sigmoid biner) sebagai berikut [SIA-05]:

- Langkah 0
Inisialisasi bobot dengan bilangan acak nilai kecil
- Langkah 1
Ulangi langkah pelatihan hingga kondisi akhir iterasi terpenuhi
- Langkah 2
Untuk setiap pasang *training data* lakukan langkah 3 – langkah 8

Fase I : *Feedforward*

- Langkah 3
Setiap unit masukan ($x_i, i = 1, \dots, n$) menerima sinyal dan meneruskan x_i ke unit-unit lapisan tersembunyi.

- Langkah 4
Hitung semua keluaran pada unit lapisan tersembunyi $z_j (j=1, 2, \dots, p)$

$$z_{\text{net}_j} = v_{j0} + \sum_{i=1}^n x_i \cdot v_{ji} \quad (2.1)$$

$$z_j = f(z_{\text{net}_j}) = \frac{1}{1 + \exp(-z_{\text{net}_j})} \quad (2.2)$$

- Langkah 5
Hitung semua keluaran pada unit $y_k (k = 1, 2, \dots, m)$

$$y_{\text{net}_k} = w_{k0} + \sum_{j=1}^p z_j \cdot w_{kj} \quad (2.3)$$

$$y_k = f(y_{\text{net}_k}) = \frac{1}{1 + \exp(-y_{\text{net}_k})} \quad (2.4)$$

Fase II : *Backpropagation*

- Langkah 6

Hitung faktor δ_{unit} keluaran berdasarkan kesalahan pada setiap unit keluaran

y_k dimana $k = 1, 2, \dots, m$

$$\delta_k = (t_k - y_k) f'(y_{\text{net}_k}) = (t_k - y_k) y_k (1 - y_k) \quad (2.5)$$

δ_k merupakan unit kesalahan yang dipakai dalam perubahan bobot layar di bawahnya (langkah 7)

Hitung perubahan bobot w_{kj} dengan percepatan α

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

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

- Langkah 7

Hitung faktor δ_{layar} tersembunyi berdasarkan kesalahan pada setiap layar tersembunyi z_j dimana $j = 1, 2, \dots, p$

$$\delta_{\text{net}_j} = \sum_{k=1}^m \delta_k \cdot w_{kj} \quad (2.7)$$

Faktor δ_{layar} tersembunyi :

$$\delta_j = \delta_{\text{net}_j} \cdot f'(z_{\text{net}_j}) = \delta_{\text{net}_j} \cdot z_j (1 - z_j) \quad (2.8)$$

Hitung perubahan bobot v_{ji}

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (2.9)$$

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

Fase III : *WeightUpdate*

- Langkah 8

Hitung perubahan bobot garis yang menuju unit keluaran

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (2.10)$$

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

Hitung perubahan bobot garis yang menuju lapisan tersembunyi

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (2.11)$$

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

2.3.6 Algoritma Al-Alaoui Backpropagation

Untuk jaringan saraf tiruan dengan banyak cabang yang memakai *backpropagation*, konvergensi diperlukan untuk menyelesaikan banyak masalah klasifikasi, tetapi tidak diperlukan adanya peningkatan dalam pengklasifikasian. Disini bisa diimplementasikan algoritma Al-Alaoui untuk mempercepat konvergensi tersebut dengan cara mengenalkan kembali, *cloning* contoh error pada klasifikasi[ALA-07].

Langkah-langkah algoritma Al-Alaoui adalah sebagai berikut:

1. Inisialisasi parameter dan menentukan nilai *error* dari klasifikasi.
2. Menjalankan Backpropagation untuk mendefinisikan maximum dari epoch.
3. Apabila maximum epoch telah terpenuhi maka berhenti, jika tidak lanjut ke langkah 4.
4. Tes input asli sebelum di kloning dan ditentukan persenan klasifikasi yang *error*.
5. Jika persenan errornya sudah dicapai maka berhenti, jika tidak lanjut ke langkah 6.
6. Kloning contoh klasifikasi yang eror dari data yang asli. Setelah itu dimasukan ke *array* masukan dan mulai lagi dari langkah 2

2.3.7 Tingkat Akurasi

Akurasi menyatakan seberapa dekat suatu hasil pengukuran terhadap angka sebenarnya. Semakin besar tingkat akurasi maka semakin akurat proses klasifikasi tersebut. Tingkat akurasi sebuah sistem dapat dihitung dengan menggunakan persamaan di bawah ini :

$$\text{Tingkat akurasi} = \frac{\sum \text{data uji benar}}{\sum \text{jumlah total data uji}} \times 100\% \quad (2.12)$$

Metode lain yang dapat digunakan untuk mengukur tingkat akurasi adalah *Mean Squared Error* (MSE). MSE adalah metode untuk mengevaluasi dengan cara

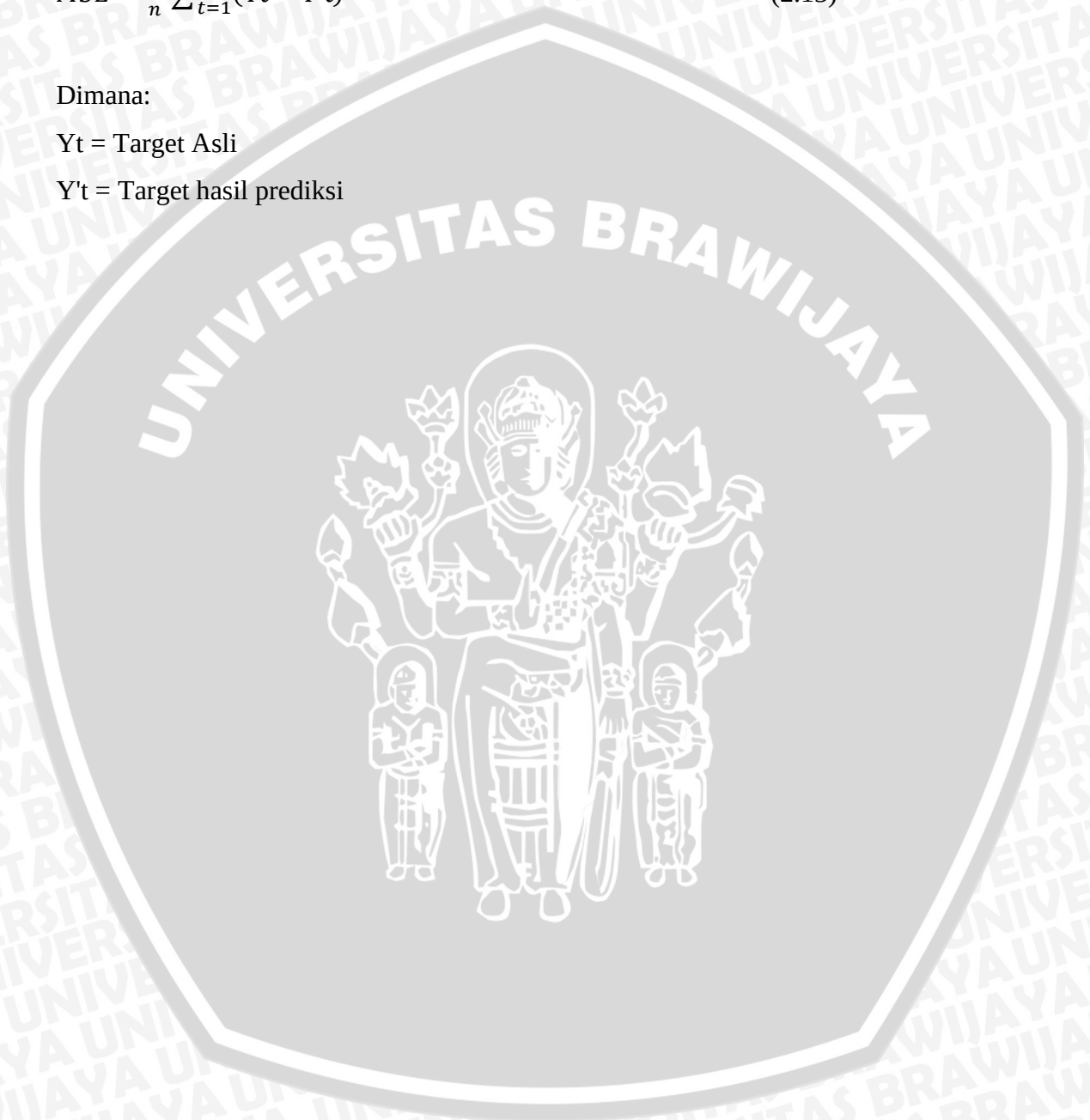
masing-masing kesalahan atau sisa dikuadratkan, kemudian dijumlahkan dan dibagi dengan jumlah observasi. Persamaan MSE dapat dilihat pada persamaan 2.13.

$$MSE = \frac{1}{n} \sum_{t=1}^n (Y_t - Y'_t)^2 \quad (2.13)$$

Dimana:

Y_t = Target Asli

Y'_t = Target hasil prediksi



BAB III

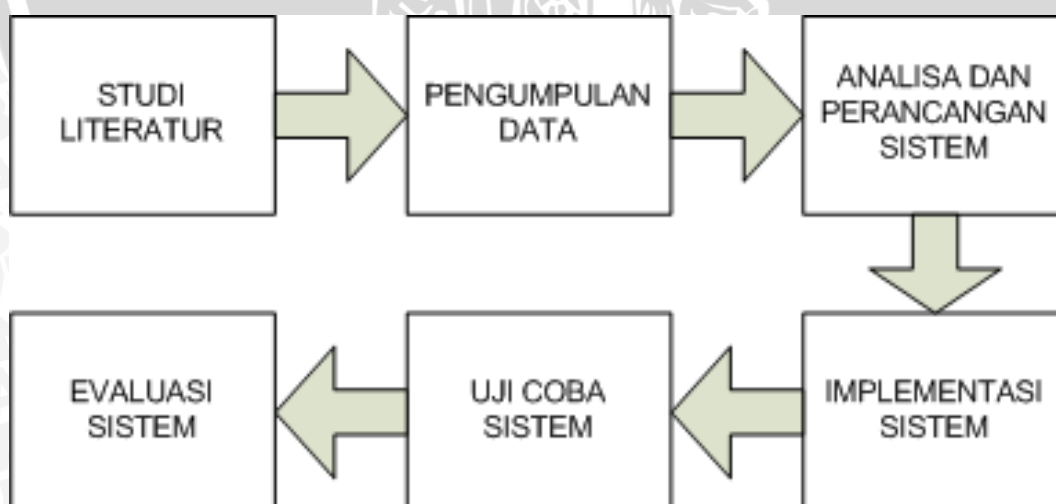
METODE PENELITIAN DAN PERANCANGAN

Dalam bab metodologi dan perancangan ini akan dibahas metode perancangan dan langkah-langkah yang dilakukan dalam penelitian menggunakan metode *Al-Alaoui Backpropagation* dalam prediksi saham.

Penelitian dilakukan dengan tahap-tahap sebagai berikut :

1. Mempelajari Studi literature yang terkait dengan masalah saham, jaringan saraf tiruan *Backpropagation* dan metode *Al-Alaoui Backpropagation*.
2. Melakukan pengumpulan data – data Saham bank BCA finance.
3. Menganalisis sistem dari data saham dan melakukan perancangan sistem yang meliputi pelatihan, dan pengujian.
4. Mengimplementasikan sistem untuk membuat perangkat lunak berdasarkan analisis dan perancangan yang telah dilakukan dengan menggunakan bahasa pemrograman Java.
5. Melakukan uji coba terhadap perangkat lunak.
6. Mengevaluasi uji coba terhadap perangkat lunak.

Langkah-langkah dapat dilihat pada gambar 3.1.



Gambar 3.1 Langkah-langkah penelitian

3.1 Studi Literatur

Studi Literatur yang digunakan berasal dari buku-buku dan jurnal baik jurnal nasional maupun jurnal internasional.

3.2 Data Penelitian

Data yang digunakan dalam sistem prediksi harga saham diambil dari situs yahoo finance dan menggunakan data bank Bank Central Asia dari agustus tahun 2011 sampai dengan akhir 2012.

3.3 Analisa dan Perancangan Sistem

3.3.1 Deskripsi umum sistem

Sistem prediksi saham dengan menggunakan algoritma *Al-Alaoui Backpropagation* merupakan sistem yang dapat memprediksi harga saham dikemudian hari dengan akurasi yang tinggi.

Sistem ini membaca semua data saham untuk dilakukan pelatihan jaringan saraf tiruan sehingga mendapatkan bobot yang digunakan untuk prediksi saham kedepan. Pelatihan yang digunakan dalam jaringan saraf tiruan ini ialah *Al-Alaoui Backpropagation*.

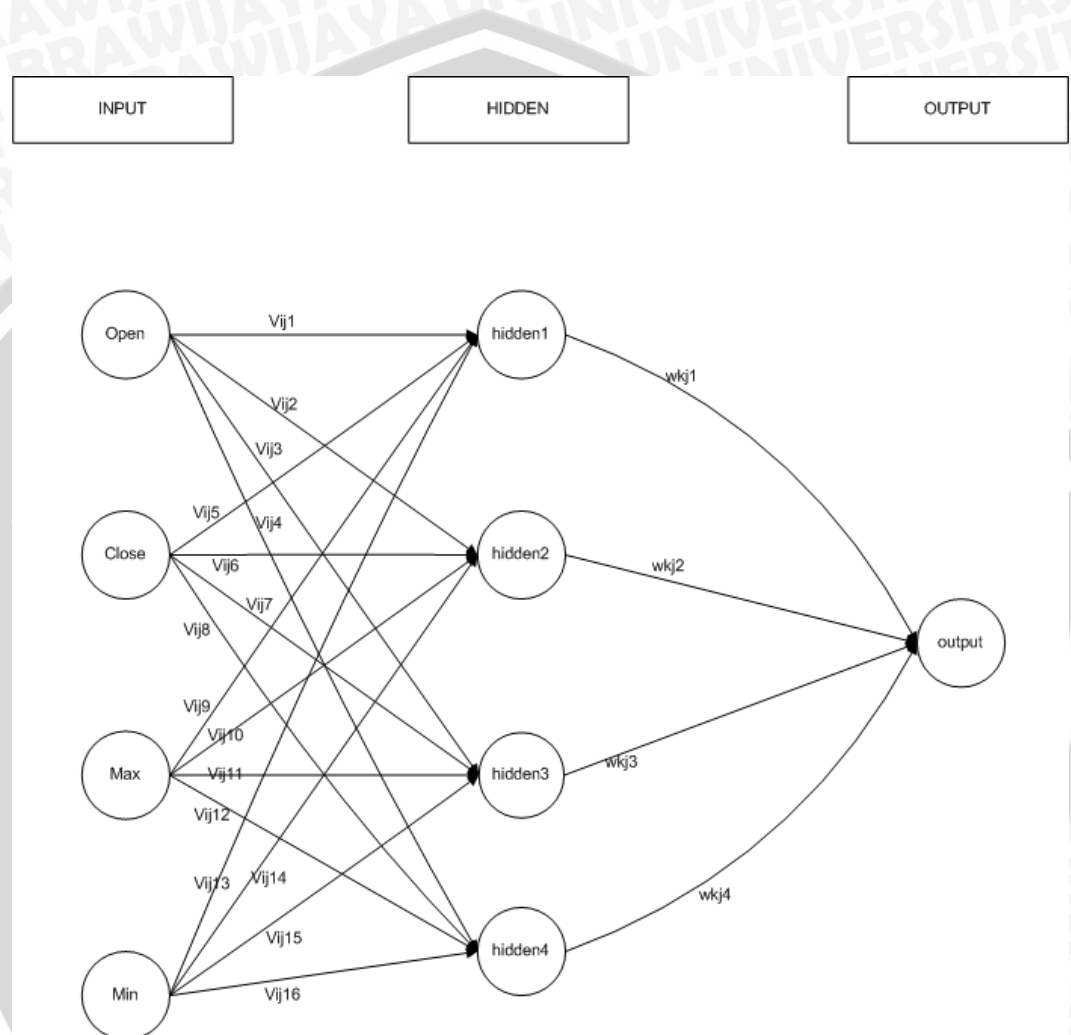
3.3.1.1 Batasan Sistem

Batasan sistem prediksi saham ini antara lain

- Input sistem berupa data saham BCA yahoo finance dari Agustus tahun 2011 sampai dengan akhir tahun 2012
- Prediksi saham hanya didasarkan dari faktor teknis tidak dari faktor external.
- Data yang diprediksi adalah data *open* di hari berikutnya.

3.3.2 Perancangan proses

Arsitektur Jaringan Syaraf Tiruan yang akan digunakan ditunjukkan pada gambar 3.2

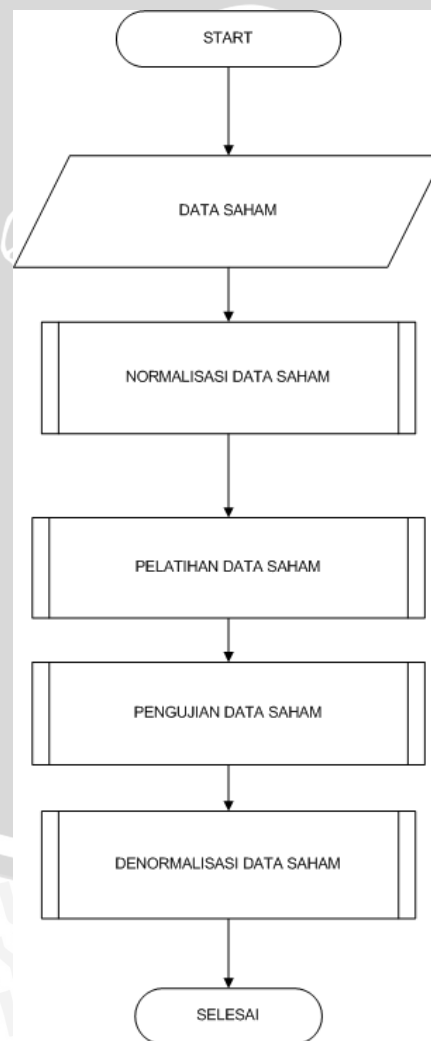


Gambar 3.2 Arsitektur Jaringan Pada Sistem

Arsitektur yang digunakan dalam sistem ini terdiri dari 3 lapisan yaitu lapisan *input*, lapisan *hidden*, serta lapisan *output*. Pada input layer menggunakan neuron untuk menampung 4 parameter input sesuai dengan parameter dari saham yang berjumlah 4. Untuk hidden layer didefinisikan sejumlah 4 diambil dari 2/3 input ditambah output. Sedangkan pada lapisan *output* jumlah *neuron* hanya 1 yaitu harga open saham yang diprediksi nantinya.

Untuk setiap data akan di *feedforward* ke dalam JST. *Output* akan dibandingkan dengan target, apabila selesih *output* dengan target melebihi galat yang ditetapkan, maka akan dilakukan *backpropagation* untuk memperbaiki bobot jaringan. Setelah itu data yang misklasifikasi akan diduplikasikan dan ditambah untuk proses pelatihan di *epoch* selanjutnya.

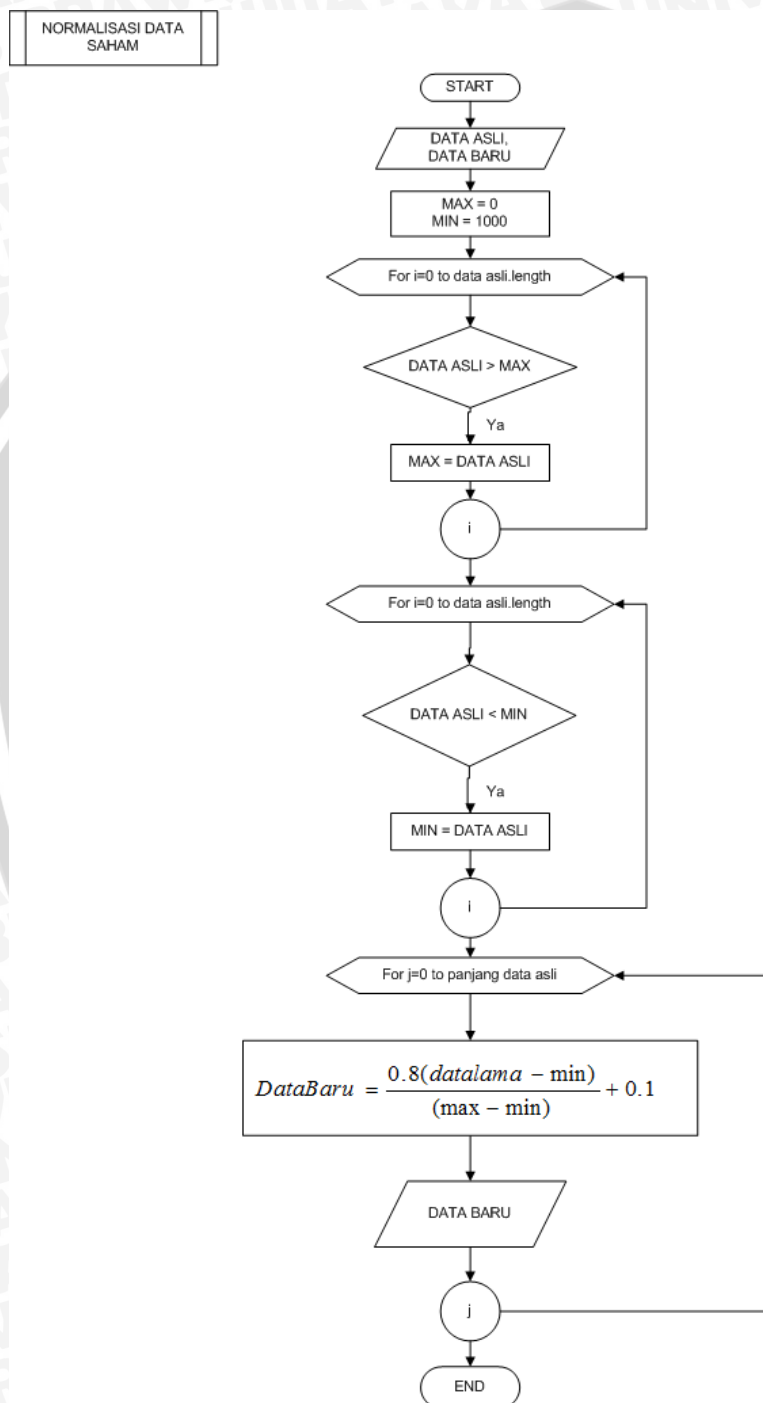
Pelatihan ini akan berhenti ketika tidak ada data yang mengalami misklasifikasi dan minimum *error* tercapai atau telah mencapai *epoch* maksimum. Setelah itu bobot terakhir disimpan untuk dilakukan pengujian. Pengujian dilakukan untuk mendapatkan harga saham di hari yang ditentukan. Alur proses deskripsi sistem secara umum ditunjukkan pada gambar 3.3.



Gambar 3.3 Deskripsi sistem secara umum

3.3.2.1 Normalisasi Data Saham

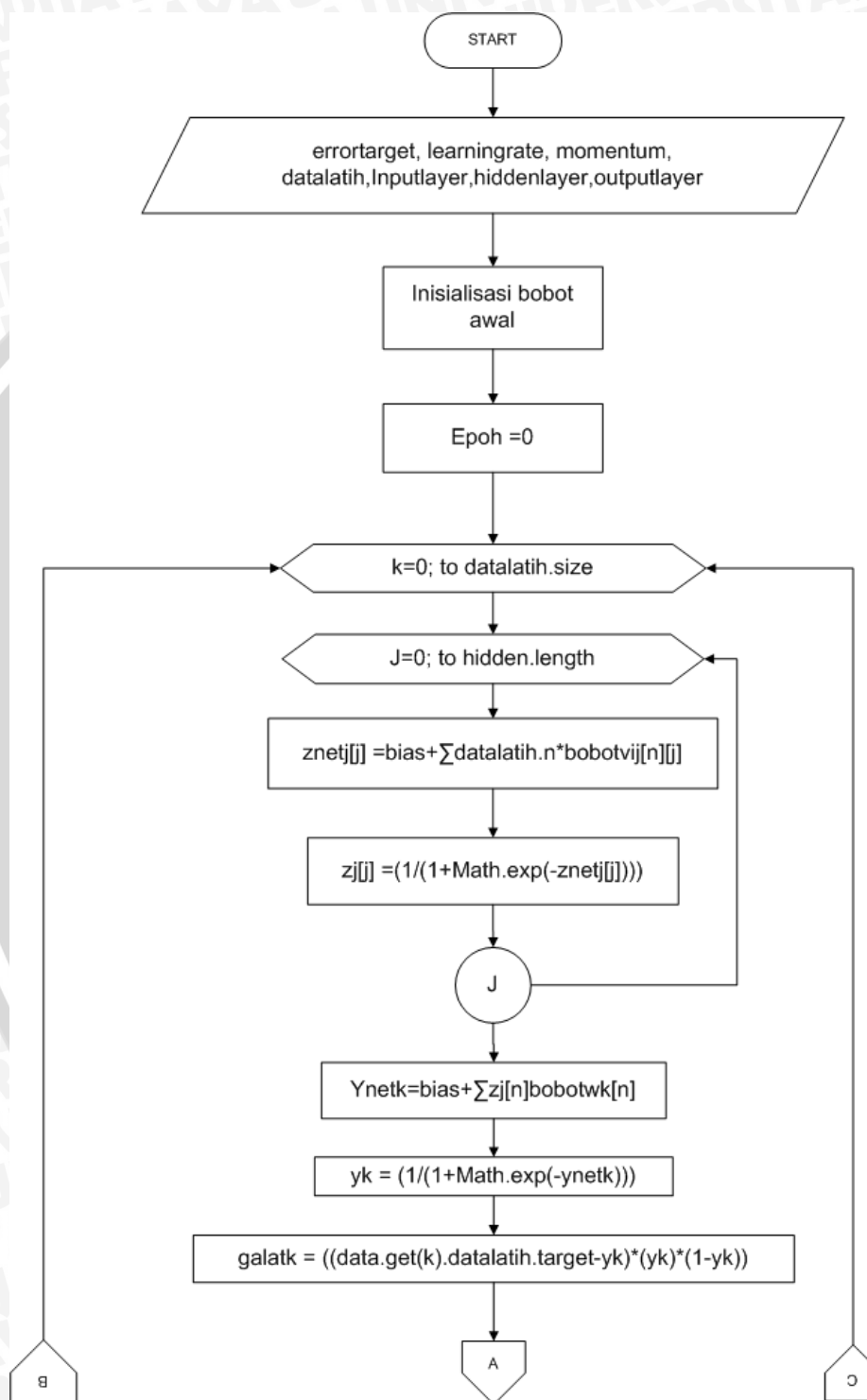
Dalam proses pelatihan digunakan rumus sigmoid biner yang mengharuskan data bernilai antara 0 dan 1, maka diperlukan normalisasi. Alur proses normalisasi ditunjukkan pada gambar 3.4



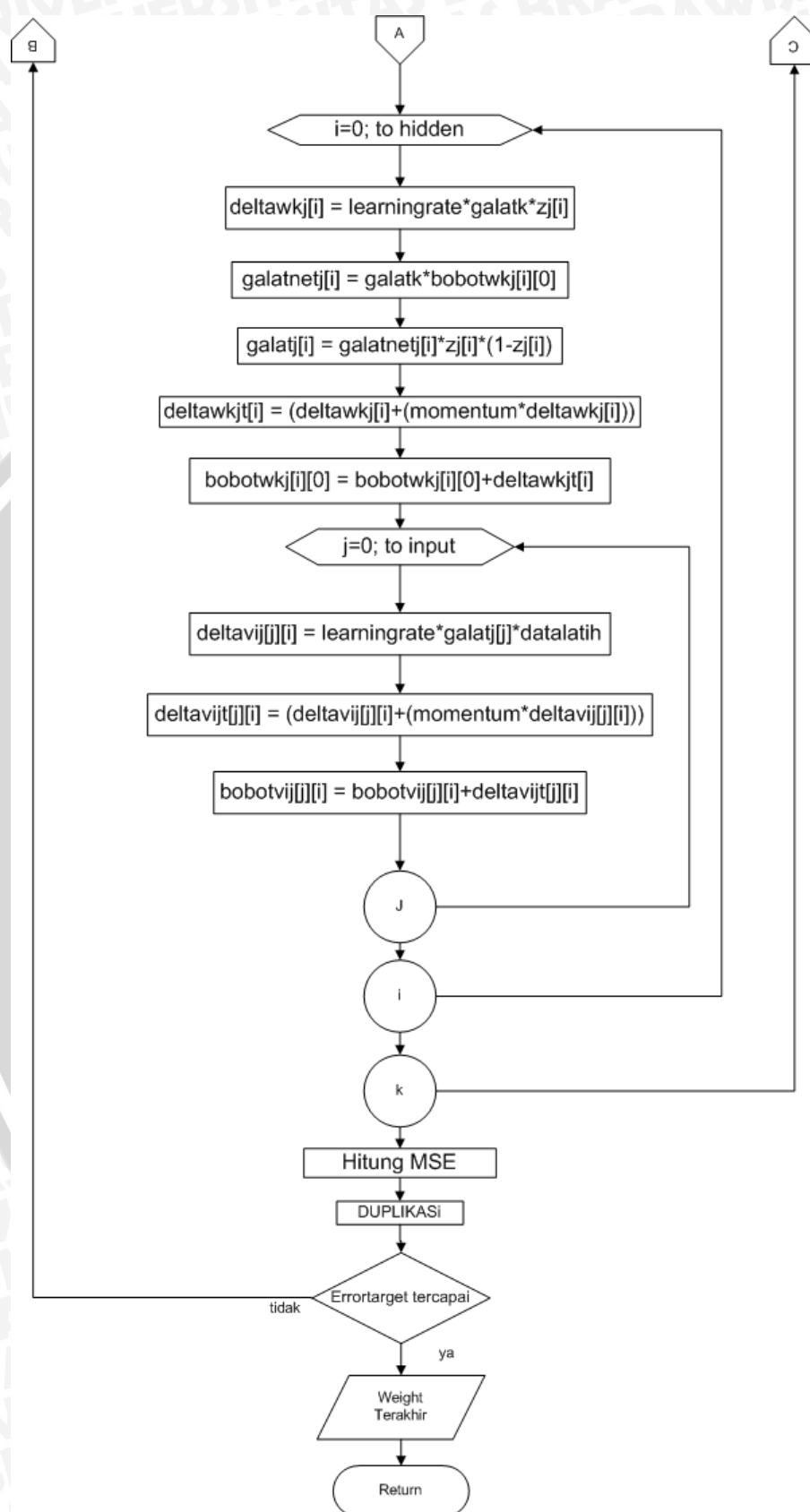
Gambar 3.4 Proses Normalisasi Data

3.3.2.2 Pelatihan data Saham

Flowchart dari proses pelatihan data saham dapat dilihat pada gambar 3.5, 3.6.



Gambar 3.5 Proses *FeedForward JST Al-Alaoui backpropagation*



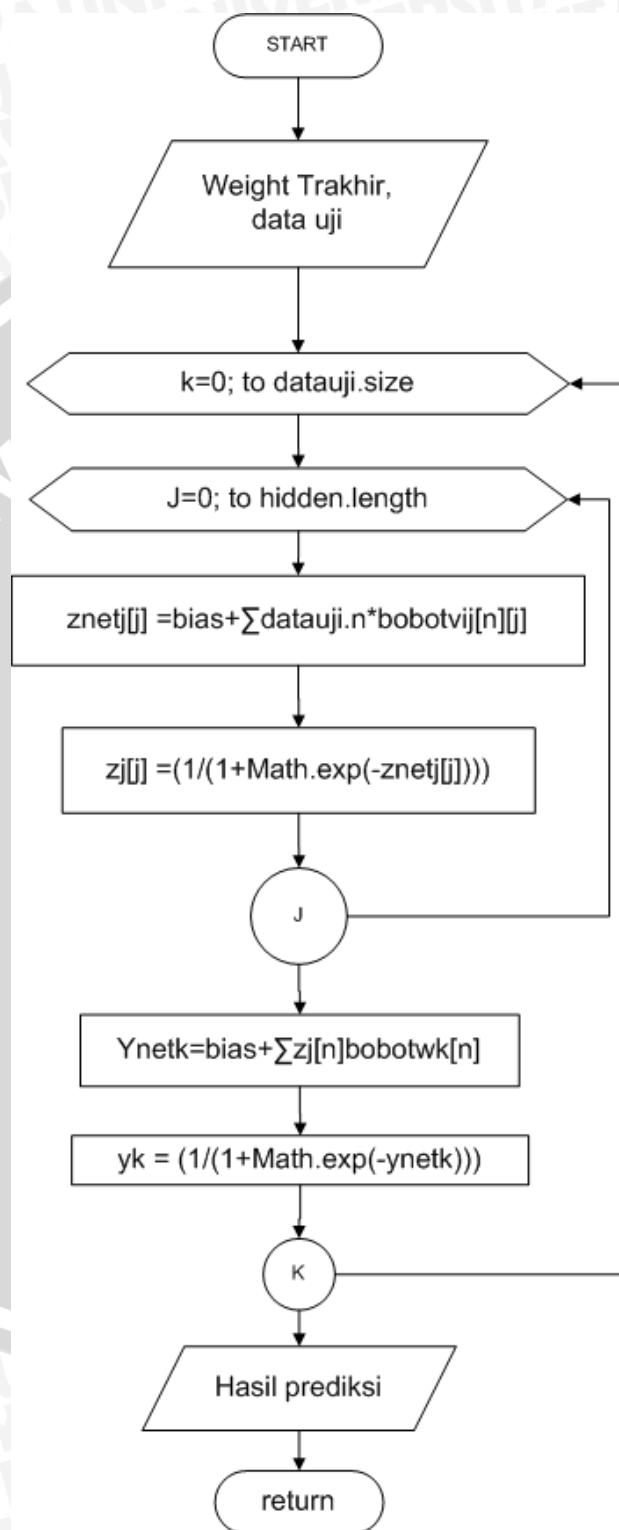
Gambar 3.6 Proses Backpropagation JST Al-Alaoui backpropagation

Proses pelatihan Jaringan Syaraf Tiruan memiliki beberapa langkah, yaitu :

1. Input data latih untuk pelatihan
2. inialisasi nilai learning rate, nilai momentum, epohmax dan errortarget, jumlah *neuron*, jumlah *hidden*, jumlah *output*.
3. Inialisasi bobot awal secara random antara -0.5-0.5 atau bilangan kecil lain
4. Jika nilai *MSE* lebih besar dari errortarget, maka dilakukan langkah ke berikutnya . Jika tidak proses pelatihan selesai.
5. Dilakukan perhitungan *feed forward* kedalam JST.
6. Menghitung nilai error menggunakan persamaan 2.5
7. Membandingkan nilai error dengan galat maksimal yang telah ditetapkan. Jika nilai error lebih besar dari galat maksimal, maka dilakukan langkah selanjutnya.
8. Melakukan perhitungan backpropagation untuk menghitung nilai perbaikan setiap bobot.
9. Duplikasi data digunakan untuk digunakan iterasi selanjutnya.
10. Jika seluruh data telah diproses maka dilakukan langkah selanjutnya. Jika belum kembali ke langkah 5.
11. Nilai iterasi ditambahkan.
12. Kembali ke langkah 4 dengan membandingkan nilai *MSE*.

3.3.2.3 Pengujian Data Saham

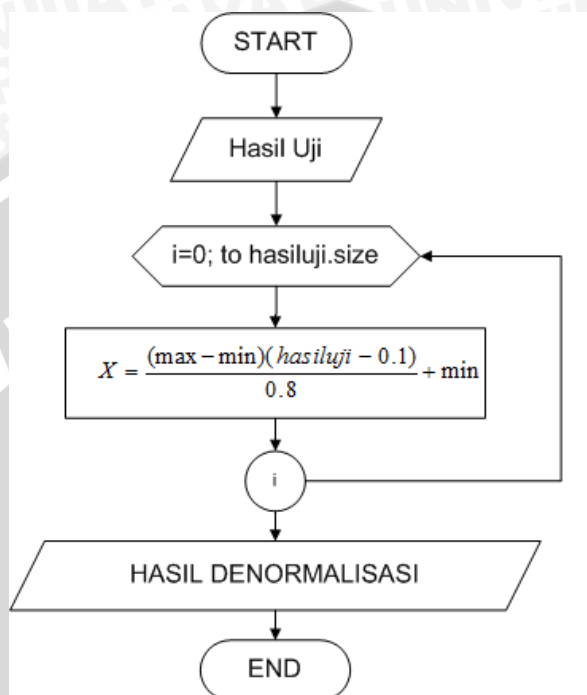
Proses pengujian data saham ditunjukkan pada gambar 3.6.



Gambar 3.7 Proses Pengujian JST Al-Aloui Backpropagation

3.3.2.4 Proses Denormalisasi Data

Nilai prediksi masih dalam nilai normalisasi untuk mengembalikan ke nilai normal diperlukan proses denormalisasi. Proses denormalisasi ini ditunjukkan pada gambar 3.7.



Gambar 3.8 Proses Denormaliasi Data

3.4 Perhitungan Manual

3.4.1 Proses penginputan data

Data yang dipakai adalah data saham pata tanggal 4-10 Agustus 2011

Tabel 3.1 Data asli saham

Date	Open	High	Low	Close	Target	Kategori
4-Aug-11	21.23	21.39	20.1	20.4	20.7	Turun
5-Aug-11	20.7	21.8	19.36	20.3	19.72	Turun
8-Aug-11	19.72	19.72	18.45	18.55	18.07	Turun
9-Aug-11	18.07	20.05	17.32	19.28	19.13	Naik
10-Aug-11	19.13	20.27	18.84	20.04	20.02	Naik

Tabel 3.2 Data setelah dinormalisasi

Open	High	Low	Close	Target open
0.9	0.742307692	0.9	0.9	0.9
0.765822785	0.9	0.6870504	0.8567568	0.60190114
0.517721519	0.1	0.4251799	0.1	0.1
0.1	0.226923077	0.1	0.4156757	0.42243346
0.36835443	0.311538462	0.5374101	0.7443243	0.69315589

Struktur Jaringan Syaraf Tiruan akan ditentukan untuk pelatihan sebagai berikut :

- Jumlah input = 4
- Jumlah hidden layer = 4
- Jumlah output = 1
- α (learning rate) = 0.2
- η (momentum) = 0.8
- δ (galat) = 0.01

3.4.2 Inisialisasi Bobot awal

inisialisasi bobot awal random pada tabel 3.3 dan 3.4:

Tabel 3.3 Tabel Data Nilai bobot Awal v_{ij}

i/j	1	2	3	4
1	0.299799994	0.9545391	0.7049095	0.8218871
2	0.903682006	0.1219987	0.6180878	0.66832211
3	0.829791701	0.6875152	0.1968769	0.15472748
4	0.434756064	0.684146	0.5250994	0.08713548

Tabel 3.4 Tabel Data Nilai bobot Awal w_{jk}

k/j	1	2	3	4
1	0.971370402	0.421753	0.4979287	0.00309462

3.4.3 Proses FeedForward

Proses feedforward bertujuan untuk mendapatkan nilai sebuah neuron pada lapisan berikutnya. Semua keluaran unit tersembunyi dihitung Z_j pada persamaan 2.2 dan fungsi aktivasi persamaan 2.1. Berikut ini contoh perhitungan Data ke-x1 yang ditunjukkan pada pada 3.5 dan tabel 3.6.

$$z_net_j = \sum_{i=1}^n x_i \cdot v_{ji}$$

Tabel 3.5 Tabel Data Znetj

Znetj	J			
	1	2	3	4
	3.078723087	3.1841408	2.7430086	2.45347569

$$z_j = f(z_net_j) = \frac{1}{1 + \exp(-z_net_j)}$$

Tabel 3.6 Tabel Data Zj

Zj	J			
	1	2	3	4
	0.956006511	0.9602331	0.9395173	0.92081525

Setelah mendapatkan nilai z_j cari nilai y_net dengan persamaan 2.3

$$y_net_k = w_{k0} + \sum_{j=1}^p z_j \cdot w_{kj}$$

y_net	K
	2.80427985

Dari y_net diatas dihitung y_k (output) dengan persamaan 2.4

$$y_k = f(y_net_k) = \frac{1}{1 + \exp(-y_net_k)}$$

yk	k
	0.942906661

Setelah didapatkan outputnya dilanjutkan proses *backpropagation*

3.4.4 Proses Backpropagation

Berdasarkan persamaan 2.6 dapat dihitung nilai aktivasi output dari nilai δ_k masing-masing neuron output data ke-1. δ_k merupakan unit kesalahan yang dipakai dalam perubahan bobot layar δ_{net_j}

$$\begin{aligned}\delta_k &= (t_k - y_k) y_k (1 - y_k) \\ &= (0.9 - 0.942906661) 0.942906661 (1 - 0.942906661) \\ &= -0.002309824\end{aligned}$$

Menghitung perubahan bobot w_{kj} dengan percepatan α dengan persamaan 2.7, hasilnya akan ditunjukkan pada tabel 3.7.

$$\begin{aligned}\Delta w_{kj} &= \alpha \delta_k z_j \\ &= 0.2 \times (-0.002309824) \times 0.956006511 \\ &= -0.0253578\end{aligned}$$

Tabel 3.7 Tabel Data Δw_{kj}

K	J			
	1	2	3	4
	-0.001104103	-0.001109	-0.0010851	-0.0010635

Menghitung faktor δ_{layer} tersembunyi data ke-x1 berdasarkan kesalahan pada setiap layer tersembunyi z_j menggunakan persamaan 2.8. Hasilnya ditunjukkan pada tabel 3.8.

$$\delta_{net_j} = \sum_{k=1}^m \delta_k \cdot w_{kj}$$

Tabel 3.8 Tabel Data δ_{net_j}

$-\delta_{\text{net}_j}$	J			
	1	2	3	4
	-0.002243695	-0.0009742	-0.0011501	-7.148E-06

Dihitung juga faktor kesalahan δ_j di layer tersembunyi dengan persamaan 2.9 dan hasilnya diperoleh pada tabel 3.9

$$\delta_j = \delta_{\text{net}_j} \cdot z_j(1 - z_j)$$

Tabel 3.9 Tabel Data δ_j

$-\delta_j$	J			
	1	2	3	4
	-9.43654E-05	-3.72E-05	-6.536E-05	-5.212E-07

Setelah didapatkan nilai galat tiap-tiap layer, maka dihitung perubahan bobot dari v_{ji} data ke- 1 dari persamaan 2.10 dan hasilnya diperoleh pada tabel 3.10

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

Tabel 3.10 Tabel Data Δv_{ji}

I	J			
	1	2	3	4
1	-1.69858E-05	-6.7E-06	-1.18E-05	-9.38E-08
2	-1.40096E-05	-5.52E-06	-9.7E-06	-7.74E-08
3	-1.69858E-05	-6.7E-06	-1.18E-05	-9.38E-08
4	-1.69858E-05	-6.7E-06	-1.18E-05	-9.38E-08

3.4.5 Proses Weight Update

Proses weight update digunakan untuk menghitung perubahan bobot baris yang menuju unit keluaran dan lapisan tersembunyi pada persamaan 2.11 dan 2.12. Sebelum menghitung persamaan di atas dihitung nilai Δw_{kj} dan Δv_{ji} . Hasilnya dapat dilihat dalam tabel 3.11, tabel 3.12, tabel 3.13 dan tabel 3.14

$$\Delta w_{kj} = (\alpha \delta_k z_j) + (\eta \Delta w_{kj})$$

Tabel 3.11 Tabel Data $\Delta w_{kj}(t)$

K	J			
	1	2	3	4
	-0.001656155	-0.001663	-0.001628	-0.001595

$$\Delta v_{ji} = (\alpha \delta_j x_i) + (\eta \Delta v_{ji})$$

Tabel 3.12 Tabel Data $\Delta v_{ji}(t)$

I	J			
	1	2	3	4
1	-3.05744E-05	-1.21E-05	-2.12E-05	-1.69E-07
2	-2.52174E-05	-9.94E-06	-1.75E-05	-1.39E-07
3	-3.05744E-05	-1.21E-05	-2.12E-05	-1.69E-07
4	-3.05744E-05	-1.21E-05	-2.12E-05	-1.69E-07

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

Tabel 3.13 Tabel Data $\Delta w_{kj}(\text{new})$

K	J			
	1	2	3	4
	0.969714247	0.42009	0.496301	0.001499

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

Tabel 3.14 Tabel Data $\Delta v_{ji}(\text{new})$

i	J			
	1	2	3	4
1	0.29976942	0.954527	0.704888	0.821887
2	0.903656788	0.121989	0.61807	0.668322
3	0.829761127	0.687503	0.196856	0.154727
4	0.434725489	0.684134	0.525078	0.087135

3.4.6 Al-Alaoui

Setelah dilakukan backpropagation sampai seluruh data, dilihat misklasifikasi yang ada pada pada setiap data

Tabel 3.15 Tabel data setelah backpropagation

No	Open	High	Low	Close	Target open	klass	klas hasil
1	21.23	21.39	20.1	20.4	20.7	Turun	Naik
2	20.7	21.8	19.36	20.3	19.72	Turun	Naik
3	19.72	19.72	18.45	18.55	18.07	Turun	Naik
4	18.07	20.05	17.32	19.28	19.13	Naik	Naik
5	19.13	20.27	18.84	20.04	20.02	Naik	Naik

Dari hasil diatas ada 3 kelas yang salah, sehingga 3 data tersebut di duplikasi

Tabel 3.16 Tabel data duplikasi

No	Open	High	Low	Close	Target open	klass
1	21.23	21.39	20.1	20.4	20.7	Turun
2	20.7	21.8	19.36	20.3	19.72	Turun
3	19.72	19.72	18.45	18.55	18.07	Turun
4	18.07	20.05	17.32	19.28	19.13	Naik
5	19.13	20.27	18.84	20.04	20.02	Naik
6	21.23	21.39	20.1	20.4	20.7	Turun
7	20.7	21.8	19.36	20.3	19.72	Turun
8	19.72	19.72	18.45	18.55	18.07	Turun

Data setelah dilakukan duplikasi, diiterasi sampai errortarget yang diinginkan tercapai. Setelah mencapai erortarget yang diinginkan bobot terakhir yang didapatkan sudah dapat digunakan untuk melakukan pengujian, berikut ini hasil akhir pelatihan

Tabel 3.17 Data hasil Pelatihan

Iterasi	MSE	Total Data
0	0.2219262	5
1	0.2564486	8
2	0.239279	11
3	0.1863338	15
4	0.1336195	20
5	0.1008455	29
6	0.0894281	38
7	0.083476	47
8	0.0795627	56
9	0.0767471	65
10	0.0746158	74
11	0.0729412	83
12	0.0715858	92
13	0.0704608	101
14	0.0695063	110
15	0.06868	119
16	0.0679507	128
17	0.0672944	137
18	0.0666919	146
19	0.0661268	155
20	0.0655841	164
21	0.0650492	173
22	0.0645061	182
23	0.0639363	191
24	0.0633164	200

iterasi	MSE	Total Data
25	0.062614	209
26	0.061782	218
27	0.0607471	227
28	0.0593883	236
29	0.057496	245
30	0.0546985	254
31	0.0503846	263
32	0.0439347	272
33	0.0363075	303
34	0.0264471	344
35	0.019674	385
36	0.0160133	426
37	0.0140616	467
38	0.0128696	508
39	0.0120309	549
40	0.011372	590
41	0.0108167	631
42	0.0103317	672
43	0.0099038	713

Dari pelatihan diatas didapatkan bobot akhir sebagai berikut

Tabel 3.18 Bobot Akhir Vij

i/j	1	2	3	4
1	-0.54301347	1.0277726	0.235929254	1.203703
2	1.45358992	0.3848761	0.031187686	0.21116
3	0.25027952	0.3649408	0.036608244	0.185967
4	0.77394581	-3.876523	1.443474727	-4.30447

Tabel 3.19 Bobot Akhir Wkj

j/k	1
1	0.78642856
2	-0.92138622
3	0.27458288
4	-4.21453352

3.4.7 Pengujian

Bobot akhir dari pelatihan akan digunakan untuk pengujian, berikut ini data setelah akan diuji

Tabel 3.20 Data Pengujian

Tanggal	Target Awal	Target hasil	Kelas Awal	Kelas Akhir
4-Aug-11	20.7	20.102212	Turun	Turun
5-Aug-11	19.72	20.098679	Turun	Turun
8-Aug-11	18.07	18.057719	Turun	Turun
9-Aug-11	19.13	19.403307	Naik	Naik
10-Aug-11	20.02	20.151543	Naik	Naik

3.5 Perancangan Antarmuka

Rancangan antarmuka pada sistem ini terdiri dari 3 bagian, yaitu load data untuk memanggil data latih, pelatihan untuk melatih data latih, dan pengujian untuk memanggil dan menguji data uji. Perancangan antarmuka sistem akan ditunjukkan pada gambar berikut ini.

Load Data

Pelatihan

Pengujian

Load data

Data Awal

Index	Tanggal	Open	High	Low	Close	Target	Kategori

Data Normalisasi

Index	Tanggal	Open	High	Low	Close	Target	Kategori

Gambar 3.9 Load data

1. Tombol Load data digunakan untuk mencari data latih yang ingin digunakan.
2. Tabel Data Awal digunakan untuk menampilkan data latih yang dipilih saat menekan tombol Load.
3. Tabel Data Normalisasi digunakan untuk menampilkan hasil normalisasi dari data latih yang dipilih.

Load Data

Pelatihan

Pengujian

Learning Rate

0.5

Momentum

0.5

Mulai

Tabel Proses Pelatihan

Epoh	MSE	TotalData

Hasil Pelatihan

H1	H2	H3	H4	Output

Gambar 3.10 Pelatihan

1. *Textfield* Learning rate digunakan untuk set data learning rate yang nanti akan digunakan pada perhitungan sistem.
2. *Textfield* Momentum digunakan untuk set data Momentum yang nanti akan digunakan pada perhitungan sistem.
3. Tabel proses pelatihan digunakan untuk menampilkan hasil dari proses pelatihan dimana menampung nilai epoh, mse, total data tiap iterasi dalam pelatihan.
4. Tabel hasil pelatihan digunakan untuk menampilkan hasil bobot akhir deltavij dari proses pelatihan.
5. Tabel output digunakan untuk menampilkan hasil bobot akhir deltawkj dari proses pelatihan.

Load Data

Pelatihan

Pengujian

Load Data Uji

Mulai Pengujian

Data Uji

Index	Tanggal	Open	High	Low	Close	Target	Kategori

Hasil Pengujian

Index	Tanggal	Target Awal	Hasil Uji	Trend Awal	Trend Akhir

AKURASI =

MSE =

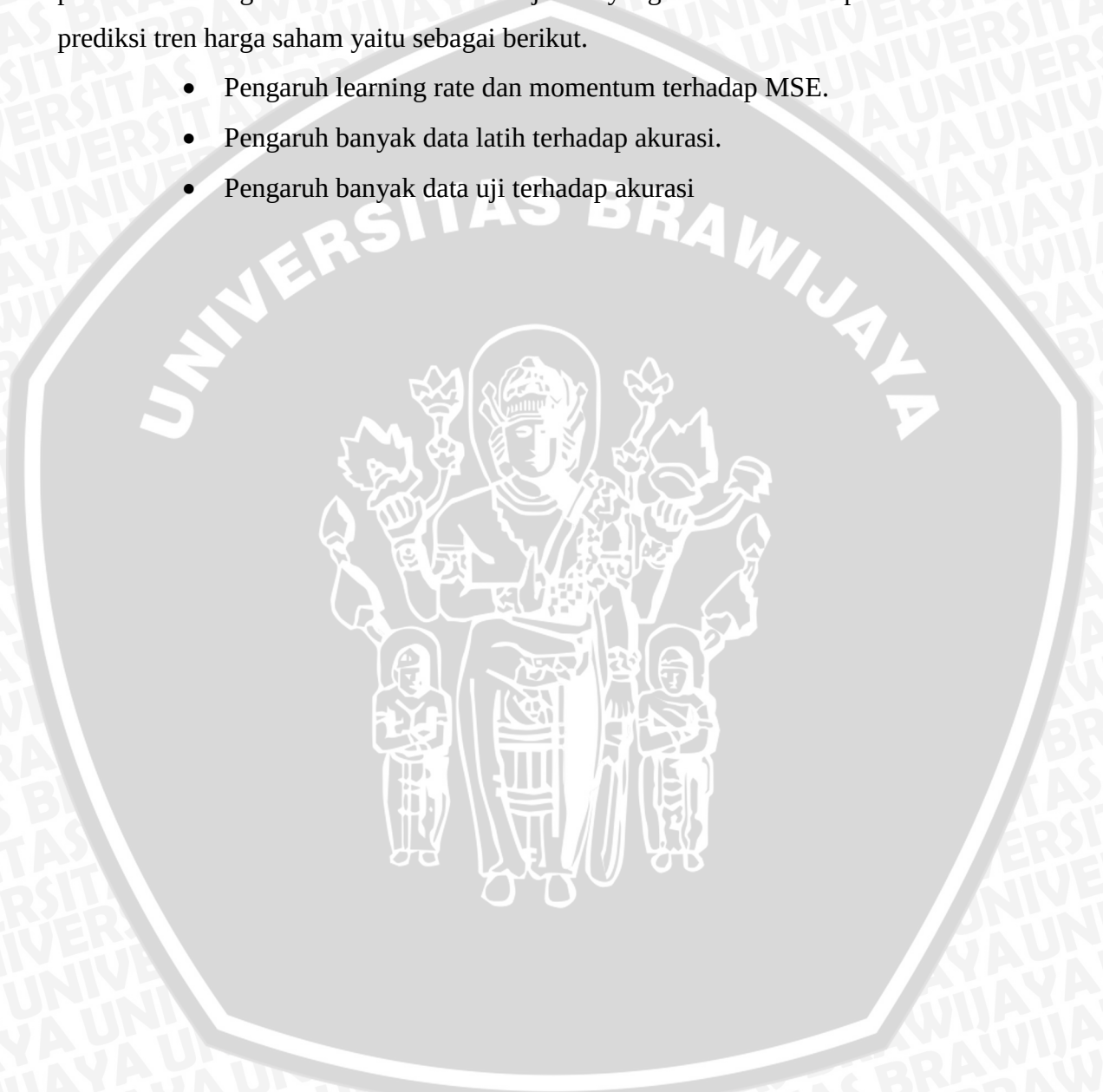
Gambar 3.11 Pengujian

1. Tombol Load Data Uji digunakan untuk mencari data uji yang ingin digunakan dalam sistem prediksi tren harga saham.
2. Tombol mulai pengujian digunakan untuk memulai pengujian terhadap data uji yang dicari saat menekan load data uji.
3. Tabel data uji digunakan untuk menampilkan nilai data uji.
4. Tabel hasil pengujian digunakan untuk menampilkan nilai hasil pengujian yang meliputi index, tanggal, target awal hasil uji, trend awal, dan trend akhir.
5. *Textfield* akurasi digunakan untuk menampilkan akurasi hasil pengujian.
6. *Textfield* MSE digunakan untuk menampilkan jumlah MSE dari pengujian.

3.6 Perancangan Ujicoba

Setelah sistem selesai dibuat, maka dilanjutkan dengan proses ujicoba, proses ujicoba ini digunakan untuk mengetahui tingkat akurasi dari sistem prediksi tren harga saham. Ada 3 macam ujicoba yang akan dilakukan pada sistem prediksi tren harga saham yaitu sebagai berikut.

- Pengaruh learning rate dan momentum terhadap MSE.
- Pengaruh banyak data latih terhadap akurasi.
- Pengaruh banyak data uji terhadap akurasi



BAB IV

IMPLEMENTASI

4.1 Lingkungan Implementasi

Dalam implementasi metode Algoritma *Al-Alaoui backpropagation* untuk meramalkan tren harga saham dibutuhkan beberapa aspek yang perlu diperhatikan yaitu segi perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Implementasi Perangkat Keras

Dalam mengembangkan sistem dan penerapan metode penelitian ini digunakan beberapa komponen perangkat keras sebagai berikut :

1. Processor : Intel ® Core™ 2Duo CPU @2.10GHz
2. Memory : 2.00 GB
3. Hard disk : 300 GB

4.1.2 Lingkungan Implementasi Perangkat Lunak

Pengembangan sistem dan penelitian ini dibutuhkan beberapa perangkat lunak yang digunakan sebagai berikut ini :

1. Sistem Operasi yang digunakan Windows 7 Ultimate 32bit
2. Aplikasi pembangunan GUI dan code menggunakan NetBeans IDE 7.3.1
3. Bahasa pemrograman yang dipakai yaitu bahasa pemrograman java.
4. Komponen java yang digunakan yaitu JDK 1.7

4.2 Implementasi Program

Berdasarkan metode penelitian dan perancangan proses yang terdapat dalam bab 3, maka pada sub bab ini akan dijelaskan implementasi proses-proses tersebut.

4.2.1 Proses Baca File

Proses baca file digunakan untuk membaca file saham yang akan digunakan dalam pelatihan dan pengujian. Tahapan proses baca file ditunjukkan pada source code 4.1

Proses Baca File

```
package alaloui;
import jxl.Workbook;
import jxl.Sheet;
import java.util.ArrayList;
import java.io.File;
import java.io.IOException;
public class bacadata
{
    //untuk menampung data dari excel disimpan di
    arraylist klas atribut
    ArrayList <Atribut> dataset = new ArrayList
    <Atribut>();
    //untuk menyimpan path dari data
    String filename;
    jst jajal = new jst();
    public bacadata(String filename)
    {
        this.filename = filename;
    }
    public ArrayList <Atribut> baca()
    {
        //mengambil data dari path data
        File file = new File(filename);
        String cell = "";
        try{Workbook workbook =
        jxl.Workbook.getWorkbook(file);
        Sheet[] sheets = workbook.getSheets();
        int row = sheets[0].getRows();
        int column = sheets[0].getColumns();
```

```
dataset.clear();
for (int i=0; i<row-1;i++)
{ dataset.add(new Atribut
(0,"",0,0,0,0,0,""));
}for (int i=1; i<row;i++){
String tanggal= sheets[0].getCell(1,
i).getContents();
double open= Double.parseDouble(sheets[0].getCell(2,
i).getContents());
double high= Double.parseDouble(sheets[0].getCell(3,
i).getContents());
double low= Double.parseDouble(sheets[0].getCell(4,
i).getContents());
double close= Double.parseDouble(sheets[0].getCell(5,
i).getContents());
double target=Double.parseDouble(sheets[0].getCell(6,
i).getContents());
String kategori= sheets[0].getCell(7,
i).getContents();dataset.set(i-1, new Atribut
((i),tanggal,open,high,low,close,target,kategori));
}}catch (Exception ex){System.out.println("Gagal
membaca XLS!"+" \nAda Error: "+ex.getMessage());}
return dataset;}}
```

Source Code 4.1 Baca File

4.2.2 Proses Normalisasi Data

Proses normalisasi data digunakan untuk normalisasi data latih atau data uji untuk bias digunakan pada system ketika melakukan perhitungan. Source code normalisasi data ditunjukkan pada source code 4.2

Proses Normalisasi data

```
package alaloui;

import java.util.ArrayList;

public class normalisasi
{
    ArrayList <Atribut> dataxxx;    double
    maxopen,minopen,maxhigh,minhigh,maxlow,minlow,maxclose,
    minclose,maxtarget,mintarget;

    double hasil;
    ArrayList<Double> databaru;

    public normalisasi(ArrayList <Atribut> datax)
    {
        this.dataxxx = datax;
        minopen = dataxxx.get(0).open;
        minhigh = dataxxx.get(0).high;
        minlow = dataxxx.get(0).low;
        minclose = dataxxx.get(0).close;
        mintarget = dataxxx.get(0).target;
    }

    public void maxmin()
    {
        for (int i=0;i<dataxxx.size();i++)
        {
            if (dataxxx.get(i).open>maxopen)
            {
                maxopen = dataxxx.get(i).open;
            }
            if (dataxxx.get(i).high>maxhigh)
            {
                maxhigh = dataxxx.get(i).high;
            }
        }
    }
}
```

```
if (dataxxx.get(i).low>maxlow)
{
    maxlow = dataxxx.get(i).low;
}
if (dataxxx.get(i).close>maxclose)
{
    maxclose = dataxxx.get(i).close;
}
if (dataxxx.get(i).target>maxtarget)
{
    maxtarget = dataxxx.get(i).target;
}
}
for (int i=0;i<dataxxx.size();i++)
{
    if (dataxxx.get(i).open<=minopen)
    {
        minopen = dataxxx.get(i).open;
    }
    if (dataxxx.get(i).high<=minhigh)
    {
        minhigh = dataxxx.get(i).high;
    }
    if (dataxxx.get(i).low<=minlow)
    {
        minlow = dataxxx.get(i).low;
    }
    if (dataxxx.get(i).close<=minclose)
    {
        minclose = dataxxx.get(i).close;
    }
    if (dataxxx.get(i).target<=mintarget)
    {
        mintarget = dataxxx.get(i).target;
    }
}
```

```

    }

    public ArrayList<Atribut> normal()
    {
        double open,high,low,close,target;
        for (int i=0; i<dataxxx.size(); i++)
        {
            open =
            (0.8*((dataxxx.get(i).open-minopen)/(maxopen-
            minopen)))+0.1;

            high = (0.8*((dataxxx.get(i).high-
            minhigh)/(maxhigh-minhigh)))+0.1;

            low = (0.8*((dataxxx.get(i).low-
            minlow)/(maxlow-minlow)))+0.1;

            close = (0.8*((dataxxx.get(i).close-
            minclose)/(maxclose-minclose)))+0.1;

            target = (0.8*((dataxxx.get(i).target-
            mintarget)/(maxtarget-mintarget)))+0.1;

            dataxxx.set(i, new Atribut
            (dataxxx.get(i).index,dataxxx.get(i)
            .tanggal,open,high,low,close,target,dataxxx.get(i).ka
            tegori));
        }
        return dataxxx;
    }

    public ArrayList<Double> denormalisasi
    (ArrayList<Double> dataq)
    {
        this.databaru = dataq;
        for (int i=0; i<dataq.size();i++)
        {
            hasil = mintarget+((maxtarget-
            mintarget)*((dataq.get(i)-0.1)/0.8));
            databaru.set(i, hasil);
        }
        return databaru;
    }
}

```

Source kode 4.2 Normalisasi Data

4.2.3 Proses Pelatihan

Proses pelatihan digunakan untuk melakukan *feedforward*, *backpropagation*, *duplikasi* data, untuk mendapatkan bobot akhir yang akan digunakan untuk pengujian nantinya. Source kode pelatihan ditunjukkan pada source kode 4.3 berikut

Proses Pelatihan

```
package alaloui;

import java.util.ArrayList;

public class pelatihan
{
    ArrayList <Atribut> data;
    ArrayList <Double> hasil = new ArrayList();
    ArrayList<Integer> khasil = new ArrayList();
    ArrayList<Integer> kelas = new ArrayList();
    ArrayList<Integer> temporari = new ArrayList();
    ArrayList<Double> MSEx = new ArrayList();
    ArrayList<Integer> totaldata= new ArrayList();
    double learningrate;
    double momentum;
    double [][] bobotvij,bobotwkj,deltavij,deltavijt;
    int input,hidden,output,kondisi,epoh;
    double[]
    znetj,zj,deltawkj,galatnetj,galatj,deltawkjt;
    double ynetk,yk,galatk,MSE,tempmse;
    double error = 0.001;
    public pelatihan(ArrayList <Atribut> a, double
    learningrate, double momentum)
    {
        this.data = a;
        this.learningrate = learningrate;
        this.momentum = momentum;
        input = 4;
        hidden = 4;
        output = 4;
```

```
for (int i=0; i<data.size();i++)
{
    if (data.get(i).open<data.get(i).target)
    {
        kelas.add(1);
    }
    else
    {
        kelas.add(0);
    }
    khasil.add(0);
}

public void setbobot()
{
    bobotvij = new double[input][hidden];
    bobotwkj = new double[hidden][output];
    bobotvij[0][0] = 0.29979999419025605;
    bobotvij[1][0] = 0.9036820056891591;
    bobotvij[2][0] = 0.8297917011928085;
    bobotvij[3][0] = 0.43475606374438736;
    bobotvij[0][1] = 0.9545391021454477;
    bobotvij[1][1] = 0.12199865835488044;
    bobotvij[2][1] = 0.6875151736089494;
    bobotvij[3][1] = 0.6841460294495141;
    bobotvij[0][2] = 0.7049094865204215;
    bobotvij[1][2] = 0.6180877946583554;
    bobotvij[2][2] = 0.19687693389449257;
    bobotvij[3][2] = 0.5250994020089463;

    bobotvij[0][3] = 0.8218870983345538;
    bobotvij[1][3] = 0.6683221110694832;
    bobotvij[2][3] = 0.15472747763631856;
    bobotvij[3][3] = 0.08713547835859392;
    bobotwkj[0][0] = 0.9713704017330818;
```

```
bobotwkj[1][0] = 0.4217530423870427;
bobotwkj[2][0] = 0.4979287020789054;
bobotwkj[3][0] = 0.0030946206296594836;
}
public void feedforward()
{
    znetj = new double[hidden];
    zj = new double[hidden];
    deltawkj = new double[hidden];
    galatnetj = new double[hidden];
    galatj = new double[hidden];
    deltawkjt = new double[hidden];
    deltavij=new double[input][hidden];
    deltavijt = new double[input][hidden];
    for(int i=0;i<data.size();i++)
    {
        hasil.add(0.0);
    }
    do
    {
        tempmse = MSE;
        for (int k=0; k< data.size();k++)
        {
            for (int j=0;j<hidden;j++)
            {
                znetj[j] =
                (1+(data.get(k).open*bobotvij[0][j]+data.get(k).high*b
                obotvij[1][j]
                +data.get(k).low*bobotvij[2][j]+data.get(k).close*bobo
                tvij[3][j]));
                zj[j] = (1/(1+Math.exp(-znetj[j])));
            }
            ynetk = (1+
            (zj[0]*bobotwkj[0][0]+zj[1]*bobotwkj[1][0]+zj[2]*bobot
```



```

wkj[2][0]+zj[3]
                                *bobotwkj[3][0]));
yk = (1/(1+Math.exp(-ynetk)));
hasil.set(k, yk);
if (data.get(k).open < hasil.get(k) )
{
    khasil.set(k, 1);
}
else
{
    khasil.set(k, 0);
}

//=====Masuk Tahapan
Backpropagation=====\\
    galatk = ((data.get(k).target-yk)*(yk)*(1-
yk));
    for (int i=0;i<hidden;i++)
    {
        deltawkj[i] =
learningrate*galatk*zj[i];
        galatnetj[i] = galatk*bobotwkj[i][0];
        galatj[i] = galatnetj[i]*zj[i]*(1-
zj[i]);
        deltawkjt[i] =
(deltawkj[i]+(momentum*deltawkj[i]));
        bobotwkj[i][0] =
bobotwkj[i][0]+deltawkjt[i]; //bobot yang baru
        for (int j = 0; j<input; j++)
        {
            if (i==0)
            {
                deltavi[j][i] =
learningrate*galatj[j]*data.get(k).open;
            }
            else if(i==1)

```

```
{
    deltavij[j][i] =
learningrate*galatj[j]*data.get(k).high;
}
else if(i==2)
{
    deltavij[j][i] =
learningrate*galatj[j]*data.get(k).low;
}
else if (i==3)
{
    deltavij[j][i] =
learningrate*galatj[j]*data.get(k).close;
}
deltavijt[j][i] =
(deltavij[j][i]+(momentum*deltavij[j][i]));
bobotvij[j][i] =
bobotvij[j][i]+deltavijt[j][i];
}
}
MSE = MSE+(Math.pow(data.get(k).target-
hasil.get(k),2));
if (khasil.get(k) != kelas.get(k))
{
    temporari.add(k);
}
}
MSE = MSE/data.size();
MSEx.add(MSE);
totaldata.add(data.size());
for (int i=0; i<temporari.size();i++)
{data.add(new Atribut(0, "",
data.get(temporari.get(i)).open,
data.get(temporari.get(i)).high,
```

```
data.get(temporari.get(i)).low,
data.get(temporari.get(i)).close,

data.get(temporari.get(i)).target, "");

hasil.add(hasil.get(temporari.get(i)));

khasil.add(khasil.get(temporari.get(i)));

kelas.add(khasil.get(temporari.get(i)));
}

temporari.clear();
if (MSE<error)
{
    kondisi = 1;
}
else if (MSE==tempmse)
{
    kondisi = 1;
}
epoh++;
}
while(kondisi != 1);
}
public double [][] getbobotvij()
{
    return bobotvij;
}
public double [][] getbobotwkj()
{
    return bobotwkj;
}
}
```

Source code 4.3 Pelatihan

4.2.4 Proses Pengujian

Proses pengujian digunakan untuk menguji bobot akhir yang didapatkan dari pelatihan. Source kode pengujian ditunjukkan pada source kode 4.4 berikut ini.

Proses Pelatihan

```
package alaloui;
import java.util.ArrayList;
public class pengujian
{
    int hidden;
    ArrayList<Atribut> data;
    double[][] vij;
    double[][] wkj;
    double[] znetj;
    double[] zj;
    double ynetk;
    ArrayList<Double> yk= new ArrayList();
    double MSE;
    public pengujian (ArrayList <Atribut>
data,double[][] vij, double[][] wkj)
    {
        this.data = data;
        this.vij = vij;
        this.wkj = wkj;
        hidden=4;
        znetj = new double[hidden];
        zj=new double[hidden];
    }
    public ArrayList<Double> uji()
    {
        for (int k=0; k<data.size();k++)
        {
            for (int j=0;j<hidden;j++)
            {
```

```

        znetj[j] =
            (1+(data.get(k).open*vij[0][j]+data.get(k).high*vij
            [1][j]
            +data.get(k).low*vij[2][j]+data.get(k).close*vij[3]
            [j]));

        zj[j] = (1/(1+Math.exp(-
        znetj[j]))));

    }
    ynetk = (1+
    (zj[0]*wkj[0][0]+zj[1]*wkj[1][0]+zj[2]*wkj[2][0]+zj
    [3]
    *wkj[3][0]));
    yk.add((1/(1+Math.exp(-ynetk))));
    MSE =
    MSE+(Math.pow(data.get(k).target-yk.get(k),2));
    }
    MSE = MSE/data.size();
    return yk;
    }
    public double getmse()
    {
        return MSE;
    }
}

```

Source Kode 4.4 Pengujian

4.2.5 Denormalisasi Data

Proses denormalisasi digunakan untuk mengembalikan data hasil pengujian yang masih dalam kondisi normalisasi, agar bisa dibandingkan dengan data yan asli. Source kode denormalisasi dapat dilihat pada source kode 4.5

Denormalisasi

```
public ArrayList<Double> denormalisasi
(ArrayList<Double> dataq)
{
    databaru.clear();
    for (int i=0; i<dataq.size();i++)
    {
        databaru.add(dataq.get(i));
    }
    for (int i=0; i<databaru.size();i++)
    {
        hasils = mintarget+((maxtarget-
mintarget)*((dataq.get(i)-0.1)/0.8));
        databaru.set(i, hasils);
    }
    return databaru;
}
```

Source kode 4.5 denormalisasi

BAB V

PENGUJIAN DAN ANALISIS

Pada bab ini akan dilakukan pengujian dan analisis dari sistem prediksi tren harga saham menggunakan algoritma *Al-Alaoui Backpropagation*.

5.1 Pengujian

Pengujian dilakukan untuk mengetahui hasil akurasi dari implementasi yang dilakukan. Pengujian dilakukan dengan 3 macam jenis pengujian yaitu pengujian *learning rate* dan *momentum* terbaik, pengujian tingkat akurasi terhadap data latih, dan pengujian tingkat akurasi terhadap banyaknya data uji.

5.1.1 Pengujian *learning rate* dan *momentum* terbaik

Pada pengujian *learningrate(a)* dan *momentum(m)* terbaik, data latih yang digunakan adalah sebanyak 353 data. Tabel 5.1 berikut ini menunjukkan MSE terakhir dari hasil pengujian pengaruh *learningrate(a)* dan *momentum(m)*.

Tabel 5.1 Tabel Pengujian *learning rate* dan *Momentum*

a/m	0.5	0.6	0.7	0.8	0.9
0.1	9.91E-04	8.11E-04	8.20E-04	9.80E-04	9.56E-04
0.2	7.63E-04	6.68E-04	9.29E-04	9.86E-04	Heap
0.3	9.84E-04	9.86E-04	9.94E-04	9.73E-04	9.95E-04
0.4	9.39E-04	Heap	9.52E-04	9.82E-04	9.63E-04
0.5	9.99E-04	9.61E-04	9.60E-04	9.88E-04	9.53E-04
0.6	9.88E-04	Heap	9.82E-04	9.67E-04	9.37E-04
0.7	9.67E-04	1.00E-03	9.90E-04	9.91E-04	9.85E-04
0.8	9.92E-04	9.79E-04	9.94E-04	9.80E-04	9.72E-04
0.9	9.98E-04	9.80E-04	9.58E-04	9.94E-04	9.99E-04

Dari hasil yang terdapat pada table 5.1 dapat disimpulkan bahwa pasangan *learningrate(a)* dan *momentum(m)* terbaik adalah 0.2 dan 0.6 dengan MSE terakhir sebesar 6.68E-04. Sedangkan didapatkan heap untuk pasangan a=0.4 m=0.6, a=0.6

$m=0.6$, dan $a=0.2$ $m=0.9$ dikarenakan keterbatasan perangkat yang tidak cukup menampung memori.

5.1.2 Pengujian tingkat akurasi terhadap banyaknya data latih

Pada pengujian tingkat akurasi terhadap banyaknya data latih, data latih yang digunakan adalah 1 bulan, 2 bulan, 4 bulan, 6 bulan, 16 bulan, dan 17 bulan. Sedangkan data uji yang digunakan tetap 1 bulan. Untuk *learningrate* dan *momentum* yang digunakan adalah 0.2 dan 0.6 dari pengujian sebelumnya. Hasil dari pengujian tingkat akurasi terhadap banyaknya data latih dapat dilihat pada tabel 5.2.

Tabel 5.2 Hasil uji terhadap jumlah data latih

Banyak Data	Akurasi
1 bulan	90%
2 bulan	90%
4 bulan	85%
6 bulan	90%
16 bulan	80%
17 bulan	95%

Dari hasil pada tabel 5.2 diatas dapat disimpulkan akurasi terbaik didapatkan ketika banyak data yang dilatih berjumlah 17 bulan.

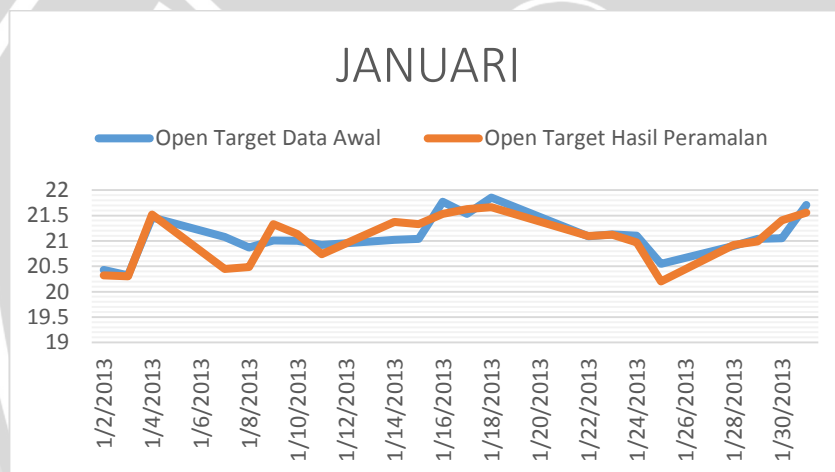
5.1.3 Pengujian tingkat akurasi terhadap banyaknya data uji

Pada pengujian tingkat akurasi terhadap banyaknya data uji, data latih yang digunakan adalah 17 bulan. Sedangkan data uji yang digunakan adalah 1 bulan, 2 bulan, 3 bulan, 4 bulan, 5 bulan. Untuk *learningrate* dan *momentum* yang digunakan adalah 0.2 dan 0.6 dari pengujian sebelumnya. Hasil dari pengujian tingkat akurasi terhadap banyaknya data uji dapat dilihat pada tabel 5.3.

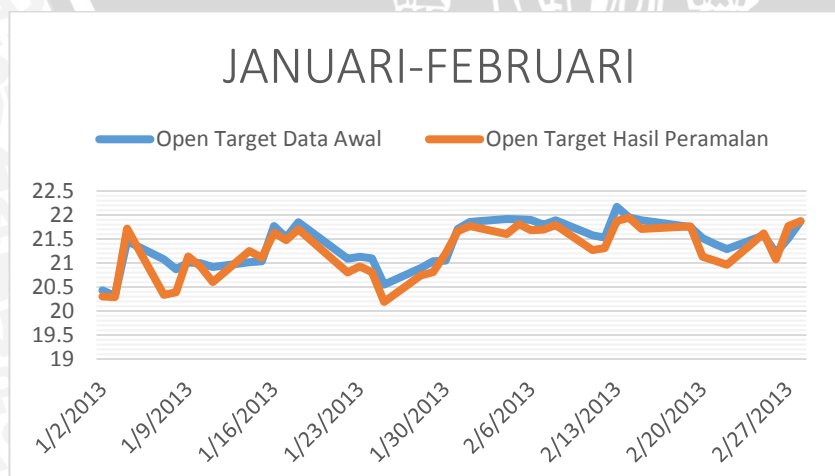
Tabel 5.3 Hasil uji terhadap jumlah data uji

Banyak Data	Akurasi
1 Bulan	95%
2 Bulan	90%
3 Bulan	86%
4 Bulan	81%
5 Bulan	82%

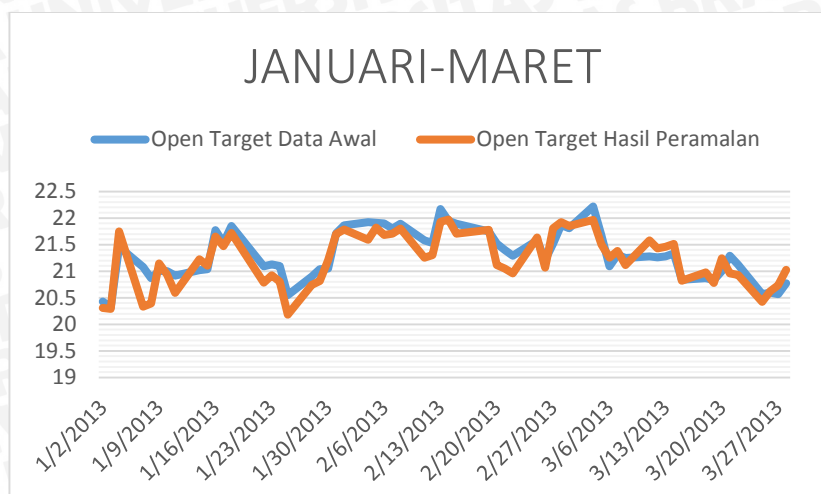
Berdasarkan pengujian pengaruh jumlah data terhadap akurasi, dibuat grafik hubungan data asli dengan data hasil prediksi untuk melihat keakuratan sistem. Grafik ditunjukkan pada grafik 5.1-5.5.



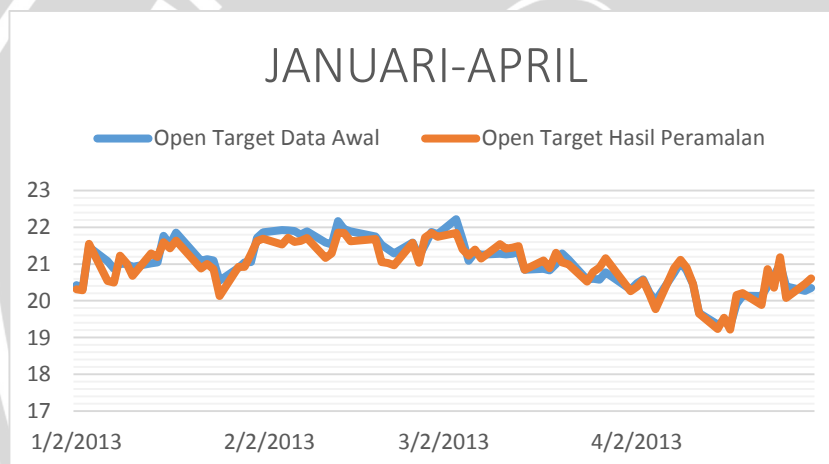
Gambar 5.1 Grafik 1 Bulan



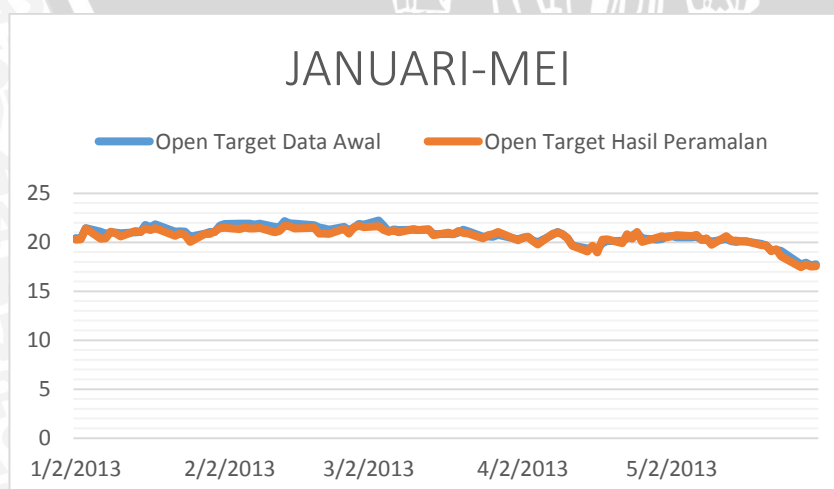
Gambar 5.2 Grafik 2 Bulan



Gambar 5.3 Grafik 3 Bulan



Gambar 5.4 Grafik 4 bulan



Gambar 5.5 Grafik 5 bulan

Dari 5 grafik tersebut algoritma *Al-Alaoui Backpropagation* sudah bisa mengikuti tren dari data aslinya. Sehingga hasil prediksi algoritma *Al-Alaoui Backpropagation* ini cukup akurat.

5.2 Analisis

Pada sub bab ini akan dilakukan analisa dari pengujian sistem prediksi tren harga saham dengan menggunakan algoritma *Al-alaoui backpropagation*. Analisis ini dilakukan untuk mengetahui karakteristik data dari pengujian yang telah dilakukan.

5.2.1 Analisis hasil pengujian *learningrate* dan *momentum* terbaik

Pada pengujian pengaruh *learningrate* dan *momentum* terhadap akurasi dapat disimpulkan bahwa hasil paling bagus untuk semua periode ialah ketika *learningrate* 0.2 dan *momentum* 0.5. Hasil bisa dilihat pada tabel 5.1. dari hasil yang ada pada tabel 5.1 dapat disimpulkan besarnya *learning rate* dan besarnya *momentum* mempengaruhi perubahan MSE namun tidak mempengaruhi bagusnya nilai MSE.

5.2.2 Analisis hasil pengaruh jumlah data latih terhadap akurasi

Pada pengujian pengaruh jumlah data latih terhadap akurasi, banyaknya data latih dapat memperbanyak variasi data dan merusak pola sehingga mempengaruhi akurasi data.

5.2.3 Analisis hasil pengaruh jumlah data uji terhadap akurasi

Pada pengujian pengaruh jumlah data uji terhadap akurasi, banyaknya data uji dapat mengurangi akurasi dengan pola data menurun, dapat dilihat pada tabel 5.3.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil penelitian tentang penggunaan algoritma *al-alaoui backpropagation* pada tren harga saham, dapat disimpulkan bahwa:

1. Jaringan saraf tiruan *al-alaoui backpropagation* untuk prediksi tren harga saham diimplementasikan dengan menggunakan struktur jaringan saraf tiruan yang terdiri dari 4 buah *neuron input*, 4 buah *neuron hidden*, 1 buah *neuron output*, dan menggunakan *learningrate* 0.2, *momentum* 0.5.
2. Berdasarkan pengujian dengan data latih 17 bulan dan pengujian dengan 1 bulan data dihasilkan akurasi sebesar 95%, untuk 2 dan 3 bulan dihasilkan 90%, untuk 4 bulan 87%, dan 5 bulan dihasilkan 84%.

6.2 Saran

Beberapa saran yang dapat dijadikan bahan pertimbangan untuk penelitian selanjutnya adalah sebagai berikut:

1. Penelitian ini hanya melakukan pengenalan terhadap tren dari harga saham. Diperlukan penelitian lebih lanjut untuk meneliti tidak hanya tren mungkin juga harga saham itu sendiri.
2. Penelitian hanya dilakukan terhadap saham suatu perusahaan tertentu, diharapkan penelitian lebih lanjut untuk bisa memprediksi saham tidak terbatas pada satu perusahaan.
3. Penelitian ini belum meneliti seberapa tinggi akurasi terhadap tingkat duplikasi. Diperlukan penelitian lebih lanjut untuk menelitinya.