

PERAMALAN HARGA SAHAM MENGGUNAKAN ALGORITMA BACKPROPAGATION MOMENTUM DENGAN AUTOKORELASI ARIMA

Dwi Putri Aminingtias (150411200139)

Program Studi Teknik Informatika – Fakultas Teknik

Universitas Trunojoyo Madura

Jl. Raya Telang PO BOX 2 Kamal, Bangkalan – Madura

E-mail : aminingtiasd@yahoo.com

Abstact

The main problem that an investor has with the fluctuation of stock prices are dynamic and always changing every day. Such changes can make an investor confused in choosing which shares to buy and which shares to sell. Telecommunication sector one of which is very high competition such as XL axiata, Indosat, Telkomsel, and Smartfren. With the stock price prediction using Neural Network Algorithm Backpropagation method with momentum. Data from stock prices can be determined based on time series. With this algorithm, the networks can be trained using stock price data from the previous situation, so it can help the investors decision-making next period. Based on the results of stock price forecasting using Backpropagation momentum with ARIMA autocorrelation yields an average value of accuracy of 98%, with Smallest MSE 0.02 and MAPE 0.59. Difference error generated network using autocorrelation is more accurate than forecasting without using autocorrelation, MAPE generated equal to 0.14%

Keywords: Neural Network, Backpropagation, ARIMA, Momentum Predicted Stock Price.

Abstak

Masalah utama yang dimiliki seorang *investor* adalah dengan adanya *fluktuasi* harga saham yang dinamis dan selalu berubah setiap hari. Perubahan tersebut dapat membuat seorang *investor* kebingungan dalam memilih saham mana yang harus dibeli dan saham yang harus dijual. Sektor *Telekomunikasi* salah satunya yang persaingannya sangat tinggi seperti *XL axiata*, *Indosat*, *Telkomsel*, dan *Smartfren*. Dengan adanya prediksi harga saham menggunakan metode *Neural Network Algoritma Backpropagation* dengan *momentum*. Data dari harga saham dapat ditentukan berdasarkan runtunan waktu (*time series*). Dengan algoritma ini, jaringan-jaringan dapat dilatih dengan menggunakan data harga saham dari situasi sebelumnya, sehingga dapat membantu para *investor* pengambilan keputusan periode berikutnya. Berdasarkan hasil peramalan harga saham menggunakan Backpropagation momentum dengan *autokorelasi ARIMA* menghasilkan nilai rata-rata akurasi sebesar 98%, dengan MSE terkecil 0.02 dan MAPE 0.59. Selisih *error* yang dihasilkan jaringan menggunakan *autokorelasi* lebih akurat dibandingkan peramalan tanpa menggunakan *autokorelasi*, *MAPE* yang dihasilkan sebesar 0.14%

Kata kunci : *Jaringan Syaraf Tiruan, Backpropagation, ARIMA ,Momentum* Prediksi Harga Saham.

BAB I PENDAHULUAN

1.1 Latar Belakang

Pasar modal merupakan tempat calon pembeli atau *investor* untuk membeli saham suatu perusahaan. Banyak cara yang digunakan calon *investor* untuk memilih perusahaan yang tepat, salah satunya melakukan analisis dengan menggunakan indeks pasar saham. Data harga saham perusahaan merupakan hal yang paling menarik perhatian bagi *investor*.

Tinggi rendahnya harga saham dapat dipengaruhi oleh banyak faktor seperti kondisi dan kinerja perusahaan, resiko, tingkat suku bunga, kondisi perekonomian, kebijaksanaan pemerintah, laju inflasi, penawaran dan permintaan serta masih banyak faktor lainnya. Karena dimungkinkan adanya perubahan faktor-faktor di atas harga saham dapat naik atau turun. Sektor *Telekomunikasi* salah satunya yang persaingannya sangat tinggi seperti *XL axiata*, *Indosat*, *telkomsel*, dan *smartfren*. Masalah utama yang dimiliki seorang *investor* adalah dengan adanya *fluktuasi* harga saham yang dinamis dan selalu berubah setiap hari. Perubahan tersebut dapat membuat seorang investor kebingungan dalam memilih saham mana yang harus dibeli dan saham mana yang harus dijual.

Prediksi harga saham sangat bermanfaat bagi investor untuk dapat melihat bagaimana *investasi* saham sebuah perusahaan di masa yang akan datang dan mengantisipasi naik turunnya harga saham sehingga tidak mengalami kerugian. Dengan adanya prediksi, sangat membantu para *investor* dalam pengambilan keputusan.

Salah satu metode yang dapat digunakan untuk memprediksi harga saham yang berfluktuasi dinamis adalah metode jaringan syaraf tiruan (*ArtificialNeuralnetwork*). Penelitian yang dilakukan oleh Solechan dan Shinta (2012) diperoleh bahwa metode JST memiliki nilai MSE lebih kecil dari pada metoderegresi linier. Selain itu, penelitian yang dilakukan oleh Rufiyanti (2015) diperoleh bahwa Metode JST memiliki nilai MSE lebih kecil dari metode ARIMA. Jaringan syaraf tiruan merupakan model yang dikembangkan berdasarkan sistem kerja syaraf biologi yang tidak memerlukan asumsi-asumsi yang harus dipenuhi seperti pada model-model peramalan konvensional.

Penelitian ini merupakan pengembangan dari penelitian terdahulu (Barus, 2015) yang menggunakan metode *Backpropagatio* dengan data *high*, *low*, dan *close* pada saham sector Telekomunikasi. Sedangkan penelitian yang akan dilakukan adalah prediksi harga saham dengan metode *Backpropagation* pada Telekomunikasi Tbk. Dengan menambah *momentum* yang berfungsi agar bobot yang dihasilkan tidak terlalu mencolok dan menambahkan *Autokorelasi ARIMA* yang berfungsi untuk menentukan *input* data yang berpengaruh serta *K-Fold Cross Validation* untuk menentukan data *training* dan data *testing* dan mengetahui bentuk pola data.

Oleh karena itu untuk akan dibuat Sistem Peramalan Harga Saham Menggunakan Metode *Neural Network Algoritma Backpropagation Dengan Momentum* untuk menyelesaikan suatu masalah tersebut, dimana jaringan syaraf tiruan memerlukan algoritma belajar. *Algoritma* belajar yang digunakan untuk memprediksi harga saham ini adalah dengan metode Propagasi Balik (*Backpropagation*) pada sektor *Telekomunikasi*. Dengan *algoritma* ini, jaringan-jaringan dapat dilatih dan data harga saham ditentukan berdasarkan runtunan waktu (*time series*) dengan uji *ARIMA Box-Jenkins* untuk mengetahui *plot* data harga saham dan hasil sebagai data *training*, menambahkan nilai *momentum* untuk menghindari perubahan bobot yang mencolok, menggolongkannya dan menyesuaikan bobot penghubung dalam jaringan sebagai input baru dan meramalkan harga saham berikutnya. Penelitian untuk prediksi harga saham dilakukan secara harian (*short term*) artinya butuh data hari sebelumnya untuk melakukan prediksi pada hari berikutnya agar data hasil prediksi dengan target lebih akurat.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas, dapat dirumuskan suatu permasalahan yaitu :

1. Berapa akurasi tingkat error yang dihasilkan jika tanpa menggunakan proses *Autokorelasi ARIMA* pada peramalan harga saham menggunakan *backpropagation* dengan *momentum*?
2. Seberapa besar keakuratan hasil peramalan harga saham terhadap nilai error menggunakan algoritma *Backpropagation* dengan *momentum*?

1.3 Tujuan

Adapun tujuan penelitian tersebut antara lain:

1. Mengetahui perbandingan akurasi error yang dihasilkan menggunakan *autokorelasi* dan tanpa *autokorelasi* pada peramalan harga saham *Backpropagation momentum*.
2. Mengetahui berapa besar akurasi nilai yang diperoleh menggunakan metode *Neural Network* algoritma *Backpropagation* dengan *momentum* untuk menentukan harga saham untuk periode yang akan datang.

1.4 Manfaat

Manfaat dari skripsi ini yaitu dengan adanya sistem peramalan harga saham ini, dapat mempermudah *investor* terutama sektor telekomunikasi untuk menjual atau membeli sahamnya agar di periode yang akan datang tidak mengalami penurunan atau kerugian.

1.5 Batasan Masalah

Batasan masalah pembuatan sistem ini mencakup:

1. Data harga saham yang digunakan adalah jumlah *high*, *low* dan *close* pada masing-masing perusahaan di sektor telekomunikasi.
2. Untuk *input* dengan jumlah banyak, sistem berproses dengan lambat

BAB II

LANDASAN TEORI

2.1 Penelitian Sebelumnya

Penelitian yang dilakukan oleh Edy Supriyanto (2004) dengan metode “Jaringan Saraf Tiruan *Backpropagation*”kelebihannya Proses belajar dapat menyesuaikan bobot-bobot koneksi dan kekurangannya Untuk menentukan parameter yang berpengaruh baik dilakukan pengujian berulang kali, karna hasil tidak pasti.

Penelitian yang dilakukan Prisa Marga Kusumantara, I Gede Susrama (2007) menggunakan metode “Jaringan Syaraf Tiruan *Backpropagation* dan *Exponential Smoothing*”. Dengan kelebihan *Exponential Smoothing* lebih besar menghasilkan nilai akurasi *MAPE*, karena metode tersebut tidak cocok untuk prediksi jangka pendek. Dan kekurangannya yaitu Metode *Backpropagation* menghasilkan nilai akurasi yang kecil, karena proses pelatihan yang dilakukan dengan menggunakan *hidden layer* yang terbaik.

Penelitian Warih Maharani Jurnal SNATI (2009) menggunakan “*JST Backpropagation Momentum Dengan Adaptive learning Rate*”. Kelebihannya adanya *momentum* dan *adaptive learning rate*, memparcepat proses pembelajaran. Kekurangannya Jika kurang tepat memilih nilai

momentum dan *learning rate*, proses pembelajaran akan lambat.

Penelitian Dwi Efri Rufiyanti (20015) menggunakan metode “Penerapan Jaringan Saraf tiruan *backpropagation*, dan ARIMA ” Kelebihannya menggunakan *JST BP* lebih optimal dengan nilai akurasi terkecil dari metode ARIMA dan metode Hibrid ARIMA-JST. Kelemahannya proses pembelajaran atau pelatihan lebih lambat dari faktor banyak data.

Penelitian yang dillakukan Yure Firdaus Arifin,Dian Eka Ratnawati , Putra Pandu Adikara (2017) menggunkan “Metode *Bayesian* dan Metode *Backpropagation*” Kelebihannya gabungan antara *Backpropagation* dan metode *Bayesian*, Karena lebih sedikit iterasi. Kekurangannya hasil prediksi paling optimal pada tahun pertama, untuk tahun berikutnya hasil semakin tidak akurat

2.2 Landasan Teori

Landasan teori adalah kumpulan teori yang diambil dari beberapa sumber referensi untuk merancang sistem peramalan harga saham menggunakan metode *Backpropagation*. Teori-teori tersebut akan dijelaskan di bawah ini.

2.2.1 Pengertian Saham

Saham adalah sertifikat atau tanda otentik yang mempunyai kekuatan hukum bagi pemegangnya sebagai keikutsertaan di dalam perusahaan serta mempunyai nilai nominal (mata uang) serta dapat diperjualbelikan.Saham merupakan surat berharga yang bersifat kepemilikan. Artinya si pemilik saham merupakan pemilik perusahaan. Semakin besar saham yang dimiliki, maka semakin besar pula kekuasaannya di perusahaan tersebut [1].

2.2.2 Peramalan (Forecasting)

Peramalan (*forecasting*) merupakan suatu usaha untuk meramalkan keadaan di masa mendatang melalui pengujian keadaan di masa lalu. Proses peramalan merupakan suatu unsur yang sangat penting dalam pengambilan keputusan, sebab efektif tidaknya suatu keputusan sering kali dipengaruhi beberapa faktor yang tidak tampak pada saat keputusan itu diambil. Peramalan bertujuan untuk mendapatkan perkiraan atau prediksi yang bisa meminimumkan kesalahan dalam meramal yang biasanya diukur dengan *Mean Square Error*[2].

2.2.3 Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan syaraf ini

diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran[4].

Jaringan Syaraf Tiruan ditentukan oleh tiga hal :

1. Pola hubungan antar neuron
 2. Metode untuk menentukan bobot penghubung (metode *Training/Learning/Algoritma*)
 3. Fungsi aktivasi
- Beberapa istilah dalam jaringan syaraf tiruan yang sering ditemui antara lain:
1. Neuron atau node atau unit
Sel syaraf yang merupakan elemen pengolahan jaringan syaraf tiruan. Setiap neuron menerima data input, memroses input tersebut (melakukan sejumlah perkalian dengan melibatkan *summation function* dan fungsi aktivasi), dan mengirimkan hasilnya berupa sebuah output.
 2. Jaringan
Kumpulan neuron yang saling terhubung dan membentuk lapisan.
 2. Input atau masukan
Berkoresponden dengan sebuah atribut tunggal dari sebuah pola atau data lain dari dunia luar. Sinyal-sinyal input ini kemudian diteruskan ke lapisan selanjutnya.
 3. Output atau keluaran
Solusi atau hasil pemahaman jaringan terhadap data input. Tujuan pembangunan jaringan syaraf tiruan sendiri adalah untuk mengetahui nilai *output*.
 4. Lapisan tersembunyi (*Hidden layer*)
Lapisan yang tidak secara langsung berinteraksi dengan dunia luar. Lapisan ini memperluas kemampuan jaringan syaraf tiruan dalam menghadapi masalah-masalah yang kompleks.
 5. Bobot
Bobot dalam jaringan syaraf tiruan merupakan nilai matematis dari koneksi, yang mentransfer data dari satu lapisan ke lapisan lainnya. Bobot ini digunakan untuk mengatur jaringan sehingga jaringan syaraf tiruan bisa menghasilkan output yang diinginkan sekaligus bertujuan membuat jaringan tersebut belajar.
 6. *Summation function*
Fungsi yang digunakan untuk mencari rata-rata bobot dari semua elemen *input*.
 7. Fungsi aktivasi atau fungsi transfer
Fungsi yang menggambarkan hubungan antara tingkat aktivasi internal (*summation function*) yang mungkin berbentuk linier atau nonlinier.
 8. Paradigma pembelajaran

Cara pembelajaran atau pelatihan jaringan syaraf tiruan yaitu apakah terawasi, tidak terawasi, atau merupakan gabungan keduanya (*hybrid*).

9. Aturan pembelajaran .

Aturan kerja secara umum dari teknik/algoritma jaringan syaraf tiruan. [4].

2.2.4 Algoritma Backpropagation

Salah satu algoritma pelatihan jaringan syaraf tiruan yang dapat dimanfaatkan dalam menyelesaikan sistem pendukung keputusan adalah propagasi balik. Algoritma ini umumnya digunakan pada jaringan syaraf tiruan yang berjenis *multi-layer feed-forward*, yang tersusun dari beberapa lapisan dan sinyal dialirkan secara searah dari input menuju output[5].

Algoritma pelatihan *propagasi* balik pada dasarnya terdiri dari tiga tahapan yaitu:

- a. *Input* nilai data pelatihan sehingga diperoleh nilai *output*.
- b. *Propagasi* balik dari nilai *error* yang diperoleh.
- c. Penyesuaian bobot koreksi untuk meminimalkan nilai *error*.

Ketiga tahapan tersebut diulangi terus-menerus sampai mendapatkan nilai *error* yang diinginkan. Setelah *training* selesai dilakukan, hanya tahap pertama yang diperlukan untuk memanfaatkan jaringan syaraf tiruan tersebut. informasi *error* dipropagasikan secara berurutan bermula dari *output* layer dan berakhir pada *input* layer, sehingga algoritma ini diberi nama *propagasi* balik (*backpropagation*).

Notasi yang digunakan dalam algoritma pelatihan jaringan syaraf tiruan adalah :

x = Data training *input* $x = (x_1, \dots, x_i, \dots, x_n)$

t = Data *training* untuk target *output* $t = (t_1, \dots, t_k, \dots, t_m)$

α = *Learning rate* yaitu parameter untuk mengontrol perubahan bobot selama pelatihan. Semakin besar *learning rate*, maka jaringan syaraf tiruan akan semakin cepat belajar tetapi hasilnya kurang akurat. Semakin kecil *learning rate*, maka jaringan syaraf tiruan akan semakin lambat belajar tetapi hasilnya lebih akurat[5].

X_i = Unit *input* ke- i

Z_j = Hidden unit ke- j

Y_k = Unit *output* ke- k

v_{0j} = Bias untuk *hidden* unit ke- j

v_{ij} = Bobot antara unit *input* ke- i dengan hidden unit ke- j

w_{0k} = Bias untuk unit *output* ke- k

W_{jk} = Bobot antara hidden unit ke- j dengan unit *output* ke- k

δ_k = Faktor koreksi *error* untuk bobot w_{jk}

δ_j = Faktor koreksi *error* untuk bobot v_{ij}

m = Momentum

Langkah 0 : Inisialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil di sekitar -0.5 sampai 0.5)

Langkah 1 : Jika *stopping condition* masih belum terpenuhi, jalankan langkah 2-9

Langkah 2 : Untuk setiap data pelatihan, lakukan langkah 3-8

Feedforward:

Langkah 3 : Untuk tiap unit masukan ($X_i, i=1, \dots, n$) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).

Langkah 4 : Untuk tiap unit tersembunyi ($Z_j, j=1, 2, 3, \dots, p$) akan menjumlahkan sinyal-sinyal *input* yang sudah terbobot :

$$\mathbf{z_in}_j = \mathbf{v}_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.1)$$

Kemudian dihitung nilai *output* dengan menggunakan fungsi aktivasi yang dipilih :

$$\mathbf{z}_j = \mathbf{f}(\mathbf{z}_{ij}) \quad (2.2)$$

dimana fungsi aktifasi yang digunakan ialah fungsi *sigmoid biner* yang mempunyai persamaan :

$$\mathbf{f1}(x) = \frac{1}{1+e^{(-x)}} \quad (2.3)$$

Lalu mengirim sinyal *output* ini ke seluruh unit pada unit *output*.

Langkah 5 : Untuk tiap unit *output* ($Y_k, k=1, 2, 3, \dots, m$) akan menjumlahkan sinyal-sinyal *input* yang sudah terbobot :

$$\mathbf{y_in}_k = \mathbf{w}_{0k} + \sum_{j=1}^p \mathbf{z}_j \mathbf{w}_{jk} \quad (2.4)$$

Kemudian dihitung nilai *output* dengan menggunakan fungsi aktivasi :

$$\mathbf{y}_k = \mathbf{f}(\mathbf{y_in}_k) \quad (2.5)$$

Propagasi error (backpropagasi of error) :

Langkah 6: Untuk tiap unit *output* ($Y_k, k=1, \dots, m$) menerima pola target yang bersesuaian dengan pola masukan, dan kemudian dihitung informasi kesalahan :

$$\delta_k = (t_k - y_k) f'(\mathbf{y_in}_k) \quad (2.6)$$

Sebagaimana *input* data pelatihan, *output* data pelatihan t_k juga telah diskalakan menurut fungsi aktivasi yang dipakai.

Faktor δ_k digunakan untuk menghitung koreksi error (ΔW_{jk}) yang nantinya akan dipakai untuk memperbaharui W_{jk} , dimana :

$$\Delta W_{jk} = \alpha \delta_k z_j \quad (2.7)$$

Selain itu, juga dihitung koreksi bias ΔW_{0k} yang nantinya akan dipakai untuk memperbaharui ΔW_{0k} , dimana :

$$\Delta W_{0k} = \alpha \delta_k \quad (2.8)$$

Faktor δ_k kemudian dikirimkan ke *layer* yang berada pada langkah 7.

Langkah 7: Setiap *hidden unit* ($Z_j, j = 1, \dots, p$) menjumlah *input* delta (yang dikirim dari *layer* langkah 6) yang sudah terbobot.

$$\delta_in_j = \sum_{k=1}^m \delta_k W_{jk} \quad (2.9)$$

Kemudian hasilnya dikalikan dengan turunan dari fungsi aktivasi yang digunakan jaringan untuk menghasilkan factor koreksi *error* j , dimana :

$$\delta_j = \delta_in_j f'(\mathbf{z_in}_j) \quad (2.10)$$

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.11)$$

Selain itu juga dihitung koreksi bias ΔV_{0j} yang nantinya akan dipakai untuk memperbaharui V_{0j} , dimana :

$$\Delta v_{0j} = \alpha \delta_j \quad (2.12)$$

Pembaharuan bobot (adjustment) dan bias

Langkah 8 : Setiap unit *output* ($Y_k, k = 1, \dots, m$) akan memperbaharui bias dan bobotnya dari setiap *hidden unit* ($j = 0, \dots, p$)

$$\mathbf{w}_{jk}(\text{baru}) = \mathbf{w}_{jk}(\text{lama}) + \Delta \mathbf{w}_{jk} \quad (2.13)$$

Demikian pula, setiap *hidden unit* ($Z_j, j = 1, \dots, p$) akan memperbaharui bias dan bobotnya dari setiap unit *input* ($i = 0, \dots, n$)

$$\mathbf{v}_{ij}(\text{baru}) = \mathbf{v}_{ij}(\text{lama}) + \Delta \mathbf{v}_{ij} \quad (2.14)$$

Langkah 9 : Memeriksa *stop condition*.

Jika *stop condition* terpenuhi, pelatihan jaringan dapat dihentikan.

Untuk menentukan *stopping condition* terdapat tiga cara yang biasa dipakai yaitu sebagai berikut :

- Pertama, dengan membatasi iterasi yang ingin dilakukan.

- Cara kedua adalah membatasi *error*. Pada metode *Backpropagation*, dipakai metode *Mean Square Error* untuk menghitung rata-rata *error output* yang dikehendaki pada data pelatihan dengan *output* yang dihasilkan oleh jaringan.

$$MSE = 0.5 \times \{(t_{k1} - y_{k1})^2 + (t_{k2} - y_{k2})^2 + \dots + (t_{km} - y_{km})^2\}$$

- Ketiga, ada kalanya sebelum mencapai kondisi seperti yang diinginkan, *error* justru semakin besar (*overtraining*). Jika salah satu dari *training set error* atau *test set error* bertambah besar, pelatihan harus dihentikan[6].

2.2.5. Momentum

Modifikasi dalam *Backpropagation* standart dapat dilakukan dengan melakukan perubahan bobot yang didasarkan atas arah gradient pola terakhir pola sebelumnya yang disebut momentum.

Penambahan momentum ditujukan untuk menghindari perubahan bobot yang mencolok. Akibat adanya data yang diberikan ke jaringan memiliki pola

serupa, maka perubahan bobot dilakukan secara cepat. Namun apabila data terakhir yang dimasukkan memiliki pola yang berbeda dengan pola sebelumnya, maka perubahan dilakukan secara lambat.

Dengan penambahan momentum, bobot baru pada waktu ke (t+1) didasarkan atas bobot pada waktu t dan (t-1). Disini harus ditambahkan 2 variabel baru yang mencatat besarnya momentum untuk 2 iterasi terakhir. Jika μ adalah konstanta ($0 \leq \mu \leq 1$) yang menyatakan parameter momentum maka bobot baru dihitung berdasarkan persamaan :

$$w_{jk} = w_{kj}(t) + \alpha \delta_k z_j + \mu [w_{jk}(t) - w_{jk}(t-1)]$$

Persamaan pada lapisan *hidden*

$$v_{ij}(t+1) = v_{ij}(t) + \alpha \delta_j x_i + \mu [v_{ij}(t) - v_{ij}(t-1)]$$

Dimana μ adalah konstanta *momentum* yang biasanya berharga positif dengan range dari 0 sampai 1 [7].

2.2.6. Model Autoregresif Integrated Moving Average (ARIMA)

ARIMA sering juga disebut metode runtun waktu Box-Jenkins. ARIMA sangat baik ketepatannya untuk peramalan jangka pendek, sedangkan untuk peramalan jangka panjang ketepatan peramalannya kurang baik. Biasanya akan cenderung *flat* (mendatar/konstan) untuk periode yang cukup panjang.

ARIMA cocok jika observasi dari deret waktu (*time series*) secara statistik berhubungan satu sama lain (*dependent*) [8]. Langkah-langkah analisis runtun waktu sebagai berikut :

1. Plot data Langkah pertama yang harus dilakukan adalah memplot data asli, dari plot tersebut bisa dilihat apakah data sudah stasioner. Jika data belum stasioner dalam mean maka perlu dilakukan proses *differencing*.
2. Identifikasi model Setelah data stasioner dalam *mean* dan variansi langkah selanjutnya adalah melihat plot ACF dan PACF. Dari plot ACF (*autocorrelation function*) dan PACF (*partial autocorrelation function*) tersebut bisa diidentifikasi beberapa kemungkinan model yang cocok untuk dijadikan model.
3. Estimasi model Setelah berhasil menetapkan beberapa kemungkinan model yang cocok dan mengestimasi parameternya. Lalu dilakukan uji signifikansi pada koefisien. Bila koefisien dari model tidak signifikan maka model tersebut tidak layak digunakan untuk peramalan.
4. Uji asumsi residual (*diagnostic checking*) Dari beberapa model yang signifikan tersebut dilakukan uji asumsi residual.
5. Pemilihan model terbaik Hal-hal yang perlu diperhatikan dalam mengambil model adalah sebagai berikut :

- a. Prinsip parsimony yaitu model harus bisa sesederhana mungkin. Dalam arti mengandung sesedikit mungkin parameternya, sehingga model lebih stabil.
 - b. Model sebisa mungkin memenuhi (paling tidak mendekati) asumsi-asumsi yang melandasinya.
 - c. Dalam perbandingan model, selalu pilih model yang paling tinggi akurasi, yaitu yang memberikan galat (*error*) terkecil.
6. Peralaman Langkah terakhir dari proses runtun waktu adalah prediksi atau peramalan dari model yang dianggap paling baik, dan bisa diramalkan nilai beberapa periode ke depan

2.2.7. Fungsi Autokorelasi

Salah satu bentuk analisis dalam teori statistika adalah analisis data deret waktu (*time series*), yaitu analisis terhadap data yang merupakan fungsi atas waktu atau tempat. Analisis data deret waktu merupakan analisis khusus dari analisis regresi, sebab dalam data deret waktu terlibat suatu besaran yang dinamakan *Autokorelasi*. Keberadaan autokorelasi bisa merupakan autokorelasi periodik, yaitu autokorelasi dengan nilai periodisitasnya lebih dari satu, dan autokorelasi seperti ini banyak terdapat pada data deret waktu yang memiliki komponen musiman-periodik. Perumusan *autokorelasi* sama dengan perumusan korelasi antar dua variabel. Dalam metode Statistika, jika dimiliki sampel atas data bivariat (X, Y). [6]

Dalam hal ini data harga saham berdasarkan waktunya akan diproses dengan rumus autokorelasi sehingga didapatkan *time lags* atau waktu yang signifikan dengan waktu yang diramalkan. Kemudian data harga saham pada waktu yang signifikan tersebut akan menjadi data masukan pada proses pelatihan *Jaringan Syaraf Tiruan Backpropagation*.

Rumus untuk mencari fungsi autokorelasi

$$r_k = \frac{\sum_{t=1}^n (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^n (y_t - \bar{y})^2} \quad k = \text{time lag} \quad (2.19)$$

2.2.8. Normalisasi Data

Data-data yang ada dilakukan normalisasi dengan membagi nilai data tersebut dengan nilai *range* data (nilai data maksimum-nilai data minimum). Normalisasi data *input* bertujuan untuk menyesuaikan nilai *range* data dengan fungsi aktivasi dalam sistem *backpropagation*. Ini berarti nilai kudrat *input* harus berada pada *range* 0 sampai 1. Sehingga *range input* yang memenuhi syarat adalah nilai data *input* dari 0 sampai 1 atau dari -1 sampai 1. Oleh karena itu output yang dihasilkan pun akan berada pada *range* 0 sampai

1. kemudian untuk mendapatkan nilai sebenarnya dari *output* perlu dilakukan proses denormalisasi.

Normalisasi data dengan menggunakan rumus[9] :

$$f(x) = \frac{X_i - X_{min}}{X_{max} - X_{min}} \quad (2.20)$$

Dimana:

X_i = data ke-i

X_{min} = data dengan nilai minimum

X_{max} = data dengan nilai maksimum

Pada proses *testing*, *output* yang dihasilkan oleh jaringan berkisar antara 0 sampai dengan 1 sehingga perlu dilakukan denormalisasi yang berguna untuk mengkonversikan kembali hasil *output* jaringan menjadi harga material normal. setelah itu akan dilakukan perbandingan antara data sebenarnya dengan data hasil prediksi, sehingga dapat dihitung *error* atau prosentase *error*nya.

Denormalisasi data dengan menggunakan rumus :

$$X_i = y(X_{max} - X_{min}) + X_{min} \quad (2.21)$$

Dimana:

X_i = harga material normal

Y = hasil output jaringan

X_{min} = data dengan nilai minimum

X_{max} = data dengan nilai maksimum

2.2.9. Ukuran Ketepatan Metode Peramalan

Ukuran akurasi hasil peramalan yang merupakan ukuran kesalahan peramalan merupakan ukuran tentang tingkat perbedaan antara hasil peramalan dengan permintaan yang sebenarnya terjadi. Ada empat ukuran yang biasa digunakan, yaitu: [5]

1. Mean Absolute Deviation

Mean absolute Deviation (MAD) mengukur akurasi persamaan dengan merata-ratakan nilai *absolut* kesalahan peramalan.

$$MAD = \sum \left| \frac{A_t - F_t}{n} \right|$$

Dimana:

A = Permintaan aktual pada periode -t

F_t = Peramalan permintaan (*forecast*) pada periode-t

N = Jumlah periode peramalan yang terlibat.

2. Means Square Error (MSE)

MSE adalah nilai tingkat kesalahan dari nilai yang diramalkan. Semakin kecil nilai MSE maka semakin kecil pula nilai kesalahan peramalan yang dihasilkan. MSE dapat ditulis dengan rumus :

$$MSE = \sum \frac{(A_t - F_t)^2}{n} \quad (2.23)$$

3. Mean Absolute Percentage Error

Mean Absolute Percentage Error (MAPE) memberikan petunjuk seberapa kesalahan peramalan yang dibandingkan dengan nilai sebenarnya. Hasil perhitungan ditunjukkan dalam satuan persentase.

$$MAPE = \left(\frac{100}{n} \right) \sum \left| \frac{A_t - F_t}{A_t} \right| \quad (2.24)$$

2.2.10. K-Folds Cross Validation

K-fold cross validation merupakan teknik yang membagi data ke dalam k bagian untuk kemudian masing – masing bagian data tersebut akan dilakukan proses klasifikasi.



Gambar 2.6 Dataset menggunakan K-Fold Cross Validation.

Pada gambar 2.6 merupakan arsitektur dataset menggunakan *k-fold cross validation*. Dimana data akan dibagi dalam k-bagian. Dengan menggunakan *k-fold cross validation* akan dilakukan percobaan sebanyak k buah. Tiap percobaan itu akan menggunakan satu buah data *testing* dan k-1 bagian menjadi data *training*, dan kemudian data *testing* tersebut akan ditukar dengan satu buah data *training* sehingga untuk tiap percobaan akan didapatkan data *testing* yang berbeda – beda[13].

BAB III

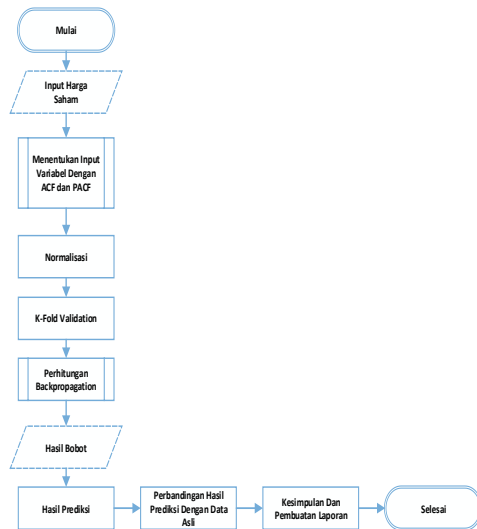
ANALISIS DAN PERANCANGAN SISTEM

3.1 Desain Sistem

Perancangan sistem yang dibuat untuk membangun sistem prediksi dengan *flowchart* yang akan diimplementasikan. Tujuan agar dalam pembuatan sistem akan lebih terkonsep dan memiliki acuan sehingga akan memudahkan ketika melakukan implemtasi pada bahasa pemrograman.

a. Perancangan Flowchart Diagram

Flowchart merupakan kerangka untuk menunjukkan alur sistem. Diagram ini dapat memberi solusi langkah demi langkah untuk menyelesaikan masalah yang ada dalam algoritma tersebut. Berikut merupakan *flowchart* sistem secara umum.



Gambar 3.1 Flowchart Sistem Keseluruhan

Keterangan dari gambar 3.1, Secara garis besar, peramalan harga saham menggunakan Metode *Backpropagation* dapat dituliskan sebagai berikut :

1. Mulai

2. Masukan Inputan

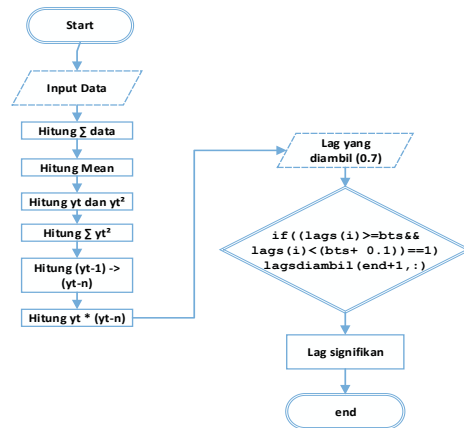
Variabel input yang digunakan dalam penelitian ini adalah data jumlah harga saham dari bulan Desember sampai Februari. yaitu data *low*, *high* dan *close*. [lampiran 1]

3. Menentukan *Input Variabel* dengan Autokorelasi (ACF)

Karena data yang digunakan cukup banyak maka perlu dilakukan normalisasi dengan cara mengkorelasikan setiap data. Sehingga data yang tidak cukup berpengaruh terhadap harga saham berikutnya tidak digunakan untuk proses peramalan. Proses tersebut dapat dilakukan dengan proses *autokorelasi* (7.18).

Di bawah ini merupakan *Flowchart* sistem proses autokorelasi. Yang digunakan untuk menyeleksi data yang digunakan untuk input menuju proses *backpropagation*.

Pada gambar 3.2 merupakan alur menentukan nilai inputan yang berpengaruh menggunakan *autokorelasi* dimana langkah pertama yaitu menyiapkan data yang akan di autokorelasi dan di jumlah. Kemudian hitung nilai *mean*, menentukan nilai *yt*, *yt* kuadrat, menjumlahkan hasil kuadrat dan menghitung *lags* dengan batas *Lags signifikan*



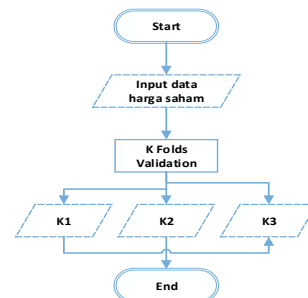
Gambar 3.2 Flowchart Sistem Autokorelasi

4. Normalisasi Data

Proses Normalisasi yang digunakan dalam sistem ini menggunakan normalisasi minimum-maximum. Data-data yang ada dilakukan normalisasi dengan membagi nilai data tersebut dengan nilai *range* data (nilai data maksimum-nilai data minimum). Normalisasi data *input* bertujuan untuk menyesuaikan nilai *range* data dengan fungsi aktivasi dalam sistem *backpropagation*. Ini berarti nilai kuadrat *input* harus berada pada *range* 0 sampai 1. Sehingga *range input* yang memenuhi syarat adalah nilai data *input* dari 0 sampai 1 atau dari -1 sampai 1. Oleh karena itu *output* yang dihasilkan pun akan berada pada *range* 0 sampai 1. kemudian untuk mendapatkan nilai sebenarnya dari *output* perlu dilakukan proses denormalisasi.

5. *K-Fold Cross Validation*

Proses membagi data menjadi data *trining* dan data *testing*. Yang di bagi menjadi 3 skenario. Di bawah ini merupakan *Flowchart* sistem proses *K-Folds Cross Validation*.

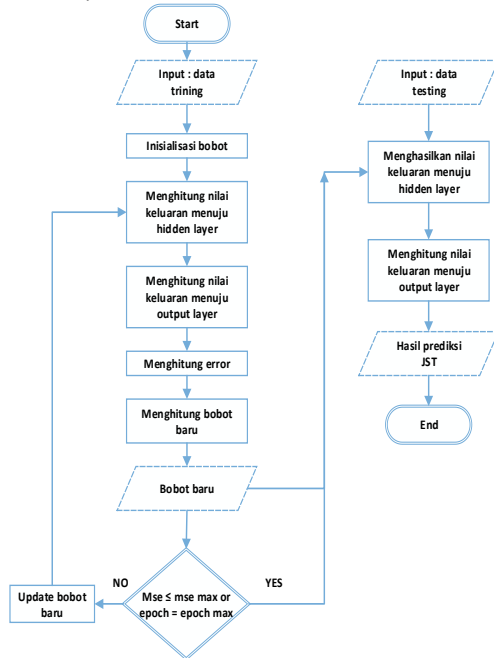


Gambar 3.3 Flowchart K Folds Validation

Menentukan data *training* dan data *testing* pada penelitian ini menggunakan *K Folds Cross Validation*, dimana *k* dapat ditentukan sendiri [6]. Misal ditentukan *k=3* sehingga akan didapat *dataset k1*, *k2*, dan *k3*. Dari 3 *dataset* tersebut akan dipilih 2 menjadi data *training* dan 1 menjadi data *testing*. Misalkan untuk data *training* dipilih *k1* dan *k2*, sedangkan *k3* menjadi data *testing*. Semakin banyak pembagian *k* semakin banyak dan semakin bervariasi uji coba yang dilakukan akan tetapi pengerjaan suatu sistem akan semakin lama.

6. Perhitungan Backpropagation

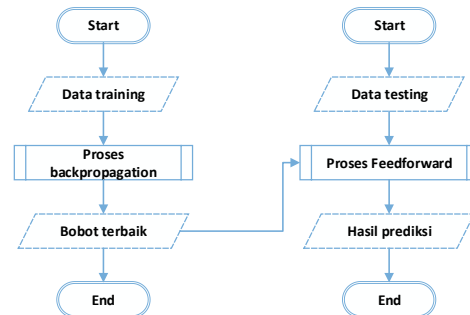
Di bawah ini merupakan *Flowchart* sistem proses menggunakan metode Backpropagation. Pada gambar 3.4 langkah di mulai dari hasil output dari proses autokorelasi yang akan diproses menggunakan Backpropagation untuk memperbarui bobot dan dikirim ke *hidden layer*.



Gambar 3.4 Flowchart Sistem Backpropagation

Pada *hidden layer* akan dijumlahkan input *delta* yang sudah terbobot dan hasilnya akan dikalikan dengan turunan fungsi aktivasi untuk menghasilkan faktor koreksi *error*. Proses tersebut dilakukan terus menerus sampai toleransi *error* atau *maximum epoch* terpenuhi. Terdiri dari beberapa tahap yaitu fase pertama propagasi maju. Kedua propagasi mundur dan yang ketiga perubahan bobot. Ketiga fase tersebut diulang terus hingga kondisi yang diinginkan terpenuhi (jumlah iterasi atau kesalahan *error*). Dalam menggunakan algoritma BP, proses pelatihan dengan proses pengujian memiliki langkah yang berbeda. Dalam proses pelatihan terdiri dari dua proses utama, *feed forward* dan *backpropagation of error* atau *backward*. Sedangkan untuk proses pengujian hanya menggunakan proses *feed forward*. [2.2.7 Algoritma Backpropagation]

Di bawah ini merupakan *Flowchart* proses *training* dan *testing* pada metode *Bckpropagation*,



Gambar 3.5 Flowchart Proses Training dan Testing

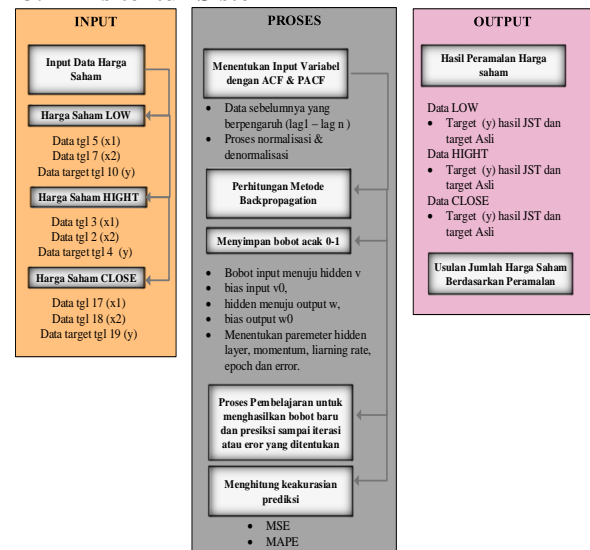
Pada gambar 3.7 proses *trining* dilakukan sehingga mencapai nilai *error* terkecil, dan menghasilkan bobot baru. Bobot-bobot baru tersebut nantinya yang akan digunakan untuk data *testin* sehingga mernghasilkan peramalan dari jaringan saraf tiruan yang digunakan.

7. Menghasilkan output

Output yang dihasilkan pada penelitian ini yaitu sistem peramalan harga saham periode berikutnya.

8. Selesai

3.2 Arsitektur Sistem



Gambar 3.6 Arsitektur Sistem

Proses arsitektur ini dilakukan supaya sistem yang akan dibuat dapat berfungsi sesuai yang diharapkan sehingga berfungsi secara maksimal. Pada gambar 3.6 merupakan arsitektur sistem peramalan harga saham dengan menggunakan metode *Backpropagation*, dibutuhkan 3 input data *low*, *high* dan *close* pada periode sebelumnya agar dapat dilakukan peramalan pada masing masing perusahaan Telekomunikasi.

Karena untuk melakukan perhitungan peramalan, data yang ada pada periode sebelumnya diplot terlebih dahulu sehingga mendapatkan data input yang berpengaruh pada peramalan yang akan datang. Setelah mendapatkan input dari data sebelumnya, sistem akan

melakukan proses perhitungan peramalan dengan menggunakan Metode *Backpropagation* dengan *Momentum*.

Dengan *Iterasi* dan batas *error* maksimal yang ditentukan, maka akan dapat meramalkan harga saham pada masing-masing perusahaan dengan *output* data harga saham *low,high* dan *close* dengan kesalahan terkecil.

3.2 Gambaran Skenario Metode *Backpropagation*

Objek penelitian yang akan dilakukan analisis pada penelitian ini adalah peramalan harga saham pada sektor *Telekomunikasi* perusahaan *XL Axiata*, *Indosat*, *Smartfren* dan *Telkomsel* yang terdaftar di Bursa Efek Indonesia. Data yang diambil adalah data sekunder yang berupa harga saham *high* (Harga saham tertinggi yang diperoleh pada satu hari transaksi), *low* (Harga saham terendah yang diperoleh pada satu hari transaksi) dan *close* (Harga saham penutupan yang ditetapkan pada satu hari transaksi sebelum hari transaksi yang akan diramalkan) pada masing-masing perusahaan. Misalkan dari perusahaan *XL axiata*, dengan banyak data 56 data harga saham mulai dari desember 2016 sampai februari 2017

Variabel *input* yang digunakan untuk contoh perhitungan manual dalam penelitian ini adalah data jumlah harga saham *close*. Karena data yang digunakan cukup banyak maka perlu dilakukan normalisasi dengan cara mengkorelasikan setiap data. Sehingga data yang tidak cukup berpengaruh terhadap harga saham berikutnya tidak digunakan untuk proses peramalan. Proses tersebut dapat dilakukan dengan proses analisa *autokorelasi* (7.18).

Contoh data Awal yang akan di *Autokorelasi*.

Table 3.1 Data Harga Saham *Close* XL Axiata

n	Bulan/tgl	CLOSE
1	Desember	2220
2		2260
3		2270
4		2280
-	Januari	2280
-		2320
-		2560
-		2650
-	Februari	2640
53		3070
54		3130
55		3240
56		3120

Senlanjutnya proses *autokorelasi* untuk menentukan *lags* mana yang berpengaruh pada data

target. Dimana pada proses *autokorelasi* mencari jumlah data, mean jumlah *yt*, *yt* kuadrat dan seterusnya sehingga menghasilkan *lags signifikan* *r1* samapai *r* ke *n*. dan menentukan las terbaik yang akan digunakan.

Table 3.2 Perhitungan Autokorelasi

t	Yt	Yt-1	yt	yt2(kuadrat)	yt-1	(yt)(yt-1)
1	2220	0	-438.036	191875.287	0	0
2	2260	2220	-398.036	158432.4298	-438.036	174353.8584
3	2270	2260	-388.036	150571.7156	-398.036	154452.0727
4	2280	2270	-378.036	142911.0013	-388.036	146691.3584
5	2280	2280	-378.036	142911.0013	-378.036	142911.0013
⋮	⋮	⋮	⋮	⋮	⋮	⋮
55	3240	3130	581.9643	338682.4298	471.9643	274666.3584
56	3120	3240	461.9643	213411.0013	581.9643	268846.7156
Σ	148850			6060483.9	-462	5708641
Jmltd	56					
Mean	2658.035714					
	2658.035714					

Dari tabel diatas kemudian hitung nilai *r_k* data yang berautokorelasi dengan mencari nilai *lags* menggunakan rumus diibawah ini:

$$r_k = \frac{\sum_{i=1}^n (y_i - \bar{y})(y_{i-k} - \bar{y})}{\left[\sum_{i=1}^n (y_i - \bar{y})^2 \right]}$$

Selanjutnya kita hitung *r_k* untuk *yt-2* dan seterusnya. Kemudian kita ambil *lags* dengan nilai *r_k* >= **0.6** untuk dijadikan *input* pada pelatihan JST (*Neural Network*).

Table 3.3 Uji *Lags Signifikan*

Lags	MSE	MAPE%
0.4	0.014	2.828
0.5	0.021	2.860
0.6	0.022	2.772
0.7	0.039	2.919
0.8	0.013	5.549

Pada tabel 3.3 merupakan hasil uji coba oleh sistem pada masing-masing *lags* dengan *MSE* dan *MAPE* yang bervariasi. Dari hasil uji coba diatas, *lags signifikan* terbaik adalah 0.6.

Table 3.4 Hasil *lags Signifikan*

Lags signifikan (0,6)	
r 1	0.941944711
r 2	0.87865646
r 3	0.828023374
r 4	0.782538098
r 5	0.722662174
r 6	0.673791973
r 7	0.620361197
r 8	0.567811421
r 9	0.51980837
r 10	0.46516865

Setelah data diproses *autokorelasi* kemudian data di normalisasi sehingga menjadi data dengan *range* 0-1

seperti tabel 3.6. Dua *lags* yang akan menjadi inputan untuk proses menuju metode *backpropagation* yaitu *lags* 6 dan *lags* 7.

Table 3.5 Data menurut Lags yang diambil

lags7	data ke	lags-6	data ke	target	data ke
3130	49	3150	50	3120	56
3040	48	3130	49	3240	55
3070	47	3040	48	3130	54
3080	46	3070	47	3070	53
3070	45	3080	46	3230	52

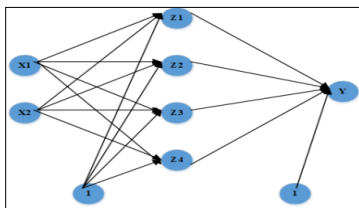
Pada gambar 3.6 adalah tabel hasil dari normalisasi data. Rumus yang digunakan dalam proses normalisasi ini adalah:

Table 3.6 Normalisasi Data

Lags6	Lags7	Target
1	1	0.8824
0.9785	0.9011	1
0.8817	0.9341	0.8922
0.914	0.9451	0.8333

$$f(x) = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}}$$

Arsitektur *backpropagation* pada data yang diuji coba untuk proses perhitungan manual menggunakan metode *Backpropagation* sebagai berikut:



Gambar 3.7 Arsitektur Backpropagation

Gambar 3.7 merupakan arsitektur *backpropagation*, yang di sesuaikan dengan hasil proses *autokorelasi*. Data yang di jadikan *input* ada 2, dengan 4 *hidden layer* dan 1 *output*.

Proses pelatihan *Neural Network* menggunakan metode *backpropagation*, yang akan menghasilkan nilai bobot akhir yang nantinya akan digunakan untuk proses uji coba. tentukan terlebih dahulu parameter pelatihan. *Variabel* yang digunakan dalam pelatihan ini yaitu: *input layer*, *hidden layer*, *output layer*, *Epo* maksimum, *learning rate*(α), *momentum* (δ) dan *error* toleransi.

Table 3.7 Parameter yang digunakan

Jumlah Neron pada <i>input layer</i>	2
Jumlah Neron pada <i>hidden layer</i>	4
Jumlah Neuron pada <i>output layer</i>	1
Momentum	0.02
Learning rate (α)	0.08
Target Error	0.001
Maksimum Epo	1000

Pada tabel 3.7 parameter yang digunakan yaitu 2 *input* yang dihasilkan dari proses *autokorelasi*, 4

hidden layer, 1 *output*, dengan *momentum* 0.02, *learning rate* 0.08, *target error* 0.001 dan maksimal *epoch* 1000. Berikut langkah-langkah perhitungan manual menggunakan metode *Backpropagation*:

1. Definisikan pola masukan dan targetnya.

Pola masukan bisa kita definisikan sebagai data *x* hari dan targetnya adalah data hari berikutnya.

Table 3.8 Inputan harga saham

x1	x2	Target
1	1	0.8824
0.9785	0.9011	1
0.8817	0.9341	0.8922
0.914	0.9451	0.8333

2. Inisialisasi bobot dengan nilai acak

Pemilihan bobot awal sangat mempengaruhi jaringan saraf dalam mencapai minimum terhadap nilai *error*,serta cepat tidaknya proses pelatihan. Biasanya bobot awal di inisialisasi secara *random* antar -0 sampai 1.

- Bobot awal input ke *hidden*

V =

Table 3.9 bobot V1-n

0.314	0.413	-0.221	0.464
0.405	0.132	0.046	-0.342

- Bobot awal bias ke *hidden*

V0 =

Table 3.10 bobot bias V0-n

-0.373	-0.402	0.457	0.470
--------	--------	-------	-------

- Bobot awal *hidden* ke *output*

W =

Table 3.11 bobot W1-n

0.457
-0.014
0.300
-0.358

- Bobot awal bias ke *output*

W0 = -0.078

Tentukan *iterasi error* yang diinginkan atau membatasi iterasi.

3. Setiap pasangan elemen akan dilakukan pembelajaran

- Fase pertama : Propagasi maju

Sinyal masukan dipropagasikan ke layar tersembunyi menggunakan fungsi *aktivasi* yang ditentukan. Keluaran dari tiap unit tersembunyi akan menghasilkan keluaran. Nilai keluaran dibandingkan dengan target yang harus dicapai. Selisih ($t_k - t_k$) adalah kesalahan yang terjadi. Jika kesalahan lebih kecil dari toleransi, maka iterasi di hentikan. Dan jika kesalahan masih lebih besar maka bobot diproses ulang..

- Fase kedua : Propagasi Balik

Berdasarkan kesalahan ($t_k - t_k$) dihitung faktor δ_k untuk modifikasi bobot ke k y_k ke semua unit

tersembunyi hingga sampai ke unit masukan dihitung.

- Fase ketiga : Perubahan bobot
Setelah semua faktor δ_k dihitung semua bobot dimodifikasi bersamaan.

Pembelajaran/pelatihan (*trining*):

- Epoch ke1:

Data ke 1

- ✓ Operasi pada *hidden layer* (langkah 4)

- Penjumlahan terbobot:

$$Y_j = f(V_{0j} + \sum X_i V_{ij})_{i=1}^n$$

$$Y_1 = v_{01} + v_{11} * x_{11}$$

$$= -0.373 + 0.314 * 1$$

$$= 0.593$$

$$Y_2 = v_{02} + v_{12} * x_{11}$$

$$= -0.402 + 0.413 * 1$$

$$= 0.011$$

$$Y_3 = v_{03} + v_{13} * x_{11}$$

$$= 0.457 + (-0.221) * 1$$

$$= 0.236$$

$$Y_4 = v_{04} + v_{14} * x_{11}$$

$$= 0.470 + 0.464 * 1$$

$$= 0.934$$

- Pengaktifan aktivasi:

$$z(x) = \frac{1}{1 + e^{-y_j}}$$

$$zin_1 = \frac{1}{1 + e^{-0.593}} = 0.5860$$

$$zin_2 = \frac{1}{1 + e^{-0.011}} = 0.5357$$

$$zin_3 = \frac{1}{1 + e^{-0.236}} = 0.5702$$

$$zin_4 = \frac{1}{1 + e^{-0.934}} = 0.6440$$

- ✓ Operasi pada *output layer* (Langkah 5) :

$$Z1 = w_0 + w_1 * z_1 + w_2 * z_2 + w_3 * z_3 + w_4 * z_4$$

$$= -0.078 + 0.457 * 0.5860 + (-0.014) * 0.5357 +$$

$$0.300 * 0.5702 +$$

$$(-0.358) * 0.6440$$

$$= 0.1224$$

- Pengaktifan:

$$Z = \frac{1}{1 + e^{-0.1224}} = 0.5305$$

Hitung error:

$$\delta_k = (t_k - z_k) f'(W_{0k} + \sum_{j=1}^p Y_j W_{jk})$$

$$\Delta W_{jk} = \alpha \delta_k Y_j$$

$$\Delta W_{0k} = \alpha \delta_k$$

$$\text{Error} = 0 - 0.6336 = -0.6336$$

$$\text{Jumlah kuadrat error} = (-0.6336)^2 = 0.4014$$

- ✓ Propagasi Error Langkah 6 :

$$\delta = (T_1 - Z) * 1 * 1 - 1$$

$$\frac{1}{1 + e^{-y_j}}$$

$$\frac{1}{1 + e^{-y_j}}$$

$$\delta = (0.8824 - 0.5305) * \left[\frac{1}{1 + e^{-0.5305}} \right] * 1 - \left[\frac{1}{1 + e^{-0.5305}} \right]$$

$$\delta = 0.3519 * 0.5305 * 0.4695 = 0.087618$$

$$\Delta w_1 = \alpha * \delta * zin_1 * mom$$

$$= 0.08 * 0.0876 * 0.5860 * 0.02$$

$$= 0.00008215$$

$$\Delta w_2 = \alpha * \delta * zin_2 * mom$$

$$= 0.08 * 0.0876 * 0.5357 * 0.02$$

$$= 0.00007511$$

$$\Delta w_3 = \alpha * \delta * zin_3 * mom$$

$$= 0.08 * 0.0876 * 0.5702 * 0.02$$

$$= 0.00007994$$

$$\Delta w_4 = \alpha * \delta * zin_4 * mom$$

$$= 0.08 * 0.0876 * 0.6440 * 0.02$$

$$= 0.00009$$

$$\Delta w_0 = \alpha * \delta * mom$$

$$= 0.08 * 0.0876 * 0.5673 * 0.02$$

$$= 0.00014$$

$$\delta_{in_1} = \delta * w_{lama}$$

$$\delta_{in_1} = \delta * w_1 = 0.087618 * 0.457 = 0.040056$$

$$\delta_{in_2} = \delta * w_2 = 0.087618 * -0.014 = -0.00128$$

$$\delta_{in_3} = \delta * w_3 = 0.087618 * 0.300 = 0.02631$$

$$\delta_{in_4} = \delta * w_4 = 0.087618 * -0.358 = -0.03138$$

- Langkah 7 :

$$\delta_{11} = \delta_{in_1} * 1 * \left[\frac{1}{1 + e^{-zin_1}} \right] * -1 * \left[\frac{1}{1 + e^{-zin_1}} \right]$$

$$\delta_1 = 0.040056 * \left[\frac{1}{1 + e^{-0.5860}} \right] * 1 - \left[\frac{1}{1 + e^{-0.7606}} \right]$$

$$= 0.009718$$

$$\delta_2 = -0.00128 * \left[\frac{1}{1 + e^{-0.5357}} \right] * 1 - \left[\frac{1}{1 + e^{-0.5357}} \right]$$

$$= -0.00032$$

$$\delta_3 = 0.02631 * \left[\frac{1}{1 + e^{-0.5702}} \right] * 1 - \left[\frac{1}{1 + e^{-0.5702}} \right]$$

$$= 0.006448$$

$$\delta_4 = -0.03138 * \left[\frac{1}{1 + e^{-0.6440}} \right] * 1 - \left[\frac{1}{1 + e^{-0.6440}} \right]$$

$$= -0.00719$$

$$\Delta v_{11} = \alpha * \delta_1 * x_{11} * mom = 0.08 * 0.009718 * 1 * 0.02$$

$$= 0.000015548$$

$$\Delta v_{12} = \alpha * \delta_2 * x_{12} * mom = 0.08 * (-0.00032) * 1 * 0.02$$

$$= -0.00000510$$

$$\Delta v_{13} = \alpha * \delta_3 * x_{13} * mom = 0.08 * 0.006448 * 1 * 0.02$$

$$= 0.000010316$$

$$\Delta v_{14} = \alpha * \delta_4 * x_{14} * mom = 0.08 * (-0.00719) * 1 * 0.02$$

$$= -0.000011509$$

$$\begin{aligned}\Delta v_{01} &= \alpha * \delta_1 * mom = 0.08 * (-0.00032) * 0.02 = 0.0000155 \\ \Delta v_{02} &= \alpha * \delta_2 * mom = 0.08 * (-0.00032) * 0.02 = -0.0000005 \\ \Delta v_{03} &= \alpha * \delta_3 * mom = 0.08 * (0.006448) * 0.02 = 0.0000103 \\ \Delta v_{04} &= \alpha * \delta_4 * mom = 0.08 * (-0.00719) * 0.02 = -0.0000115\end{aligned}$$

✓ Menentukan bobot baru (Langkah 8) :

$$\begin{aligned}V_{11baru} &= V_{11lama} + \Delta v_{11} = 0.1231 + -0.000000164 = 0.314739 \\ V_{12baru} &= V_{12lama} + \Delta v_{12} = 0.1023 + -0.000000167 = 0.413375 \\ V_{13baru} &= V_{13lama} + \Delta v_{13} = 0.1063 + -0.000000045 = -0.22149 \\ V_{14baru} &= V_{14lama} + \Delta v_{14} = 0.1793 + -0.000000445 = 0.464877\end{aligned}$$

$$V_{01baru} = v_{01lama} + \Delta v_{01} = 0.1316 + -0.000000211 = -0.37299$$

$$\begin{aligned}V_{02baru} &= v_{02lama} + \Delta v_{02} = 0.1368 + -0.000000214 = -0.40246 \\ V_{03baru} &= v_{03lama} + \Delta v_{03} = 0.1462 + -0.000000058 = 0.457517 \\ V_{04baru} &= v_{04lama} + \Delta v_{04} = 0.1472 + -0.000000572 = 0.470581\end{aligned}$$

$$\begin{aligned}W_{1baru} &= W_{1lama} + \Delta w_1 = 0.1093 + -0.000008336 = 0.457249 \\ W_{2baru} &= W_{2lama} + \Delta w_2 = 0.1106 + -0.000008383 = -0.01455 \\ W_{3baru} &= W_{3lama} + \Delta w_3 = 0.0219 + -0.00000832 = 0.30036 \\ W_{4baru} &= W_{4lama} + \Delta w_4 = 0.2102 + -0.000006217 = -0.35802\end{aligned}$$

$$W_{0baru} = W_{0lama} + \Delta w_0 = 0.0405 + -0.00001096 = -0.0781$$

Pada data ke-2 juga dilakukan operasi-operasi yang sama dengan menggunakan bobot-bobot akhir hasil pengolahan data pertama ini sebagai bobot-bobot awalnya. Proses ini dilakukan berulang sampai pada maksimum epoch (1000) atau kuadrat error < target error (0.001).

Pada proses pelatihan, *output* yang dihasilkan oleh jaringan berkisar antara 0 sampai dengan 1 sehingga perlu dilakukan denormalisasi yang berguna untuk mengkonversikan kembali hasil *output* jaringan menjadi nilai saham normal dengan menggunakan rumus

$$X_i = y(X_{\max} - X_{\min}) + X_{\min}$$

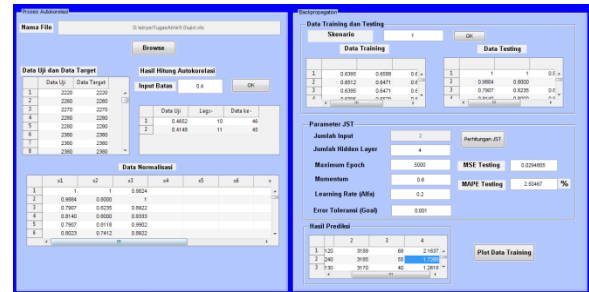
Setelah itu akan dilakukan perbandingan antara data sebenarnya dengan data hasil prediksi, sehingga dapat dihitung *error* atau prosentase errornya dan akurasi.

BAB IV

IMPLEMENTASI SISTEM

4.1 Graphical User Interface (GUI)

Pada penelitian ini dibuatkan suatu *graphical user interface* (GUI) yang berisi komponen – komponen sederhana yang akan mempermudah dalam mengetahui prediksi harga saham, pada pembuatan GUI pada sistem ini, data *training* akan di uji dengan 1 data *testing*, jadi untuk tampilan pada sistem ini hanya bisa melihat dan mengklasifikasikan 1 prediksi harga saham. pada sistem ini terdapat 2 tampilan GUI, seperti pada gambar 4.1, dan gambar 4.2



Gambar 4.1 Tampilan Awal Sistem

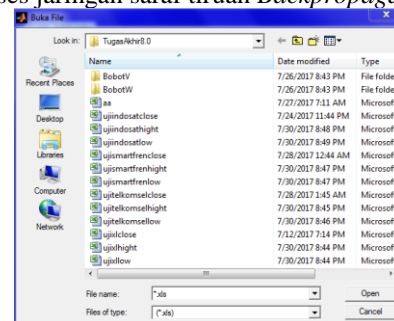
Gambar 4.1 merupakan tampilan awal pada system ini. Pada GUI tersebut terdapat beberapa komponen-komponen diantaranya, *table*, *2axes*, *4text box*, dan *push button*.

Pada komponen *tabel*, berfungsi untuk menampilkan hasil data harga saham sesuai proses yang ditentukan.

Pada komponen *axes* terdapat 2 *axes*, yang memiliki fungsi untuk menampilkan grafik. Dimana grafik tersebut menampilkan besar error setiap iterasi sesuai dengan skenario yang dipilih.

Untuk komponen *text box*, berfungsi untuk menampilkan workspace letak folder dan hasil dari klasifikasi.

Komponen selanjutnya ada *push button*. Pada sistem ini terdapat 4 *push button*. *Push button* yang pertama digunakan untuk mencari data yang akan di test dan di klasifikasi. *Push button* yang dua digunakan untuk menentukan lags signifikan pada data sudah di autokorelasi dan di klasifikasi. *Push button* yang ketiga berfungsi untuk melakukan proses menentukan *training testing*. *Push button* yang empat digunakan untuk memproses jaringan saraf tiruan *Backpropagation*.



Gambar 4.2 File Selector

Gambar 4.2 menjelaskan mengenai tombol browse. Ketika tombol browse di tekan, maka akan menampilkan seperti gambar 4.2 yang berfungsi untuk memanggil file selektor yang berisi data harga saham dalam bentuk *excel* untuk di uji. Yang nantinya akan di muncul di *table*.



Gambar 4.3 Prediksi harga saham

Gambar 4.3 menjelaskan mengenai proses prediksi untuk periode kedepan. Yang mengacu pada *lags signifikan* dan bobot hasil pelatihan *Backpropagation* yang terbaik.

Tabel 4.35. Hasil Prediksi Menggunakan Parameter Terbaik

No	Tbk	Lags	Skenario	PARAMETER						MSE
				Input	Hidden	Epoch	Mom	Lrate	Error	
1	XL Axiata									
	Data Low	0.6	3	2	10	10000	0.25	0.5	0.001	0.151
	Data High	0.7	3	2	10	10000	0.8	0.2	0.001	0.112
2	Indosat									
	Data Low	0.6	2	1	4	5000	0.01	0.01	0.001	0.057
	Data High	0.6	2	1	8	10000	0.8	0.5	0.001	0.019
3	Telkomsel									
	Data Low	0.8	1	1	8	10000	0.05	0.25	0.1	0.040
	Data High	0.7	2	1	8	10000	0.8	0.25	0.001	0.020
4	Smartfren									
	Data Low	0.8	3	1	8	10000	0.25	0.2	0.001	0.002
	Data High	0.8	3	1	8	10000	0.01	0.25	0.001	0.022
	Data Close	0.6	2	1	8	5000	0.08	0.5	0.001	0.063

Pada tabel 4.34 adalah hasil ringkasan uji coba pada masing-masing *sector telekomunikasi*. Perusahaan XL Axiata pada data *low* menghasilkan nilai akurasi sebesar 98.5%. pada data *high* menghasilkan nilai akurasi sebesar 98,3% dan pada data *close* menghasilkan nilai akurasi sebesar 98.1%.

Perusahaan Indosat pada data *low* menghasilkan nilai akurasi sebesar 98.96%. pada data *high* menghasilkan nilai akurasi sebesar 98.46% dan pada data *close* menghasilkan nilai akurasi sebesar 98.74%

Perusahaan Telkomsel pada data *low* menghasilkan nilai akurasi sebesar 99.37%. pada data *high* menghasilkan nilai akurasi sebesar 99.41% dan pada data *close* menghasilkan nilai akurasi sebesar 99.31%

Perusahaan Smartfren pada data *low* menghasilkan nilai akurasi sebesar 98.71%. pada data *high* menghasilkan nilai akurasi sebesar 99.12% dan pada data *close* menghasilkan nilai akurasi sebesar 97.13% masing-masing dengan *lags*, skenario dan parameter yang bervariasi pada tabel 4.34.

Tabel 4.35. Hasil Prediksi Menggunakan Parameter Terbaik

Perusahaan Tbk	Input Data Sebelumnya	Tanggal Prediksi	Data Aktual	Hasil Prediksi
XL Axiata				
Data Low	(6, 7)3050, 3120	3/1/2017	2970	3099
Data High	(4, 5)3250, 3150	3/1/2017	3000	3142
Data Close	(4, 5)3240, 3130	3/1/2017	2940	2942
Indosat				
Data Low	(2) 6975	4/6/2017	6800	6832
Data High	(2) 7100	4/6/2017	7000	7040
Data Close	(1) 7000	4/6/2017	6975	6730
Telkomsel				
Data Low	(1) 4030	3/20/2017	4050	3982
Data High	(1) 4140	3/20/2017	4110	4082
Data Close	(1) 4110	3/20/2017	4100	4032
Smartfren				
Data Low	(1) 53	4/26/2017	51	53
Data High	(1) 53	4/26/2017	53	54
Data Close	(3) 53	4/26/2017	52	53

Pada tabel 4.35 merupakan hasil prediksi menggunakan *lags*, scenario dan parameter terbaik yang sudah dilakukan uji coba. Input data sebelumnya diperoleh dari *lags signifikan* pada tabel ringkasan. Misal *lags signifikan* 0.6 berada pada *lags* 6 dan *lags* 7 maksudnya data ke 6 dan ke 7 yang diambil sebelum data atau tanggal yang ingin diprediksi.

BAB V PENUTUP

5.1. Kesimpulan

Berdasarkan penelitian dan pengujian sistem yang telah dilakukan, maka disimpulkan bahwa:

- Hasil uji coba yang dilakukan pada data harga saham menggunakan *autokorelasi* nilai *error* yang dihasilkan lebih akurat dibandingkan peramalan tanpa menggunakan *autokorelasi*, *MAPE* yang dihasilkan sebesar 0.14%
- Dari hasil uji coba yang dilakukan diperoleh hasil prediksi mendekati angka aktual dengan rata-rata akurasi sebesar 98% dengan parameter yang berbeda-beda dari setiap perusahaan.

5.2. Saran

Berikut ini merupakan saran yang diajukan untuk penelitian

yang selanjutnya :

- Penelitian ini digunakan untuk penelitian jangka pendek. Oleh karena itu untuk penelitian selanjutnya diharapkan agar dapat meramalkan harga saham dalam jangka panjang menggunakan jaringan syaraf tiruan.
- Perlu penelitian dalam metode menentukan bobot awal untuk mempercepat proses pembelajaran apakah memiliki pengaruh yang signifikan terhadap hasil peramalan

DAFTAR PUSTAKA

- [1] Widoatmodjo,S.2009. *Pasar Modal Indonesia*.Jakarta: Ghalia Indonesia.
- [2] Makridakis, S dkk. 1999. *Metode dan Aplikasi Peramalan Jilid 1*. Binarupa Aksara : Jakarta
- [3] Anugerah, PSW. 2007. *Perbandingan Jaringan Syaraf Tiruan Backpropagation Dan Metode Deret Berkala Box-Jenkins (Arima) Sebagai Metode Peramalan Curah Hujan*. Semarang. Tugas Akhir Universitas negeri Semarang.
- [4] Rufiyanti, D. E. 2015. *Implementasi Jaringan Syaraf Tiruan Backpropagation Dengan Input Model Arima Untuk Peramalan Harga Saham*. Skripsi. Fakultas Matematika Dan Ilmu Pengetahuan Universitas Negeri Semarang. Semarang
- [5] Setiawan, W. 2008. “Prediksi Harga Saham Menggunakan Jaringan Syaraf Tiruan Multilayer Feedforward Network Dengan Algoritma Backpropagation”. *Dalam Konferensi Nasional Sistem dan Informatika*. Bali.
- [6] Resty,N. I. 2010. *Sistem Pendukung Keputusan Penentuan Harga Jual Perumahan dengan Menggunakan Metode Backpropagation Jaringan Syaraf Tiruan Studi Kasus PT. Propindo Wira Utama*. Bangkalan. Teknik Informatika : Universitas Trunojoyo.
- [7] Maharani, W. 2009. “ *Klasifikasi Data Menggunakan JST Backpropagation Momentum Dengan Adaptive Learning Rate*“. Seminar Nasional Informatika 2009 (SemnasIF 2009) ISSN: 1979-2328. Yogyakarta 2009.
- [8] Hendranata, A. 2003. *ARIMA (Autoregressive Moving Average)*, Manajemen Keuangan Sektor Publik FEUI.
- [9] Jek Siang, N. 2004. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta : ANDI.
- [10] Suprianto,E, T. 2004. “*Penerapan Jaringan Syaraf Tiruan Untuk Memprediksi Harga Saham*”. Bandung. Sekolah tinggi Teknik Informatika dan Ilmu Komputer.
- [11] Enireddy,V, dkk. 2010. “*Prediction of Rainfall Using Backpropagation Neural Network Model*”. (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, 1119-1121.
- [12] Muhammad,M. (2010). “*Perbandingan Jaringan Saraf tiruan Backpropagation dan Metode ARIMA (box-Jenkins) Sebagai metode Peramalan Kurs Rupiah Terhadap Dolar Amerika Serikat*”.Sumatera Utara.Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam.
- [13] Anggraeni, Dyta. *Klasifikasi Topik Menggunakan Machine Learning*.Universitas Indonesia.