

BAB III

ANALISA DAN PERANCANGAN SISTEM

Pada bab ini dijelaskan proses perancangan sistem dengan *flowchart* dan mengimplementasikan metode *Backpropagation* dalam memprediksi harga saham meliputi tahapan tiap proses.

3.1 Analisa sistem

Analisa sistem dilakukan agar sistem prediksi harga saham dapat sesuai dengan keadaan nyata.

Sistem yang dirancang pada skripsi ini adalah sistem prediksi harga saham menggunakan metode *backpropagation* dengan *momentum*. Proses awal yang dilakukan pada sistem ini yakni, menentukan data inputan dan data target yang ingin dilatih dan diuji,

Proses perhitungan Autokorelasi dilakukan pertama kali sehingga mendapatkan *lags* terbaik, dan data akan otomatis ternormalisasi. Kemudian dari data yang diperoleh dibagi menjadi data trining dan data testing dengan menggunakan *Cross Validation*, dan proses selanjutnya metode *Backpropagation* dilakukan dengan menginputkan parameter-parameter seperti *hidden layer*, *momentum*, *learning rate*, *epoch max* dan *toleransi error*. Sehingga diperoleh bobot-bobot baru yang terbaik dari nilai *MSE* dan *MAPE*. Dari bobot terbaik tersebut di simpan dan di gunakan untuk *testing*. Proses akhirnya ialah *output* nilai dari *JST* yang akan di *denormalisasi* sehingga data kembali ke nilai awal.

3.1.1 Analisa kebutuhan

Kebutuhan-kebutuhan yang diperlukan untuk membangun sistem prediksi harga saham ini meliputi kebutuhan *interface*, kebutuhan data, kebutuhan *funksional*, dan kebutuhan sistem secara lebih rinci akan dijelaskan berikut ini:

a. Kebutuhan Interface

Sistem yang dibangun memiliki tampilan atau *interface* yang memudahkan pengguna dalam mengoperasikan proses pada sistem secara keseluruhan. Kebutuhan *interface* dalam sistem ini adalah sebagai berikut:

1. Halaman Proses Autokorelasi : terdapat tombol *browse*, dimana *user* dapat mengambil data dalam bentuk *excel* dari *computer*.

- Inputan batas *lags* dimana *user* menginputkan batas *lags* terbaik hasil dari proses *autokorelasi*.
2. Halaman Proses *Backpropagation* : terdapat form data *training* dan data *testing*, inputan untuk *scenario* 1 samapi 3. Pada form parameter JST terdapat beberapa *input*, tombol perhitungan JST untuk melakukan proses pelatihan menggunakan metode *backpropagation*. Terdapat juga tombol plot data *training* .
 3. Halaman Prediksi : terdapat inputan untuk memprediksi harga saham pada tanggal tertentu.

b. Kebutuhan Data

Data yang dikelola oleh sistem ini adalah :

1. *Input* data dari *browse* berupa nilai harga saham yang diperoleh dari *www.IDX.com* , seperti data *low*, *high* dan *close* pada masing-masing perusahaan *sector Telekomunikasi*.
2. Data disimpan dalam bentuk *excel*

c. Kebutuhan Fungsional

Sistem ini memiliki beberapa fungsi, antara lain :

1. Sistem dapat menampilkan data input yang telah di-*input*-kan oleh user.
2. Sistem yang dibangun dapat melakukan prediksi beserta nilai error yang dihasilkan menggunakan metode *backpropagation*.
3. Sistem dapat memberi *output* berupa prediksi yang akan dibandingkan dengan data aktual.

3.2 Desain Rancangan

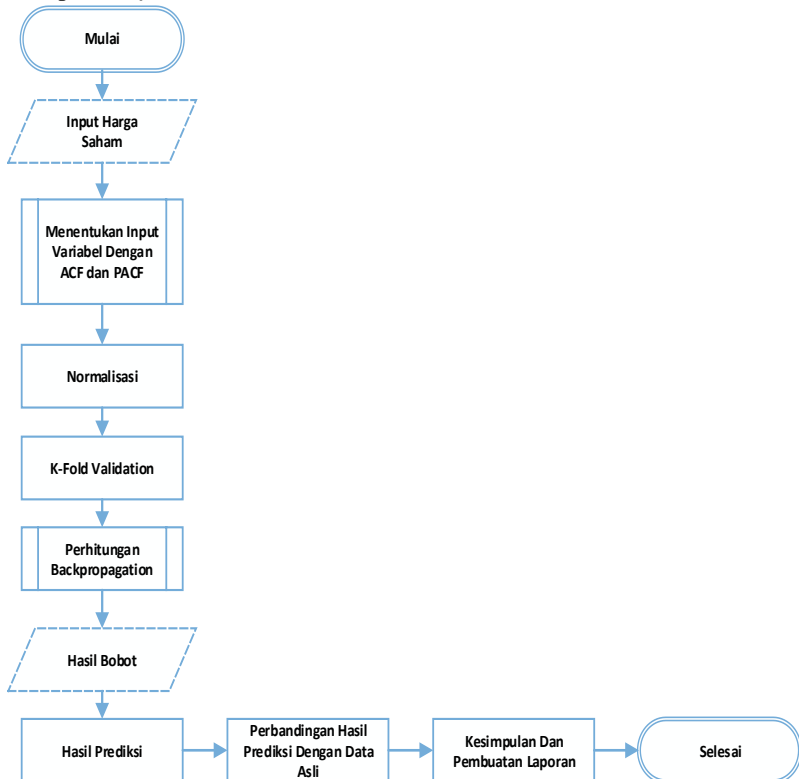
Desain rancangan sistem dibuat agar sistem yang dibangun akan lebih mudah untuk menganalisa jika terjadi kesalahan ataupun ketidaksesuaian dengan kondisi yang diharapkan.

1.2.1 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 :

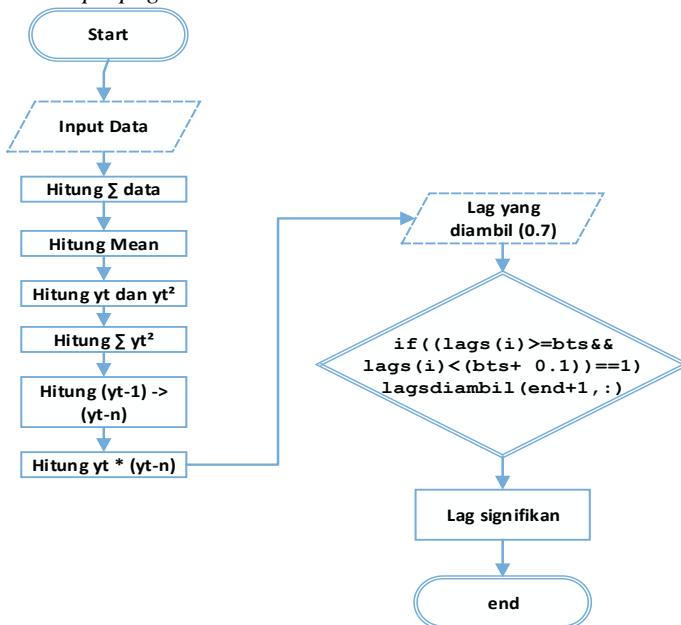
- 1. Mulai**
- 2. Masukan Inputan**

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*.



Gambar 3.2 Flowchart Sistem Autokorelasi

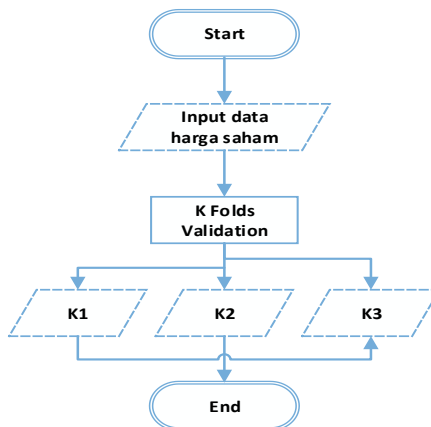
Pada gambar 3.2 merupakan alur menentukan nilai inputan yang berpengaruh menggunakan *autokorelasi* dimana langkah pertama yaitu menyiapkan data yang akan di autokorelasi dan di jumlah. Kemudian hitung nilai *mean*, menentukan nilai *yt*, *yt* kuadrat, menjumlahkan hasil kuadrat dan menghitung *lags*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

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

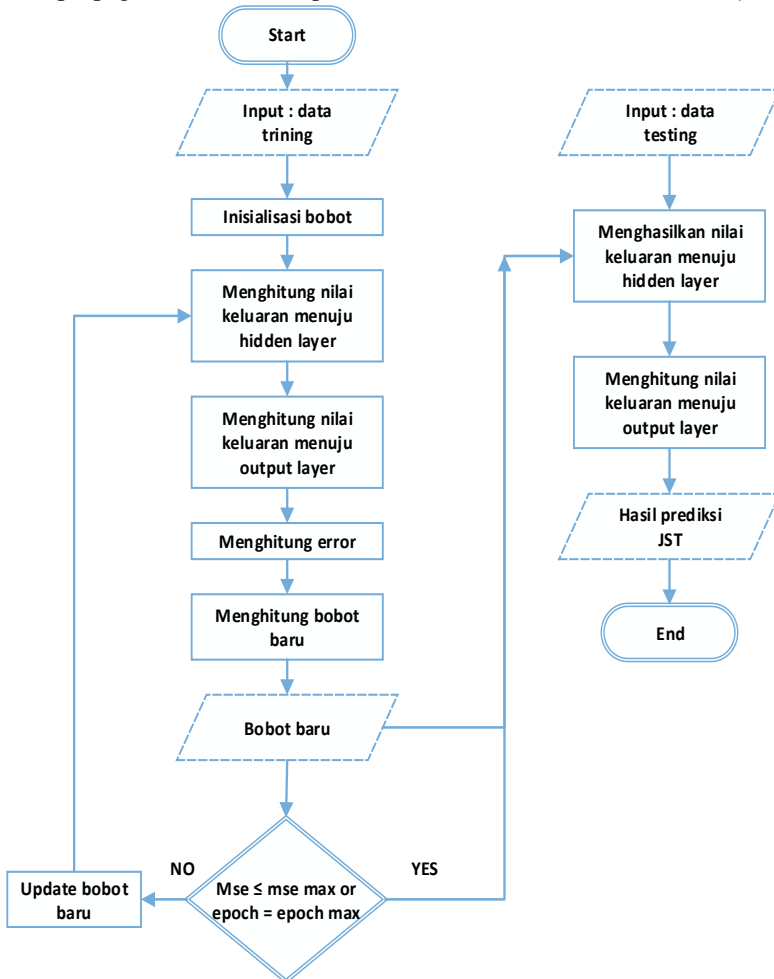


Gambar 3.3 Flowchart K Folds Validation

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*

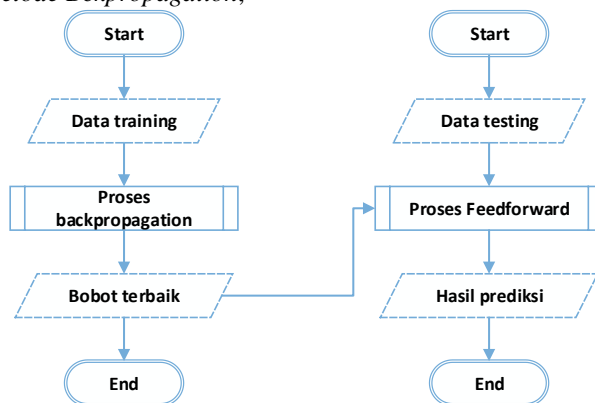
Di bawah ini merupakan *Flowchart* sistem proses menggunakan metode *Backpropagation*. Pada gambar 3.4 langkah di mulai dari hasil output dari proses autokorelasi yang akan diproses menggunakan *Backpropagation* untuk memperbarui bobot dan dikirim ke *hidden layer*.



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 *training* 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.

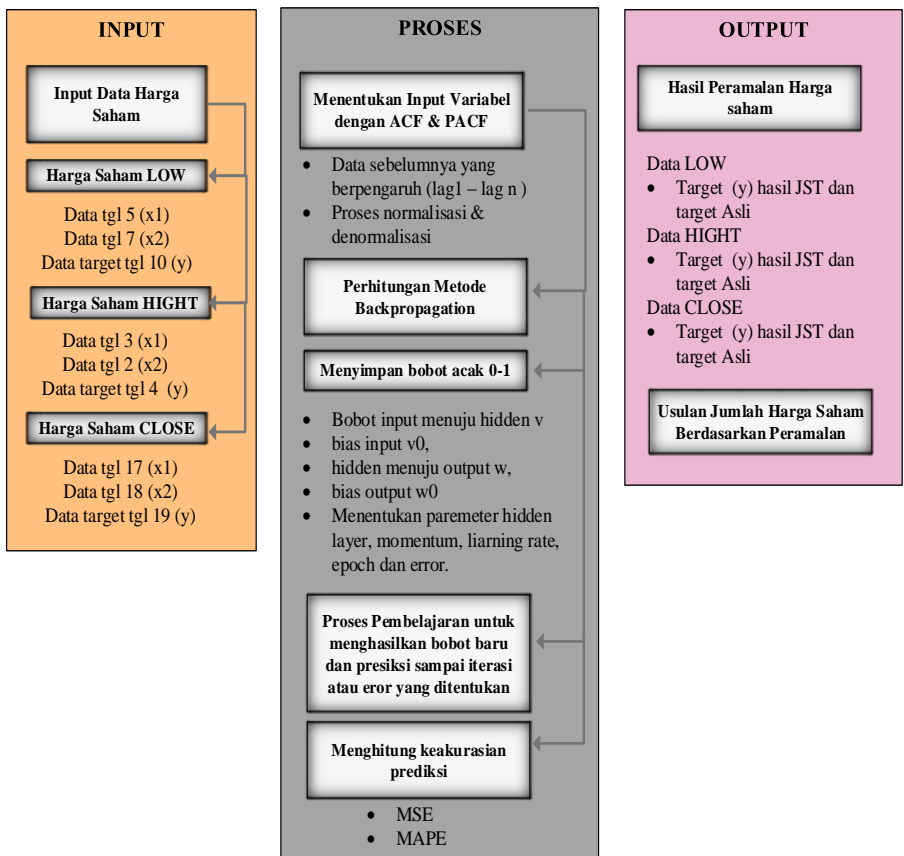
7. Menghasilkan output

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

8. Selesai

1.3 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*.



Gambar 3.6 Arsitektur Sistem

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.

1.4 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 february 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	<i>CLOSE</i>
1	Desember	2220
2		2260
3		2270
4		2280

-	Januari	2280
-		2320
-		2560
-		2650
-		2640
53	Februari	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
Jmltdt	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:

$$r_k = \frac{\left[\sum_{t=1}^n (Y_t - \bar{Y}_t)(Y_{t-k} - \bar{Y}_t) \right]}{\left[\sum_{t=1}^n (Y_t - \bar{Y}_t)^2 \right]} \dots\dots\dots (2.19)$$

Selanjutnya di hitung r_k untuk $yt-2$ dan seterusnya. Kemudian kita ambil *lags* dengan nilai $r_k \geq 0.6$ 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

lags-7	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

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

Selanjutnya proses *K Folds Validation* menghasikan 3 dataset k1, k2, dan k3. Data *training* dan data *testing* diperoleh dengan menyusun k1, k2, dan k3. Untuk skenario pertama k2 dan k3 menjadi data *training* sedangkan k1 menjadi data *testing*. Untuk skenario kedua k1 dan k3 menjadi data *training* sedangkan k2 menjadi data *testing*. Dan untuk skenario ketiga k1 dan k2 menjadi data *training* sedangkan k3 menjadi data *testing*. Berikut adalah tabel hasil *K Folds Validation* dan susunan data tiap skenario.

Tebel 3.7 Skenario Uji Coba

	Data Training	Data Testing
Skenario 1	k2, k3	k1
Skenario 2	k1, k3	k2
Skenario 3	K1, k2	k3

1. Skenario 1

Table 3.8 Data training dan data testing skenario 1

	Data	
Data Training (k2, k3)	16	16
Data Testing (k1)	17	

2. Skenario 2

Table 3.9 Data training dan data testing skenario 2

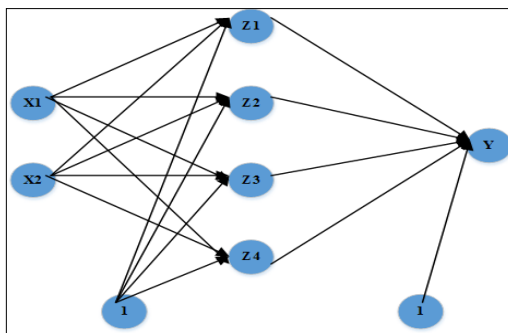
	Data	
Data Training (k1, k3)	17	16
Data Testing (k2)	16	

3. Skenario 3

Table 3.10 Data *training* dan data *testing* skenario 3

	Data	
<i>Data Training</i> (k1, k2)	17	16
<i>Data Testing</i> (k3)	16	

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 disesuaikan dengan hasil proses *autokorelasi*. Data yang dijadikan *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 3.11 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
<i>Momentum</i>	0.02
<i>Learning rate</i> (α)	0.08
<i>Target Error</i>	0.001
<i>Maksimum Epo</i>	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 3.12 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 3.13 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 3.14 bobot bias V0-n

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

- Bobot awal *hidden* ke *output*

W=

Table 3.15 bobot W1-n

0.457
-0.014
0.300
-0.358

- Bobot awal bias ke output

$$\mathbf{W0} = -0.078$$

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:

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

$$\begin{aligned} Y_1 &= \mathbf{v0_1} + \mathbf{v_{11}} * \mathbf{x_{11}} \\ &= -0.373 + 0.314 * 1 \\ &= 0.593 \end{aligned}$$

$$\begin{aligned} Y_2 &= \mathbf{v0_2} + \mathbf{v_{12}} * \mathbf{x_{11}} \\ &= -0.402 + 0.413 * 1 \\ &= 0.011 \end{aligned}$$

$$\begin{aligned} Y_3 &= \mathbf{v0_3} + \mathbf{v_{13}} * \mathbf{x_{11}} \\ &= 0.457 + (-0.221) * 1 \\ &= 0.236 \end{aligned}$$

$$\begin{aligned}
 Y_4 &= v_{04} + v_{14} * x_{11} \\
 &= 0.470 + 0.464 * 1 \\
 &= 0.934
 \end{aligned}$$

- Pengaktifan aktivasi:

$$\begin{aligned}
 z(x) &= \frac{1}{1 + e^{-y_j}} \\
 zin_1 &= \frac{1}{1 + e^{-0.593}} = 0.5860 \\
 zin_2 &= \frac{1}{1 + e^{-0.011}} = 0.5357 \\
 zin_3 &= \frac{1}{1 + e^{-0.236}} = 0.5702 \\
 zin_4 &= \frac{1}{1 + e^{-0.934}} = 0.6440
 \end{aligned}$$

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

$$\begin{aligned}
 Z1 &= w_0 + w_1 * z_1 + w_2 * z_2 + w_3 * z_3 + w_4 * z_4 \\
 &= -0.078 + 0.457 * 0.5860 + (-0.014) * 0.5357 + 0.300 * 0.5702 + \\
 &\quad (-0.358) * 0.6440 \\
 &= 0.1224
 \end{aligned}$$

- 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 :**

$$\delta = (T_1 - Z) * \frac{1}{1 + e^{-y_j}} * 1 - \frac{1}{1 + e^{-y_j}}$$

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

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

$$\begin{aligned}\Delta w_1 &= \alpha * \delta * z_{in_1} * mom \\ &= 0.08 * 0.0876 * 0.5860 * 0.02 \\ &= 0.00008215 \\ \Delta w_2 &= \alpha * \delta * z_{in_2} * mom \\ &= 0.08 * 0.0876 * 0.5357 * 0.02 \\ &= 0.00007511 \\ \Delta w_3 &= \alpha * \delta * z_{in_3} * mom \\ &= 0.08 * 0.0876 * 0.5702 * 0.02 \\ &= 0.00007994 \\ \Delta w_4 &= \alpha * \delta * z_{in_4} * mom \\ &= 0.08 * 0.0876 * 0.6440 * 0.02 \\ &= 0.00009 \\ \Delta w_0 &= \alpha * \delta * mom \\ &= 0.08 * 0.0876 * 0.5673 * 0.02 \\ &= 0.00014\end{aligned}$$

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

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

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

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

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

• **Langkah 7 :**

$$\delta_1 = \delta_{in_1} * 1 * \left[\frac{1}{1 + e^{-z_{in_1}}} \right] * -1 \left[\frac{1}{1 + e^{-z_{in_1}}} \right]$$

$$\begin{aligned}\delta_1 &= 0.040056 * \left[\frac{1}{1 + e^{-0.5860}} \right] * 1 - \left[\frac{1}{1 + e^{-0.7606}} \right] \\ &= 0.009718\end{aligned}$$

$$\begin{aligned}\delta_2 &= -0.00128 * \left[\frac{1}{1 + e^{-0.5357}} \right] * 1 - \left[\frac{1}{1 + e^{-0.5357}} \right] \\ &= -0.00032\end{aligned}$$

$$\begin{aligned}\delta_3 &= 0.02631 * \left[\frac{1}{1 + e^{-0.5702}} \right] * 1 - \left[\frac{1}{1 + e^{-0.5702}} \right] \\ &= 0.006448\end{aligned}$$

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

$$= -0.00719$$

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

$$= 0.000015548$$

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

$$= -0.000000510$$

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

$$= 0.000010316$$

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

$$= -0.000011509$$

$$\Delta v_{01} = \alpha * \delta_1 * \text{mom} = 0.08 * (-0.00032) * 0.02 = 0.0000155$$

$$\Delta v_{02} = \alpha * \delta_2 * \text{mom} = 0.08 * (-0.00032) * 0.02 = -0.0000005$$

$$\Delta v_{03} = \alpha * \delta_3 * \text{mom} = 0.08 * (0.006448) * 0.02 = 0.0000103$$

$$\Delta v_{04} = \alpha * \delta_4 * \text{mom} = 0.08 * (-0.00719) * 0.02 = -0.0000115$$

✓ **Menentukan bobot baru (Langkah 8) :**

$$V_{11\text{baru}} = V_{11\text{lama}} + \Delta v_{11} = 0.1231 + -0.000000164 = 0.314739$$

$$V_{12\text{baru}} = V_{12\text{lama}} + \Delta v_{12} = 0.1023 + -0.000000167 = 0.413375$$

$$V_{13\text{baru}} = V_{13\text{lama}} + \Delta v_{13} = 0.1063 + -0.000000045 = -0.22149$$

$$V_{14\text{baru}} = V_{14\text{lama}} + \Delta v_{14} = 0.1793 + -0.000000445 = 0.464877$$

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

$$V_{02\text{baru}} = v_{02\text{lama}} + \Delta v_{02} = 0.1368 + -0.000000214 = -0.40246$$

$$V_{03\text{baru}} = v_{03\text{lama}} + \Delta v_{03} = 0.1462 + -0.000000058 = 0.457517$$

$$V_{04\text{baru}} = v_{04\text{lama}} + \Delta v_{04} = 0.1472 + -0.000000572 = 0.470581$$

$$W_{1\text{baru}} = W_{1\text{lama}} + \Delta w_1 = 0.1093 + -0.000008336 = 0.457249$$

$$W_{2\text{baru}} = W_{2\text{lama}} + \Delta w_2 = 0.1106 + -0.000008383 = -0.01455$$

$$W_{3\text{baru}} = W_{3\text{lama}} + \Delta w_3 = 0.0219 + -0.00000832 = 0.30036$$

$$W_{4\text{baru}} = W_{4\text{lama}} + \Delta w_4 = 0.2102 + -0.000006217 = -0.35802$$

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

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

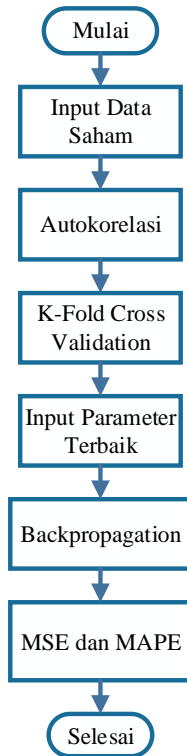
$$X_i = y(X_{\max} - X_{\min}) + X_{\min}$$

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

1.5. Rencana Pengujian

Rencana pengujian peramalan harga saham pada sector Telekomunikasi ini bertujuan untuk mengetahui seberapa akurat metode *Backpropagation* dengan *momentum* yang digunakan dalam memprediksi harga saham pada periode kedepan. Pengujian dilakukan pada data harga saham sektor Telekomunikasi yaitu XL Axiata, Indosat, Telkomsel dan Smartfren selama 3 bulan dari tanggal 1 desember sampai 24 Februari.

Hal yang akan dilakukan pada pengujian pertama kali adalah menyiapkan data saham yang kemudian di proses dengan cara *autokorelasi* dengan beberapa lags signifikan terbaik contoh 0.5 sampai 0.9. sehingga menghasilkan lag ke-n yang akan dijadikan sebagai input untuk proses selanjutnya. Dari hasil *autokorelasi*, data dibagi menjadi data *training* dan data *testing* dalam beberapa skenario menggunakan konsep *K-fold cross validation*. Berikut adalah flowchart pengujian sistem yang dibuat untuk mendapatkan hasil prediksi yang akurat.



Gambar 3.8 Flowchart Pengujian

Proses backpropagation di gunakan untuk melatih data *trining* sampai *epoch* dan *error* toleransi yang di inginkan. Dari pelatihan data *training* akan diperoleh bobot-bobot yang terbaik yang akan digunakan untuk data *testing* sehingga menghasilkan *MSE* dan *MAPE* terkecil.