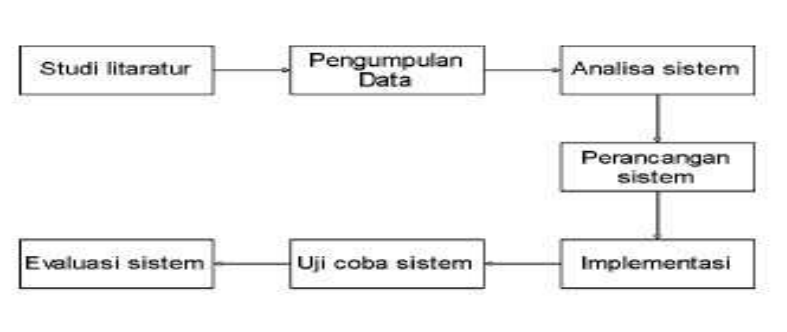
BAB III
METODOLOGI DAN PERANCANGAN SISTEM

Dalam bab ini akan dijelaskan mengenai analisis sistem dan perancangan sistem untuk peramalan harga saham menggunakan *backpropagation neural network* berbasis algoritma genetika.

Penelitian dilakukan dengan tahapan-tahapan sebagai berikut:

1. Mempelajari literatur yang terkait dengan masalah saham, jaringan syaraf tiruan *backpropagation* dan algoritma genetika.
2. Mengumpulkan data-data saham dari bursa efek.
3. Menganalisa system dan melakukan perancangan sistem menggunakan jaringan syaraf tiruan menggunakan optimalisasi algoritma genetika.
4. Mengimplementasikan system.
5. Melakukan uji coba sistem dengan memasukkan data yang berbeda dengan data training kedalam sistem.
6. Mengevaluasi hasil pengujian.

Alur penelitian yang dilakukan dapat digambarkan dalam bentuk diagram alir yang ditunjukkan pada Gambar 3.1.



Gambar 3.1 Langkah-langkah Penelitian

3.1 Studi Literatur

Untuk lebih memantapkan pengetahuan dalam merealisasikan tujuan dan pemecahan masalah penelitian ini dilakukan dengan studi literatur. Teori-teori

mengenai teori jaringan syaraf tiruan *backpropagation* dan algoritma genetika digunakan sebagai dasar penelitian ini diperoleh dari buku, jurnal, dan sumber lain.

3.2 Data Yang Digunakan

Data yang digunakan untuk penelitian ini adalah data harga saham. Data berisi 4 parameter, lima nilai saham sebagai masukan yang terdiri dari harga buka (*open price*), harga terendah (*low price*), harga tertinggi (*high price*), dan *Volume* Perdagangan (*trade volume*), serta satu parameter sebagai target yaitu harga penutupan (*close price*).

3.3 Menentukan Pola Data

Dari data yang didapat dapat disusun suatu pola / *pattern* data yang nantinya digunakan sebagai masukan dalam pelatihan jaringan syaraf tiruan.Tabel pola data ditunjukkan pada tabel 3.1.

Tabel 3.1 Pola Data

X1	X2	X3	X4	T1

Keterangan dari pola input adalah sebagai berikut:

- x1 : *open price*
- x2 : *low price*
- x3 : *high price*
- x4: trade volume
- t1: *close price*

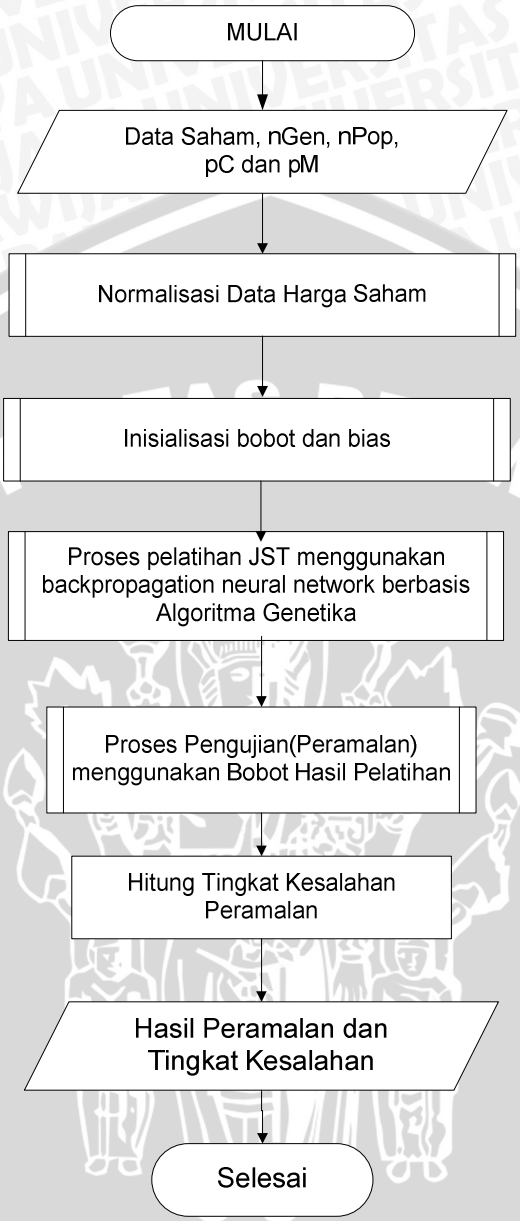
3.4 Deskripsi Umum Sistem

Sistem yang akan dibuat merupakan sistem peramalan hargasaham menggunakan jaringan syaraf tiruan *backpropagation* dengan menerapkan algoritma genetika pada proses pelatihan jaringan. Pada sistem ini diperlukan data masukan adalah data harga saham berupa *open price*, *low price*, *high price*, *close price*, dan

trade volume. Masukan lain yang digunakan adalah parameter *backpropagation* dan algoritma genetika berupa *learning rate, momentum,* jumlah generasi, ukuran populasi, probabilitas *crossover* dan probabilitas mutasi. Hasil keluaran sistem berupa peramalan *close price* saham pada hari yang sama. Sistem yang akan dibangun diharapkan agar dapat memberikan pertimbangan untuk pemain saham untuk membeli atau menjual saham yang dimiliki.

3.5 Perancangan Sistem

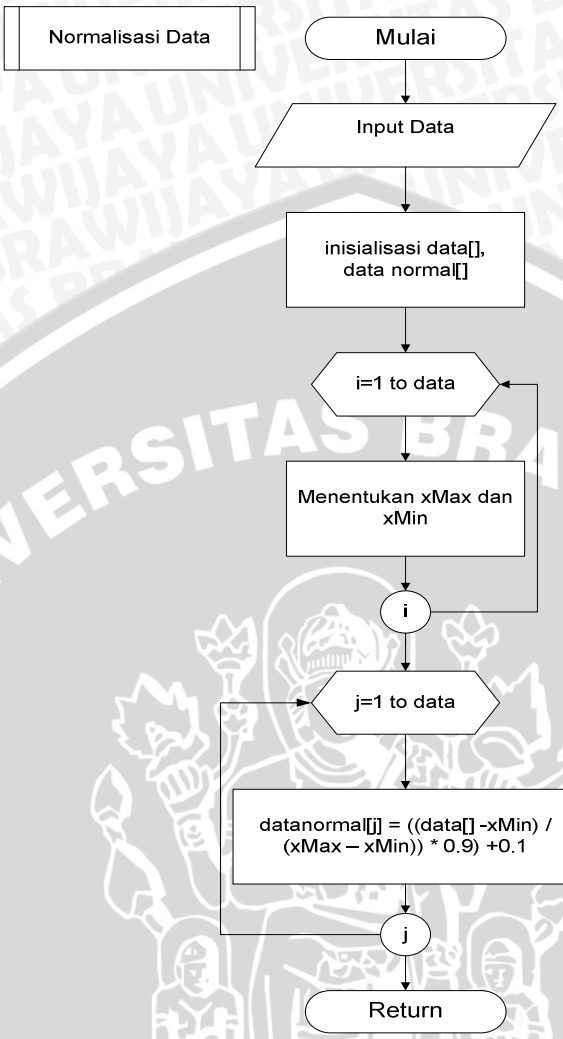
Secara keseluruhan sistem dibuat menggunakan *backpropagation* berbasis algoritma genetika untuk mencari bobot jaringan syaraf tiruan.Masukkan untuk sistem berupa harga saham yang terdiri dari empat parameter yaitu *open price, high price, low price,* dan *trade volume*. Langkah pertama seluruh harga saham dinormalisasi agar sesuai dengan fungsi aktivasi pada jaringan syaraf tiruan, selanjutnya lakukan proses pelatihan bobot jaringan terhadap data masukan harga saham. Pada proses pelatihan, digunakan *backpropagation* berbasis algoritma genetika sebagai pengganti algoritma pelatihan standar jaringan syaraf tiruan. Proses pelatihan akan menghasilkan bobot jaringan yang akan digunakan untuk tahap pengujian atau peramalan. Pada proses pengujian akan menghasilkan *output* berupa peramalan harga saham yang akan dibandingkan dengan harga saham aktual kemudian hitung tingkat kesalahan untuk mengevaluasi hasil peramalan. Proses secara keseluruhan bisa dilihat pada gambar 3.2.



Gambar 3.2 Diagram Alir Sistem

3.5.1 Normalisasi Data

Data *input* berupa harga saham yang perlu dinormalisasi dahulu sehingga masuk kedalam domain fungsi aktivasi *sigmoid* (persamaan 2.5) yaitu pada range (0,1). Namun karena nilai 1 dan 0 tidak akan pernah tercapai oleh fungsi *sigmoid* maka digunakan 0.1 dan 0.9 untuk mempresentasikan nilai terkecil dan terbesar dari data (Freeman & Skapura, 1991). Kemudian dilakukan normalisasi sesuai dengan persamaan 2.13. *Flowchart* proses ini dapat dilihat pada gambar 3.3

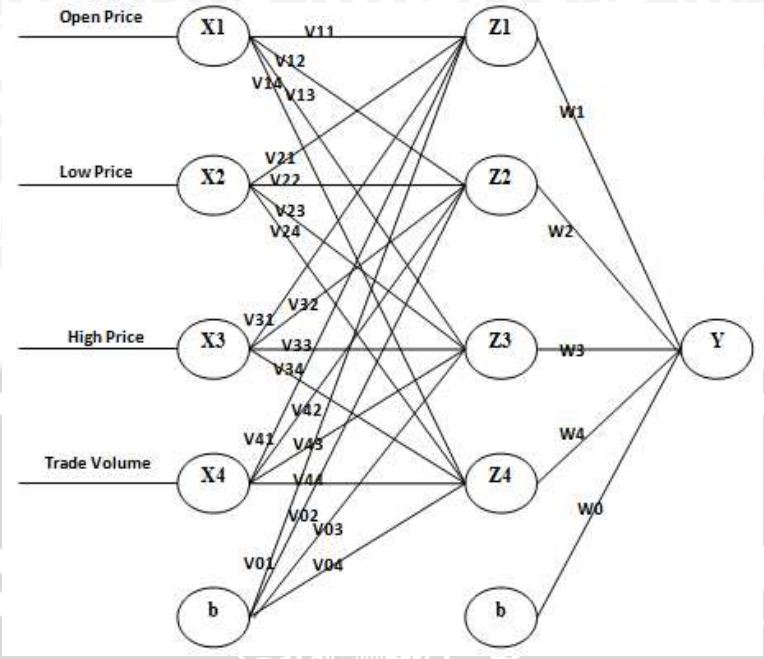


Gambar 3.3 Diagram Alir Proses Normalisasi

3.5.2 Penentuan Arsitektur Jaringan Syaraf Tiruan

Penelitian ini menggunakan 4 input parameter yaitu *open price, high price, low price*, dan *trade volume* dan 1 *output* yaitu *close price*. Penentuan *hidden layer* mengacu pada sub bab 2.11 sehingga jumlah *node* di *hidden layer* pada penelitian iniberjumlah 4 *neuron*.

Arsitektur jaringan syaraf tiruan yang digunakan pada penelitian ini dapat dilihat pada gambar 3.4

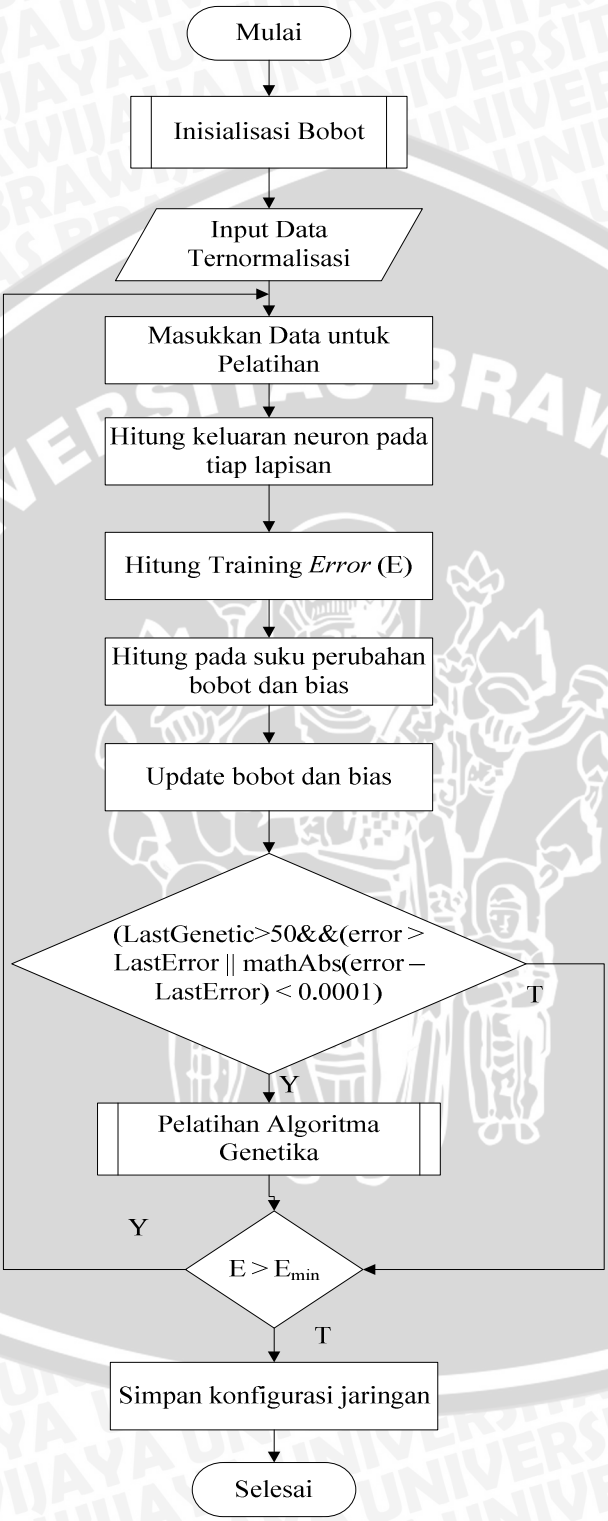


Gambar 3.4 Arsitektur Jaringan Syaraf Tiruan

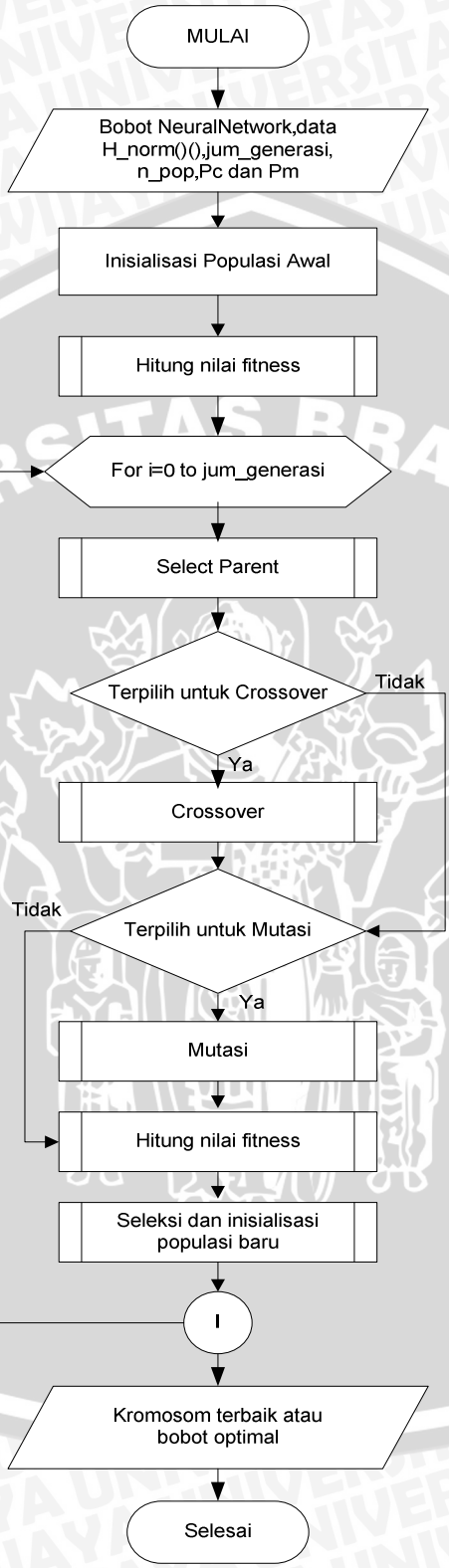
3.5.3 Pelatihan JST menggunakan *Backpropagation* Berbasis Algoritma Genetika

Tahapan algoritma genetika digunakan untuk mencari nilai-nilai bobot yang optimal untuk digunakan pada jaringan syaraf tiruan *backpropagation* untuk melakukan peramalan saham. Langkah-langkah yang dilakukan algoritma genetika dalam mencari bobot optimal untuk *backpropagation* dapat dilihat pada gambar 3.5

Flowchart proses algoritma genetika ditujukan pada gambar 3.5.



Gambar 3.5 *Flowchart* Pelatihan JST menggunakan *Backpropagation* Berbasis Algoritma Genetika



Gambar 3.6 *Flowchart* Algoritma Genetika

3.5.3.1 Pengkodean Kromosom

Kromosom yang digunakan dikodekan dengan *real-number encoding* yang mewakili masing-masing bobot pada jaringan *backpropagation*, yaitu semua bobot pada *input layer* menuju ke *hidden layer* dan juga bobot dari *hidden layer* menuju pada *output layer*. Arsitektur jaringan yang digunakan pada penelitian ini dapat dillihat pada gambar 3.4

Dari Gambar 3.4, maka panjang kromosom yang dibutuhkan adalah sebanyak 25 gen yang masing-masing bernilai riil dengan range dari 0 hingga 1. Representasi kromosom dijelaskan pada gambar 3.7.

Kromosom

V01	V02	V03	V04	V11	V12	V13	V14	V21	V22	V23	V24	V31
V32	V33	V34	V41	V42	V43	V44	W0	W1	W2	W3	W4	

Gambar 3.7 Representasi Kromosom

3.5.3.2 Fungsi *Fitness*

Perhitungan nilai *fitness* menggunakan proses *feedforward backpropagation* untuk menghitung *output* jaringan yang akan dicari selisihnya terhadap data aktual. Karena proses *feedforward backpropagation* dilakukan berulang-ulang sesuai dengan jumlah data maka perhitungan selisih antara *output* dan data aktual juga dilakukan berulang-ulang. Fungsi *fitness* yang digunakan dalam penelitian ini adalah 1/*delta*, yang mana *delta* diperoleh dari persamaan 2.12.

Berikut ini langkah-langkah untuk menghitung nilai *fitness* untuk menentukan kesalahan peramalan:

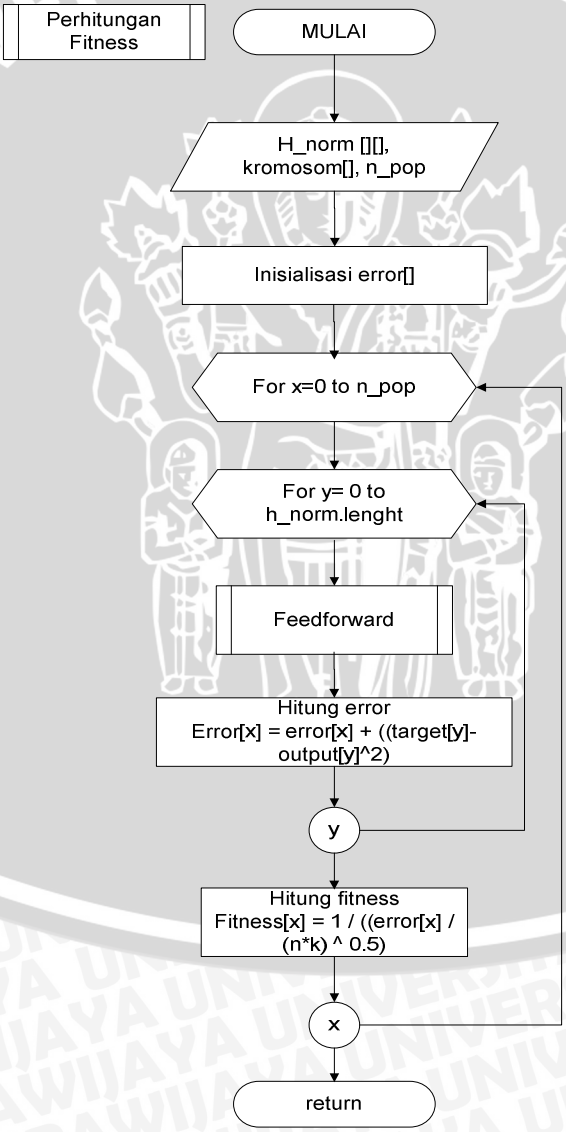
1. Memasukkan data harga saham yang telah dinormalisasi untuk menjadi input jaringan, kromosom yang telah diinisialisasi pada populasi awal, dan jumlah kromosom pada populasi(npop).
2. Menginisialisasi variabel *error* untuk menghitung kesalahan dan variabel ndata untuk menentukan banyaknya data.
3. Untuk x=1 sampai npop, dilakukan:
   - Untuk y=1 sampai ndata, hitung kesalahan peramalan dengan langkah-

langkah sebagai berikut:

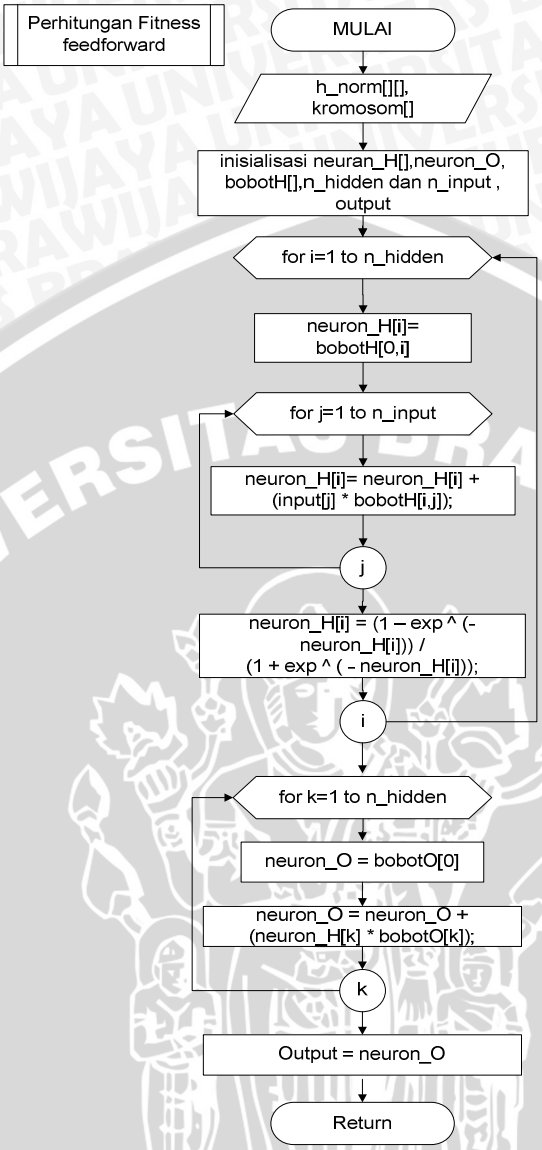
a) Lakukan proses *feedforward* pada masing-masing data *input* terhadap sebuah kromosom
b) Hitung kesalahan yang didapat dengan menhitung kuadrat selisih data *output* dengan data aktual.
c) Jumlahkan seluruh kesalahan yang didapatkan.
d) Hitung *fitness* kromosom

4. Jika x > npop maka proses hitung *fitness* berhenti.

Flowchart proses hitung *fitness* dapat dilihat pada gambar 3.8



Gambar 3.8*Flowchart* proses hitung *fitness*



Gambar 3.9*Flowchart* proses perhitungan *feedforward backpropagation* pada perhitungan nilai *fitness*

Berdasarkan Gambar 3.9, penjelasan proses *feedforward backpropagation* adalah sebagai berikut:

1. Persiapan data *input* berupa harga saham yang telah dinormalisasi dan kromosom awal hasil inisialisasi populasi awal
2. Inisialisasi neuron_h[] dan *output* untuk menyimpan hasil keluaran jaringan, bobotH[][], dan bobotO[]. BobotH merupakan *array* dua dimensi dengan 3 kolom dan 4 baris. Kolom menyatakan banyaknya *hidden neuron* sedangkan

baris menyatakan banyaknya *input neuron* jaringan. BobotO[] merupakan *array* 1 dimensi dengan panjang dengan panjang *array* 4 untuk menyimpan bobot *hidden neuron* menuju *output neuron.* Nilai-nilai untuk bobotH didapat dari kromosom yang merupakan nilai pada gen 1-12, sedangkan bobotO didapat dari nilai kromosom pada posisi gen 13-16.

3. Inisialisasi *nhidden* untuk menentukan jumlah *hidden neuron* yaitu 4 dan ninput untuk menentukan jumlah *input neuron* yaitu 4.
4. Selama j=1 hingga *nhidden*, lakukan inisialisasi nilai awal neuronH. Nilai awal neuron merupakan nilai bias untuk *hidden neuron* yang akan dihitung.

- Selama i=1 hingga *ninput* , lakukan:

- Menghitung nilai keluaran masing-masing *hidden neuron* sesuai dengan persamaan 2.6.
- Menghitung nilai aktivasi dari masing-masing *hidden neuron* menggunakan fungsi aktivasi *sigmoid bipolar*, sesuai persamaan 2.7.

5. jika j >*nhidden* maka proses perhitungan keluaran *hidden neuron* berhenti.
6. Selanjutnya menghitung keluaran jaringan, lakukan inisialisasi nilai awal *output neuron*. Nilai awal *output neuron* merupakan nilai bias *hidden neuron* menuju *output neuron*.
7. Selama j=1 hingga *nhidden*, lakukan:

- Menghitung nilai keluaran masing-masing *output neuron* sesuai dengan persamaan 2.8
- Menghitung nilai aktivasi *output neuron* menggunakan fungsi aktivasi linier sehingga menghasilkan *output* jaringan, sesuai persamaan 2.9.

3.5.3.3Proses Seleksi

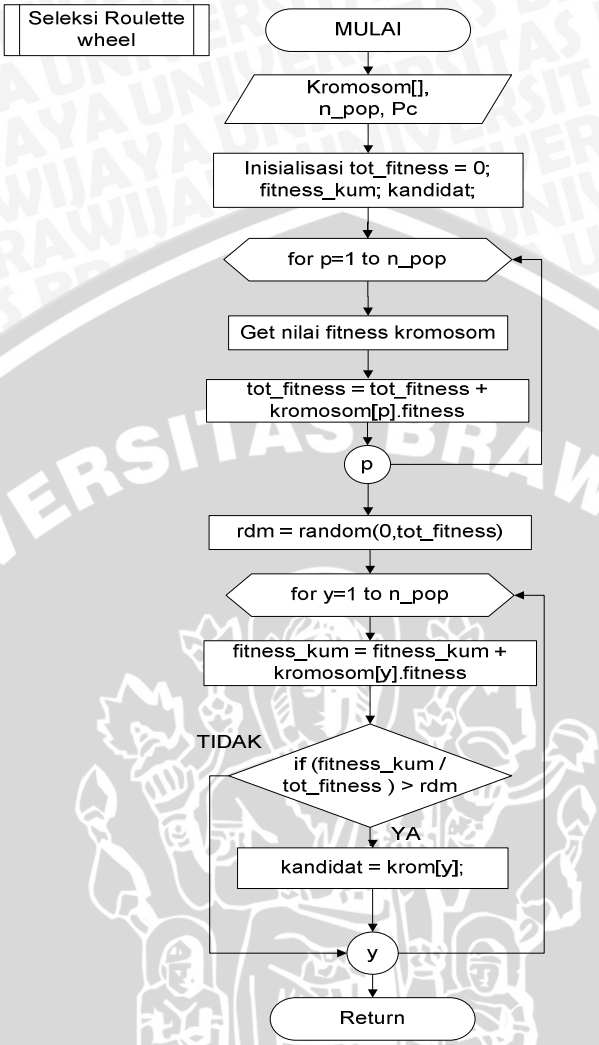
Proses seleksi dilakukan untuk memilih *parent* yang akan dipasangkan(*crossover*). Proses seleksi dilakukan sebanyak jumlah pasangan *parent* yang telah ditentukan sebelumnya. Proses ini dilakukan berdasarkan nilai *fitness* masing-masing kromosom. semakin baik nilai *fitness*-nya maka akan semakin besar kemungkinan kromosom tersebut akan terpilih untuk *crossover*. Hal ini bertujuan untuk mendapatkan individu yang lebih baik daripada induknya. Metode yang

digunakan untuk menentukan *parent* yang berhak melakukan *crossover* dalam penelitian ini adalah metode *roulette wheel.*

Langkah-langkah dalam metode *roulette wheel* adalah sebagai berikut:

1. Mengambil nilai *fitness* masing-masing kromosom pada populasi.
2. Menjumlahkan seluruh nilai *fitness* untuk mendapatkan total *fitness.*
3. Menghitung *fitness relatif* tiap kromosom dengan membagi nilai *fitness* dengan total *fitness.*
4. Menghitung *fitness* kumulatif tiap kromosom, *fitness* kumulatif digunakan untuk menentukan batasan daerah tiap kromosom.
5. Melakukan proses *roulette* dengan membangkitkan bilangan acak mulai 0 hingga nilai total *fitness*.
6. Jika nilai random berada pada salah satu daerah kromosom, maka kromosom tersebut akan terpilih menjadi *parent* untuk melakukan *crossover*.
7. Melakukan proses *roulette wheel* sebanyak jumlah *parent.*

Flowchart untuk proses *roulette wheel* dapat dilihat pada gambar 3.10



Gambar 3.10*Flowchart* proses *roulette wheel*

3.5.3.4 *Crossover*

Parent yang terpilih melalui proses seleksi akan melakukan *crossover*. *Crossover* dilakukan dengan mengkombinasikan gen-gen induk untuk mendapatkan keturunan baru yang diharapkan lebih baik daripada induknya. Proses *crossover* dilakukan dengan cara*one-cut-point-crossover*. Titik potong ditentukan secara random dari 1 hingga panjang kromosom.Kemudian tukan gen-gen *parent* 1 dan *parent* 2 sesuai batas titik potong. Contoh proses *crossover* ditunjukkan pada gambar 3.11.

*parent*1

0.576	0.855	0.095	0.790	0.145	0.374	0.242	0.284	0.909	0.181	0.298	0.139	0.347
0.230	0.763	0.419	0.838	0.337	0.624	0.841	0.062	0.661	0.573	0.321	0.511	

*parent*2

0.719	0.803	0.785	0.540	0.772	0.940	0.993	0.249	0.616	0.807	0.690	0.964	0.434
0.302	0.029	0.348	0.923	0.399	0.705	0.350	0.898	0.268	0.005	0.299	0.539	

*child*1

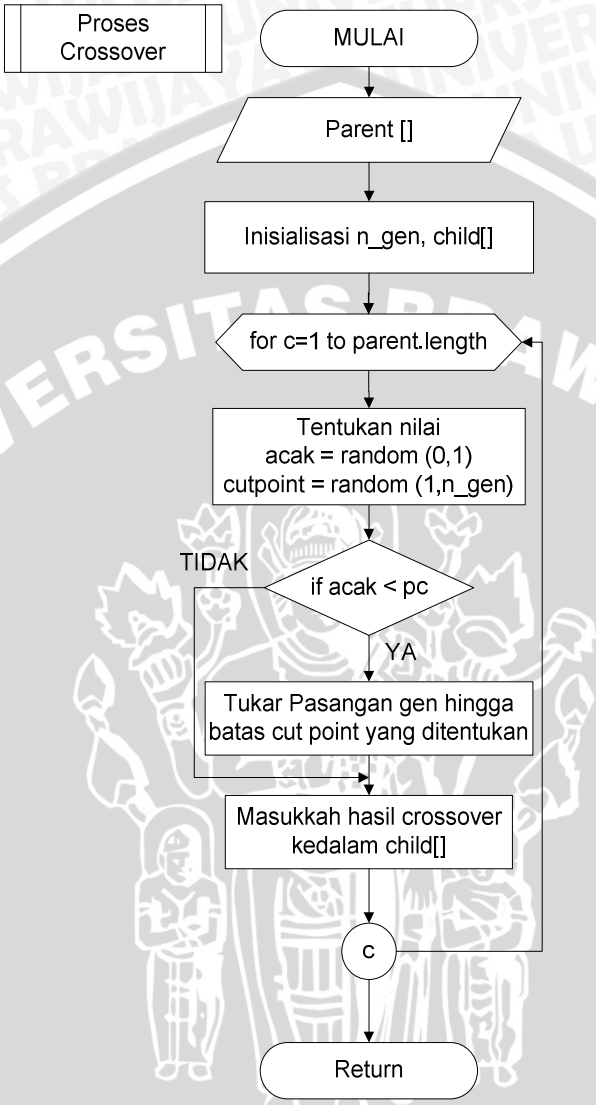
0.576	0.855	0.095	0.790	0.145	0.940	0.993	0.249	0.616	0.807	0.690	0.964	0.347
0.230	0.763	0.419	0.838	0.337	0.624	0.841	0.062	0.661	0.573	0.321	0.511	

*child*2

0.719	0.803	0.785	0.540	0.772	0.374	0.242	0.284	0.909	0.181	0.690	0.964	0.434
0.302	0.029	0.348	0.923	0.399	0.705	0.350	0.898	0.268	0.005	0.299	0.539	

Gambar 3.11 Contoh proses*one-cut-point crossover*

Flowchart untuk proses *crossover* dapat dilihat pada gambar 3.12



Gambar 3.12 *Flowchart* proses *crossover*

Dari gambar 3.12, penjelasan dari proses *crossover* sebagai berikut:

1. Persiapan data *input* yaitu kromosom *parent* yang terpilih dari proses *SelectParent*(*Parent[]*).
2. Inisialisasi variabel ngen untuk menentukan jumlah gen dalam kromosom yaitu 25, dan *child[]* untuk menyimpan kromosom hasil *crossover*.
3. Selama c=1 sampai *parent.length*, lakukan:
    - Membangkitkan nilai riil secara random dari 0 hingga 1, apabila nilai random lebih kecil dari Pc, maka

- Menentukan *cut-point* dengan membangkitkan bilangan random dari 1-25.
- Menukar pasangan gen hingga batas *cut-point*

- Bila nilai random lebih besar daripada Pc, maka tidak terjadi proses *crossover,* sehingga kromosom anak sama persis dengan induknya.
- Memasukkan hasil *crossover* ke array *child.*

4. Jika c >*parent.lentgh* maka proses *crossover* berhenti.

3.5.3.5 Mutasi

Pada algoritma genetika, mutasi dilakukan untuk mendapatkan nilai solusi yang baru dari hasil mutasi gen yang terpilih. Selain itu mutasi gen dilakukan untuk menghindari solusi terjebak dalam lokal optima dan menemukan global optima. Proses mutasi dilakukan pada kromosom yang terpilih saja.

Proses mutasi yang digunakan pada penelitian ini adalah *Exchange Mutation* yang dilakukan dengan langkah-langkah sebagai berikut:

1. Memasukkan kromosom dan nilai Pm (probabilitas mutasi)
2. Mengacak nilai dengan rentang antara 0 dan 1 (p)
3. Apabila nilai p lebih kecil dari Pm maka akan dilakukan mutasi, jika tidak maka tidak dilakukan mutasi
4. Proses Mutasi akan dilakukan dari kromosom pertama sampai kromosom terakhir yang mengalami mutasi
5. Memilih 2 posisi gen secara random pada setiap kromosom yang mengalami mutasi
6. Menukarkan isi dari kedua posisi gen tersebut
7. Proses berhenti jika telah sampai pada kromosom terakhir yang akan dimutasi.

Contoh mutasi ditunjukkan pada gambar 3.13.

V01	V02	V03	V04	V11	V12	V13	V14	V21	V22	V23	V24	V31
V32	V33	V34	V41	V42	V43	V44	W0	W1	W2	W3	W4	

V01	V02	V03	V04	V11	V12	V13	V14	V21	V22	V23	V24	V31
V32	V33	V34	V41	V42	V43	V44	W0	W1	W2	W3	W4	

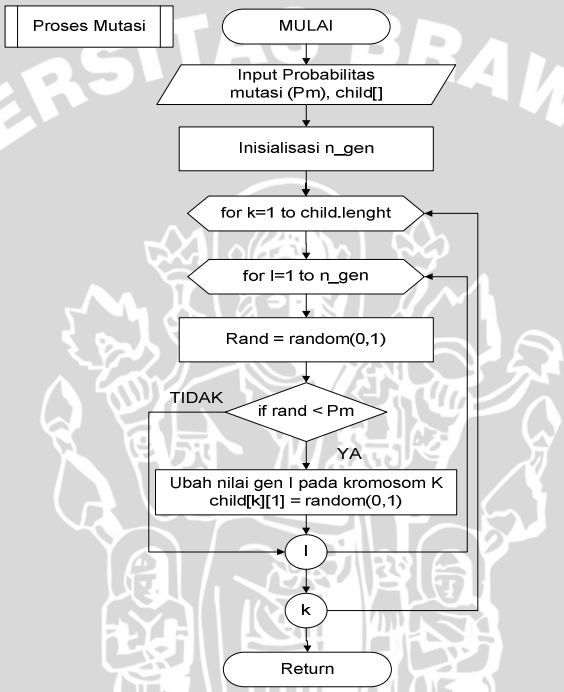
Gambar 3.13 Contoh proses mutasi

Keterangan warna:

Gen 1 yang ditukar

Gen 2 yang ditukar

Flowchart proses mutasi dapat dilihat pada gambar 3.14



Gambar 3.14*Flowchart* proses Mutasi

Dari gambar 3.13, penjelasan dari proses mutasi adalah sebagai berikut:

1. Persiapan data *input* yaitu kromosom anak (Child[]) dan probabilitas mutas(Pm).
2. Inisialisasi variabel nGen untuk menentukan jumlah gen dalam satu kromosom yaitu 25.
3. Selama k=1 hingga child.Length lakukan:
    - Selama l = 1 hingga nGen, lakukan:
        - Membangkitkan nilai riil secara random antara 0 - 1, apabilai nilai random lebih kecil daripada Pm, maka

- Mengubah nilai gen dari posisi gen dengan nilai riil baru yang dibangkitkan secara random antara 0-1.

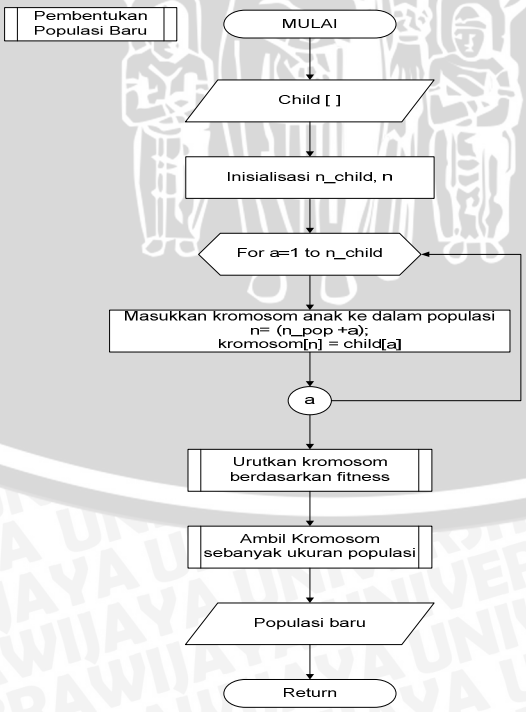
4. Jika k>n_child maka proses mutasi berhenti.

3.5.3.6 Seleksi Populasi Baru

Setelah proses *crossover* dan mutasi telah selesai dilakukan, kromosom-kromosom baru yang terbentuk dihitung *fitness*-nya. Ukuran populasi dari proses genetika selalu sama, oleh karena itu hanya individu atau kromosom baru yang terbaik yang dapat masuk dalam populasi membentuk generasi baru yang lebih baik. Untuk itu perlu dilakukan seleksi. Metode yang digunakan untuk proses seleksi adalah *rank based selection.*

Semua kromosom atau individu baru yang terbentuk dimasukkan dahulu ke dalam populasi.Kemudian seluruh kromosom diurutkan berdasarkan nilai *fitness*-nya.Setelah itu diambil kromosom yang terbaik sebanyak ukuran populasi yang telah ditentukan sebelumnya sehingga membentuk populasi baru. Individu yang nilai *fitness*-nya tidak mencukupi untuk masuk ke dalam populasi akan dihilangkan.

Flowchart proses pembentukan populasi baru dapat dilihat pada gambar 3.15



Gambar 3.15*Flowchart* prosespembentukan populasi baru

3.6 Perhitungan Manual

3.6.1 Normalisasi Data

Normalisasi dihitung menggunakan Persamaan 2.13.

Tabel 3.2 Data awal saham sebelum dilakukakan normalisasi

No	Date	Open	High	Low	Close	Trade Volume
1	2005-01-03	4875	4925	4825	4925	10475000
2	2005-01-04	4925	4975	4900	4950	27248500
3	2005-01-05	4925	4925	4875	4875	23008000
4	2005-01-06	4875	5000	4875	4975	37726500
5	2005-01-07	5000	5125	5000	5125	43628500
...	...	...	...	...	...	...
...	...	...	...	...	...	...
481	2006-12-21	9950	10050	9950	10050	27649500
482	2006-12-22	10000	10000	9850	9900	13366000
483	2006-12-26	9900	10000	9850	9900	4274000
484	2006-12-27	10000	10050	9950	10000	9615000
485	2006-12-28	10150	10150	10000	10100	22664000

Tabel 3.3 Data saham yang telah ternormalisasi

No	Date	Open	High	Low	Close	Trade Volume
1	2005-01-03	0.191	0.197	0.185	0.197	0.155
2	2005-01-04	0.197	0.203	0.194	0.200	0.280
3	2005-01-05	0.197	0.197	0.191	0.191	0.248
4	2005-01-06	0.191	0.206	0.191	0.203	0.358
5	2005-01-07	0.206	0.221	0.206	0.221	0.407
...	...	...	...	...	...	...
...	...	...	...	...	...	...
481	2006-12-21	0,793	0,805	0,793	0,805	0,283
482	2006-12-22	0,799	0,799	0,781	0,787	0,177
483	2006-12-26	0,787	0,799	0,781	0,787	0,109

484	2006-12-27	0,799	0,805	0,793	0,799	0,149
485	2006-12-28	0,817	0,817	0,799	0,811	0,246

Parameter awal untuk pelatihan yaitu

$\alpha = 0.1$

$\varepsilon = 0.2$

jumlah unit *input* = 4

jumlah unit di *hidden layer* = 4

jumlah unit *output* = 1

Inisialisasi bobot awal dengan nilai acak antara -1 hingga 1, dan hasilnya pada Tabel 3.4 dan Tabel 3.5.

Tabel 3.4 Tabel Data Nilai Bobot Awal V_{ij}

V_{ij}	x1	x2	x3	x4	$\| V_j \|$
z1	0.06	0.54	0.46	-0.33	0.837963
z2	-0.37	-0.36	0.34	0.07	0.88942
z3	0.30	0.10	-0.19	-0.24	0.501946
z4	-0.69	0.20	0.05	-0.24	1.176175

Tabel 3.5 Tabel Data Nilai Bobot Awal W_{ij}

W_{ij}	z1	z2	z3	z4	$\| W_j \|$
t1	0.17	-0.70	0.18	-0.80	1.175116

1. Jika kondisi penghentian belum terpenuhi, maka langkah 2-7 diulangi terus menerus.
2. Untuk setiap pasang data pelatihan, langkah 3-7 diulangi sampai semua pola data dilatih.

Pola data 1

3. Setiap input mengirim sinyal ke unit tersembunyi. Hitung keluaran di unit tersembunyi (z_j) dengan persamaan 2.6. hasil perhitungannya ditunjukkan pada Tabel 3.6 dan Tabel 3.7.

Tabel 3.6 Tabel Operasi pada *Hidden Neuron*

J	z_in
1	0.6442
2	-0.4472
3	0.5257
4	0.3251

Tabel 3.7 Tabel Aktivasi pada *Hidden Neuron*

j	Z
1	0.6557
2	0.39
3	0.6285
4	0.5806

4. Keluaran unit *output* dihitung dengan menggunakan persamaan 2.8 dan 2.9 , sehingga menghasilkan *y_in* menghasilkan *y*yang ditunjukkan pada Tabel 3.8 dan tabel 3.9.

Tabel 3.8 Tabel Operasi pada *Output*

J	*y_in*
1	0.2653

Tabel 3.9 Tabel Aktivasi pada *Output*

J	*Y*
1	0.5659

5. Kemudian mencari nilai factor δ di unit *output* y_k denagn menggunakan persamaan 2.10.
6. Setelah itu dihitung suku perubahan bobot ΔW dengan α = 0.1. Hasil perhitungan ΔW ditunjukkan pada Tabel 3.10.

Tabel 3.10 Tabel Suku Perubahan Bobot

ΔW	k_1
0	-0.05509
1	-0.03612
2	-0.02149
3	-0.03462
4	-0.03198

7. Kemudian hitung penjumlahan kesalahan dari unit tersembunyi menggunakan persamaan 2.11 dan dilanjutkan dengan persamaan 2.12. Perhitungan pertama yang menggunakan persamaan 2.11 hasilnya ditunjukkan pada Tabel 3.11.

Tabel 3.11 Tabel Jumlah Kesalahan Pada *Hidden* Unit

	1	2	3	4
δ_in	-0.0859	0.47315	-0.03695	0.432001

Selanjutnya dihitung faktor kesalahan δ di unit tersembunyi yang dihasilnya ditunjukkan oleh Tabel 3.12.

Tabel 3.12 Tabel Faktor Kesalahan *Hidden* Unit

	1	2	3	4
Δ	-0.01939	-0.02044	-0.02006	-0.02092

Dari nilai factor kesalahan *hidden* unit dapat dihitung suku perubahan bobot pada unit tersembunyi (ΔV_{ij}) dengan mengunakan persamaan 2.12. Hasilnya ditunjukkan pada Tabel 3.13.

Tabel 3.13 Tabel Suku Perubahan Bobot ke *Hidden* Unit

ΔV	x1	x2	x3	x4	1
z1	-0.0017	-0.00019	-0.00175	-0.00019	-0.00194
z2	-0.0017	-0.00019	-0.00175	-0.00019	-0.00194

z3	-0.0017	-0.00019	-0.00175	-0.00019	-0.00194
z4	-0.0017	-0.00019	-0.00175	-0.00019	-0.00194

Langkah yang terakhir yaitu peng-*update*-an bobot dan bias dengan menggunakan persamaan 2.22 untuk W_{ij} (bobot *output*unit) dan persamaan 2.13 untuk V_{ij} (bobot *hidden* unit). Bobot unit *output* dan *hidden* unit yang baru ditunjukkan pada Tabel 3.14 dan Tabel 3.15.

Tabel 3.14 Tabel Bobot *Output* Unit yang Baru

W_{ij}	t_1
z1	0.082
z2	-0.597
z3	0.120
z4	-0.676
1	0.128

Tabel 3.15 Tabel Bobot *Hidden* Unit yang Baru

V_{ij}	x1	x2	x3	x4	1
1	0.07	0.60	0.52	-0.37	0.01
2	-0.39	-0.38	0.36	0.08	0.30
3	0.56	0.19	-0.37	-0.44	-0.02
4	-0.55	0.16	0.04	-0.19	-0.13

Proses ini diulangi terus-menerus hingga kondisi pelatihan algoritma genetika terpenuhi. Ketika telah terpenuhi maka akan dilanjutkan dengan pelatihan menggunakan algoritma genetika,

3.6.2 Pembangkitan Populasi Awal

Inisialisasi populasi awal dilakukan dengan pengubahan bobot dari pelatihan *backpropagation* menjadi kromosom dan dilakukan sebanyak jumlah populasi yang akan dibentuk, kromosom lain yang terbentuk didapatkan dari nilai acak.:

Kromosom yang terbentuk dari bobot JST

0.01	0.30	-0.02	-0.13	0.07	-0.39	0.56	-0.55	0.06	-0.38	0.19	0.16	0.52
0.36	-0.37	0.04	-0.37	0.08	-0.44	-0.19	0.12	0.08	-0.59	0.12	-0.67	

3.6.3 Perhitungan *Fitness*

Perhitungan *fitness* dilakukan dengan menggunakan *feedforward backpropagation* yaitu persamaan 2.5 hingga persamaan 2.8. Setiap kromosom berupa susunan bobot jaringan dikalikan dengan data saham yang ternormalisasi, kemudian dihitung total kesalahan yang terjadi

Dengan menggunakan kromosom pertama, perhitungan *fitness*-nya sebagai berikut:

Data 1:

Menghitung keluaran unit tersembunyi dengan persamaan 2.6:

$z\_in_j = v_{0j} + \sum_{i=1}^{n} xivij$

Keterangan:

nilaix_i = nilai normaliasasi harga saham yang berasal dari tabel 3.3

$z\_in_1 = v_{01} + x_1v_{11} + x_2v_{21} + x_3v_{31} + x_4v_{41}$

z_in_1 =0.01+(0.191)(0.07)+(0.197)(0.6)+(0.185)(0.52)+(0.155)(-0.37)

=0.18

$z\_in_2 = v_{02} + x_1v_{12} + x_2v_{22} + x_3v_{32} + x_4v_{42}$

z_in_2 =0.03+(0.191)(0.6)+(0.197)(-0.38)+(0.185)(0.36)+(0.155)(0.08)

= 0.41

$z\_in_3 = v_{03} + x_1v_{13} + x_2v_{23} + x_3v_{33} + x_4v_{43}$

z_in_3 = -0.02+(0.191)(0.56)+(0.197)(0.19)+(0.185)(-0.37)+(0.155)(-0.44)

= -0.01

$z\_in_4 = v_{04} + x_1v_{14} + x_2v_{24} + x_3v_{34} + x_4v_{44}$

z_in_4 = -0.13+ (0.191)(-0.55) + (0.197)(0.16)+(0.185)(0.04)+(0.155)(-0.19)

= -0.22

Menghitung aktivasi unit tersembunyi dalam persamaan 2.7 :

$z_1 = f(z_{in1}) = \frac{1-e^{-0.18}}{1+e^{-0.18}} = 0.452$

$z_2 = f(z_{in\,2}) = \frac{1-e^{-0.41}}{1+e^{-0.411}} = 0.484$

$z_3 = f(z_{in3}) = \frac{1-e^{0.01}}{1+e^{0.01}} = 0.216$

$z_4 = f(z_{in4}) = \frac{1-e^{0.22}}{1+e^{0.22}} = 0.493$

Hitung pengeluaran y_k menggunakan persamaan 2.8 :

$y\_in_k = w_{ok} + \sum_{j=1}^{p} z_j w_{jk}$

Untuk menghitung nilai *output* hanya digunakan bobot yang terhubung ke neuron *output*. Berdasarkan arsitektur jaringan yang dibuat sebelumnya , terdapat satu neuron *output* dan bobot yang terhubung ke neuron *output* sebanyak empat buah , yaitu yang berasal dari ketiga *hidden* neuron dan neuron bias. Maka persamaan 2.8 dapat dijabarkan sebagai berikut :

$y\_in = w_0 + z_1w_1 + z_2w_2 + z_3w_3 + z_4w_4$

y_in = 0.062 +(0.452)(0.661) + (0.484)(0.573) + (0.216)(0.321)+ (0.493)(0.511)

= 0.961

Aktivasi keluaran unit y_k dan dengan persamaan 2.9 :

$y_k = f(y_{in\,k}) = y_{ink}$

$y = f(y_{in}) = 0.961$

hitung kembali nilai y untuk data kedua dan seterusnya dengan langkah yang sama seperti contoh perhitungan data1 dan tetap menggunakan kromosom atau nilai bobot yang sama.

Hasil perhitungan *feedforward* untuk seluruh data dengan kromosom 1 ditunjukkan pada tabel 3.16.

Tabel 3.16 Hasil perhitungan *feedforward* untuk seluruh data dengan kromosom 1

Data	z_{in1}	z_{in2}	z_{in3}	z_{in4}	Z_1	Z_2	Z_3	Z_4	y_in	Y
1	0,978	1,057	0,439	1,080	0,453	0,484	0,216	0,493	0,960	0,960
2	1,092	1,105	0,527	1,191	0,497	0,502	0,257	0,534	1,034	1,034
3	1,059	1,092	0,503	1,163	0,485	0,497	0,246	0,523	1,014	1,014
4	1,158	1,128	0,572	1,254	0,522	0,511	0,278	0,556	1,073	1,073
5	1,215	1,155	0,619	1,304	0,542	0,520	0,300	0,573	1,108	1,108

Setelah semua nilai y dari data diketahui, selanjutnya hitung selisih nilai y (*output* jaringan) terhadap target, yang dalam penelitian ini adalah harga penutupan (*close price).* Nilai selisih digunakan untuk mendapatkan nilai *delta* , yang dihitung menggunakan persaman 2.12.

$$delta = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(d(i)-y(i))^2}$$

$$= \sqrt{\frac{(0{,}197-0{,}960)^2+(0{,}201-1{,}034)^2+\cdots+(0{,}799-1{,}421)^2+(0{,}811-1{,}464)^2}{485}}$$

$$= 0{,}773911$$

$$Fitness = \frac{1}{delta} = \frac{1}{0.7739} = 1.2921$$

Keterangan : nilai d_i adalah nilai target yaitu harga *close* yang telah dinormalisasikan yang ada pada tabel 3.15.

Lakukan cara yang sama untuk menghitung kromosom yang ada dalam populasi. Hasil *fitness* seluruh kromosom dalam populasi , yang ada pada contoh ini ada 4 kromosom dalam 1 populasi dapat dilihat pada tabel 3.17

Tabel 3.17 Hasil *fitness* tiap kromosom

Hasil Kromosom – ke	*Fitness*
1	1.2921
2	0.7424

3	0.8394
4	0.6300

3.6.4 Seleksi parent dengan *Roulette Wheel*

Langkah awal untuk melakukan seleksi *roulette wheel* adalah dengan menghitung total*fitness*untuk memperoleh *fitness* kumulatif tiap kromosom.

Total *fitness*=1.2921+ 0.7424+ 0.8394 + 0.6300 = 3.5039

Hitung masing-masing *fitness relative* terhadap total *fitness* yaitu dengan membagi masing-masing *fitness* individu dengan total *fitness.*

Fitness relatif kromosom 1 = 1.2921 / 3.5039= 0.3688

Fitness relatif kromosom 2 = 0.7424/ 3.5039 = 0.2118

Fitness relatif kromosom 3 = 0.8394/ 3.5039 = 0.2396

Fitness relatif kromosom 4 = 0.6300/ 3.5039 = 0.1798

Hitung *fitness* komulatif tiap kromosom untuk memperoleh skala atau daerah *rhoulette whell.*

Kromosom 1 = 0+0.3688 = 0.3688

Kromosom 2 = 0.3688+ 0.2118= 0.5806

Kromosom 3 = 0.5806+ 0.2396= 0.8202

Kromosom 4 = 0.8202+ 0.1798 = 1

Dari *fitness* komulatif tersebut maka masing- masing kromosom memiliki daerah sebagai berikut :

Kromosom 1 0 – 0.3688

Kromosom 2 0.3689 – 0.5806

Kromosom 3 0.5807 – 0.8202

Kromosom 4 0.8203 – 1

Bangkitkan nilai random antara 0 dan 1 untuk menentukan kromosom yang akan terpilih melakukan *crossover*. Misalnya nilai random pertama 0,5587 dan nilai random kedua 0,7032. Berdassarkan batasan atau daerah masing – masing kromosom, maka untuk random pertama yang terpilih adalah kromosom ke- 2.

Sedangkan random kedua yang terpilih adalah kromosom ke-3. Maka selanjutnya proses *crossover* antara kromosom 2 dan kromosom 3 akan dilakukan.

3.6.5 Proses *Crossover*

Proses *crossover* dilakukandengan cara mengkombinasikan 2 kromosom *parent* yang terpilih dengan dua titik potong yang ditentukan secara acak.

Langkah 1 :

Parent 1 :

Kromosom 2 :

0.106	0.839	0.698	0.795	0.355	0.528	0.155	0.675	0.365	0.029	0.067	0.197	0.627
0.124	0.969	0.487	0.973	0.784	0.493	0.675	0.659	0.802	0.054	0.454	0.769	

Parent 2 :

Kromosom 3 :

0.719	0.803	0.785	0.540	0.772	0.940	0.993	0.249	0.616	0.807	0.690	0.964	0.434
0.302	0.029	0.348	0.923	0.399	0.705	0.350	0.898	0.268	0.005	0.299	0.539	

Langkah 2 :

Menentukan bilangan random antara 1 sampai (banyaknya gen – 1) yaitu 24.Misalnya nilai random yang muncul adalah 9. Maka tiap potongan berada pada posisi ke –9 .

Child 1 :

0.106	0.839	0.698	0.795	0.355	0.940	0.993	0.249	0.616	0.807	0.690	0.964	0.434
0.302	0.029	0.348	0.923	0.399	0.493	0.675	0.659	0.802	0.054	0.454	0.769	

Child 2 :

0.719	0.803	0.785	0.540	0.772	0.528	0.155	0.675	0.365	0.029	0.067	0.197	0.627
0.124	0.969	0.487	0.973	0.784	0.705	0.350	0.898	0.268	0.005	0.299	0.539	

Keterangan warna :

: Kromosom dari parent 1

: kromosom dari parent 2

3.6.6 Proses Mutasi

Proses mutasi ditentukan dengan sebuah parameter yaitu probabilitas mutasi. Misalnya nilai probabilitas mutasi adalah 15% maka diharapkan 15% dari jumlah gen pada seluruh individu dalam satu populasi akan mengalami mutasi. Proses mutasi diawali dengan pembangkitan nilai acak untuk menentukan apakah gen pada individu tersebut dimutasi atau tidak. Misalnya kromosom *child* pertama, pada tiap gen nya akan dibangkitkan nilai riil antara 0-1 secara acak. Apabila nilai acak yang muncul kurang dari nilai probalitas mutasi, maka gen tersebut akan dimutasi. Langkah mutasi yagn dilakukan dalam penelitian ini adalah sebagai berikut :

1. Memilih secara acak posisi 2 gen. Misalnya posisi gen ke-5 dan ke-11. ***Child 1***

0.106	0.839	0.698	0.795	0.355	0.528	0.155	0.675	0.365	0.807	0.690	0.964	0.434
0.302	0.029	0.348	0.923	0.399	0.705	0.350	0.898	0.268	0.005	0.299	0.539	

2. Menukar isi gen pada posisi gen ke-5 dan posisi gen ke-9.

Child 1

0.106	0.839	0.698	0.795	0.690	0.528	0.155	0.675	0.365	0.807	0.355	0.964	0.434
0.302	0.029	0.348	0.923	0.399	0.705	0.350	0.898	0.268	0.005	0.299	0.539	

3.6.7 Pembentukan Populasi Baru

Individu – individu yang telah melalui proses *crossover* dan mutasi kemudian dimasukkan kedalam satu populasi sementara untuk di seleksi membentuk satu populasi baru. Pembentukan populasi baru dilakukan berdasarkan nilai *fitness* terbaik. Untuk itu perlu dilakukan perhitungan *fitness* lagi untuk masing- masing kromosom. Dengan cara yang sama seperti sebelumnya, *fitness* masing- masing kromosom dapat dilihat pada tabel 3.18

Tabel 3.18 Hasil *fitness* tiap kromosom setalah proses reproduksi.

	Fitness
Kromosom 1	1.2921
Kromosom 2	0.7424
Kromosom 3	0.8394
Kromosom 4	0.6300
Child 1	0,8564
Child 2	0,6554
Child 1 (mutasi)	0,8792

Selanjutnya urutkan kromosom berdasarkan *fitness* dimulai dari yang terbesar hingga terkecil. Hasil pengurutan dapat dilihat pada tabel 3.19.

Tabel 3.19 Hasil pengurutan individu berdasarkan *fitness*.

	Fitness
Kromosom 1	1.2921
Child 1 (mutasi)	0,8792
Child 1	0,8564
Kromosom 3	0.8394
Kromosom 2	0.7424
Child 2	0,6554
Kromosom 4	0.6300

Kemudian kromosom teratas untuk dijadikan populasi baru, sedangkan individu yang *fitness*-nya tidak mencukupi akan dibuang. Individu yang terpilih untuk membentuk populasi baru disebut juga generasi ke-1. Proses genetika ini diulang terus hingga generasi maksimum yang ditentukan per pemanggilan. Setelah iterasi maksimum tercapai maka kromosom yang terbaik yang telah dihasilkan akan dirubah kembali ke bobot JST untuk dilanjutkan untuk digunakan oleh *backpropagation neural network.*

Proses ini diulangi terus-menerus hingga semua pola data dilatihkan dan akan berhenti jika memenuhi kondisi berhenti misalnya *error* lebih kecil dari *error* minimal yang ditentukan atau sudah mencapai batas epoch maksimal.

3.7 Perancangan Antar Muka

Pada bagian ini akan dijelaskan antarmuka sistem yang akan dibangun. Gambar 3.15 menunjukkan tampilan rancangan antarmuka untuk inisialisasi data *training* pada sistem. Berikut adalah penjelasan gambar 3.15 berdasarkan no:

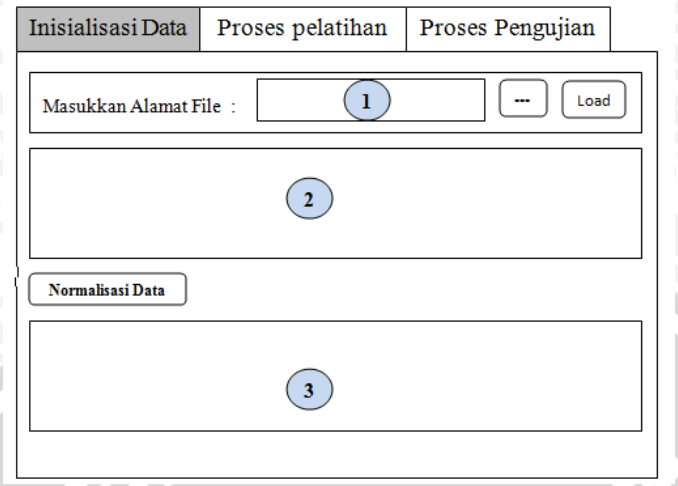
1. Bagian ini berfungsi untuk memasukkan alamat data yang akan dilatih
2. Menampilkan Data yang belum dinormalisasi
3. Bagian ini untuk menampilkan data yang telah ternormalisasi

Antarmuka pelatihan ditunjukkan pada gambar 3.16 ,penjelasana berdasarkan nomer pada gambar adalah sebagai berikut:

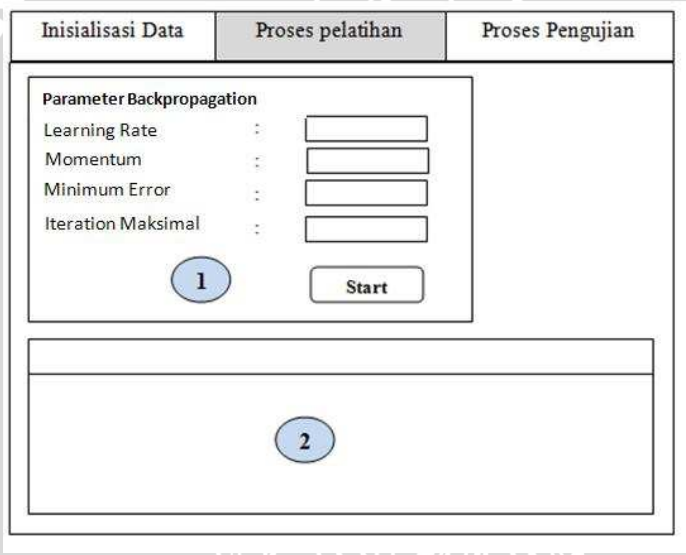
1. Bagian ini merupakan kontrol untuk menentukan nilai-nilai parameter genetika yang akan digunakan
2. Bagian kedua menampilkan proses pelatihan pada sistem.

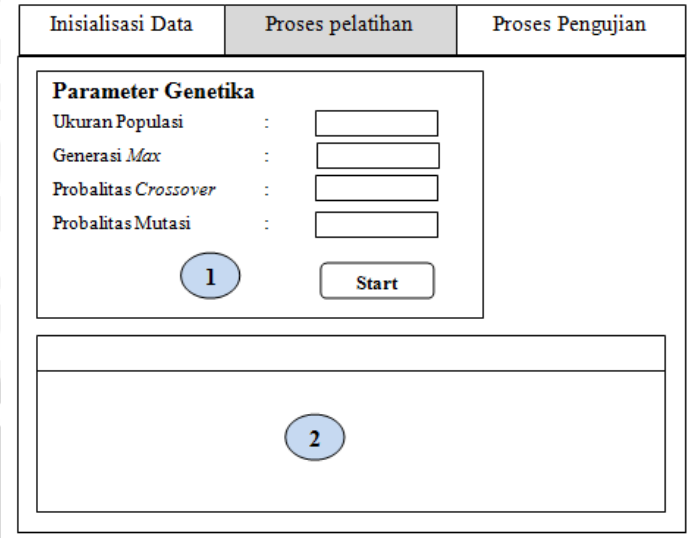
Antarmuka pengujian ditunjukkan pada gambar 3.16 , penjelasan berdasarkan nomer pada gambar adalah sebagai berikut:

1. Bagian ini menampilkan bobot dan bias dari hasil pelatihan sistem
2. Bagian ini menampilkan data uji yang telah ternormalisasi yang akan diujikan pada sistem
3. Bagian terakhir untuk menampilkan proses pengujian sistem.

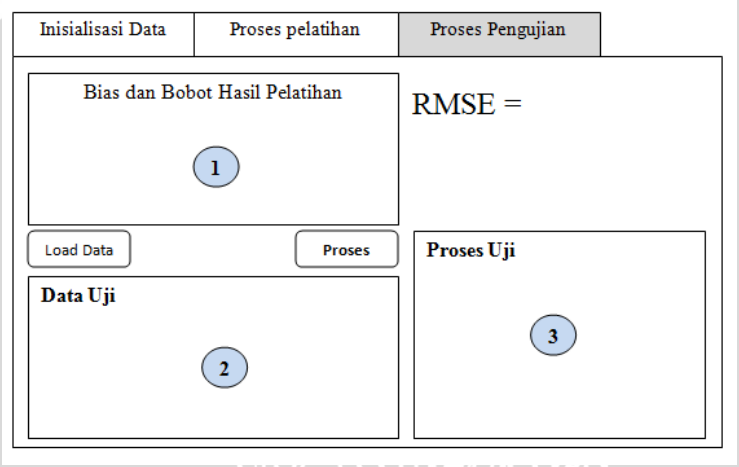


Gambar 3.16 Tampilan Rancangan antarmuka Inisialisasi Data





Gambar 3.17 Tampilan Rancangan Antarmuka Pelatihan



Gambar 3.18 Tampilan Rancangan Antarmuka Pengujian

3.8 Perancangan Uji Coba

Uji coba untuk meramalkan harga saham dilakukan dengan memasukkan nilai-nilai parameter *backpropagation neural network* dan algoritma genetika yang berbeda terhadap data uji. Tujuan dari uji coba adalah untuk mengetahui pengaruh parameter yang digunakan pada proses pelatihan. Setelah proses pelatihan selesai dilakukan, maka bobot terbaik yang dihasilkan akan dievaluasi menggunakan sejumlah data uji, kemudian dihitung tingkat kesalahan peramalannya.

Uji pengaruh parameter dilakukan pada sejumlah data latih yaitu data saham selama dua tahun. Uji parameter digunakan untuk mendapatkan bobot terbaik pada

pada parameter tertentu untuk digunakan pada proses peramalan. Pengujian dilakukan dengan menggunakan satu *hidden layer*. *Training Error* maksimal ditentukan 1%. Proses pengujian pada *backpropagation* dilakukan dengan merubah-rubah kombinasi antara nilai momentum dengan *learning rate* yang diujikan yaitu 0,1 hingga 0.001. Uji parameter pertama yang dilakukan adalah pengaruh probabilitas *crossover* dan mutasi terhadap nilai *fitness* dengan sejumlah generasi dan ukuran populasi yang sama. Untuk setiap pengujian dilakukan lima kali percobaan kemuadian diambil nilai rata-ratanya.

Tabel 3.20 Percobaan perngaruh *learning rate* terhadap pelatihan JST

No	*Hidden Unit*	*Learning Rate*	Uji ke-	RMSE (%)	Waktu(detik)
1	4	0.1	1		
			2		
			3		
			4		
			5		
2	4	0.01	1		
			2		
			3		
			4		
			5		

.**Tabel 3.21** Percobaan Pengaruh Perubahan Probabilitas *Crossover* dan Probabilitas Mutasi terhadap Nilai *Fitness*

Probabilitas Mutasi(Pm)	Probabilitas *Crossover* (Pc)			
	0.1	0.3	0.5	0.7
0.1				
0.2				
0.3				
0.4				

Uji parameter selanjutnya ialah pengaruh ukuran populasi dan jumlah generasi terhadap nilai *fitness*. Sama seperti pengaruh probabilitas *crossover* dan mutasi, untuk setiap pengujian dilakukan lima kali percobaan kemudian diambil nilai

rata-ratanya. Tabel untuk pengaruh jumlah generasi terhadap nilai *fitness* dapat dilihat pada tabel 3.22, sedangkan tabel uji untuk pengaruh ukuran populasi terhadap nilai *fitness* dapat dilihat pada tabel 3.23

Tabel 3.22 Percobaan Pengaruh Jumlah Generasi Terhadap Nilai *Fitness*

No	Jumlah Generasi	Fitness rata-rata (pC 90%, pM 30%)
1	10	
2	100	
3	1000	
4	...	
5	N	

Tabel 3.23 Percobaan Pengaruh Ukuran Populasi Terhadap Nilai *Fitness*

No	Ukuran Populasi	Fitness rata-rata (pC 90%, pM 30%)
1	10	
2	20	
3	30	
4	...	
5	N	

Setelah didapatkan parameter genetika yang menghasilkan kromosom dengan *fitness* terbaik. Kromosom tersebut akan digunakan sebagai bobot jaringan syaraf tiruan pada peramalan harga saham. Tabel uji untuk hasil percobaan terhadap data aktual harga saham dapat dilihat pada tabel 3.24.

Tabel 3.24 Hasil Peramalan terhadap Data Aktual Harga Saham

No	Tanggal	Harga *Close* Aktual	Harga *Close* Peramalan	*Error*

Dari tabel 3.24 dapat dihitung selisih harga saham hasil peramalan terhadap harga saham aktual.Kemudian dihitung nilai RMSE untuk mengevaluai kesalahan peramalan sekaligus tingkat akurasi hasil peramalan tersebut.