

**PREDIKSI HARGA SAHAM MENGGUNAKAN METODE  
*BACKPROPAGATION DENGAN OPTIMASI ANT COLONY  
OPTIMIZATION***

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
Memperoleh gelar Sarjana Komputer

Disusun oleh:

David Bernhard

NIM: 155150200111008



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2019

## PENGESAHAN

PREDIKSI HARGA SAHAM MENGGUNAKAN METODE BACKPROPAGATION  
DENGAN OPTIMASI ANT COLONY OPTIMIZATION

### SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
David Bernhard  
NIM: 155150200111008

Skripsi ini telah diuji dan dinyatakan lulus pada  
28 Mei 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Muhammad Tanzil Furqon, S.Kom,  
M.CompSc  
NIP: 198209302008011004

Dosen Pembimbing II

Drs. Muh Arif Rahman, M.Kom  
NIP: 196604231991111001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D  
NIP: 197105182003121001



Scanned with

### **PERNYATAAN ORISINALITAS**

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 9 Mei 2019



David Bernhard

NIM: 155150200111008



Scanned with  
CamScanner

## PRAKATA

Puji syukur atas kehadiran Tuhan Yang Maha Esa akan segala berkat dan anugerah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Prediksi Harga Saham menggunakan Metode *Backpropagation* dengan Optimasi *Ant Colony Optimization*". Penulis menyadari bahwa skripsi ini tak lepas dari bantuan semua pihak baik berupa bimbingan, motivasi, saran, kritik, maupun doa. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Muhammad Tanzil Furqon, S.Kom, M.CompSc selaku dosen pembimbing I yang telah membagikan ilmu dan waktu yang telah diluangkan dalam memberikan bimbingan, kritik, dan saran dalam penggerjaan skripsi ini.
2. Bapak Drs. Muh Arif Rahman, M.Kom selaku dosen pembimbing II yang telah membagikan ilmu dan waktu yang telah diluangkan dalam memberikan bimbingan, kritik, dan saran dalam penggerjaan skripsi ini.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer, Universitas Brawijaya.
4. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer, Universitas Brawijaya.
5. Ibu Ellen Zusy, S.E selaku orang tua penulis yang telah memberi dukungan doa, finansial, dan moril kepada penulis.
6. Sahabat-sahabat penulis, Jason Hartanto, Yosua Martin, dan Yehuda Tri Nugroho yang selalu memberi motivasi dan menjadi tempat dalam berbagi ilmu.
7. Seluruh bapak/ibu dosen dan staff yang telah memberikan banyak ilmu, bantuan, dan dukungan selama menempuh studi di Fakultas Ilmu Komputer, Universitas Brawijaya.
8. Semua pihak yang tidak dapat penulis sebutkan satu-persatu dalam membantu menyelesaikan skripsi ini.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dan dukungan dari beberapa pihak. Selain itu, skripsi ini juga masih jauh dari sempurna karena masih memiliki kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun penelitian ini. Akhir kata penulis berharap skripsi ini dapat memberikan manfaat dan berkah bagi semua pihak.

Malang, 9 Mei 2019

Penulis

atkgdavid21@gmail.com

## ABSTRAK

**David Bernhard, Prediksi Harga Saham menggunakan Metode *Backpropagation* dengan Optimasi *Ant Colony Optimization***

**Pembimbing: Muhammad Tanzil Furqon, S.Kom, M.CompSc dan Drs. Muh Arif Rahman, M.Kom**

Saham merupakan tanda kontribusi penanaman modal seseorang atau pihak terhadap suatu perusahaan atau perseroan terbatas. Pergerakan harga saham berpengaruh terhadap keuntungan dan kerugian yang akan diperoleh investor. Kendalanya, harga saham dapat berubah dalam setiap menit pada hari kerja. Dibutuhkan metode yang mampu memprediksi harga saham dengan akurat dan konsisten, sehingga dapat meminimalkan risiko investasi saham. Disamping kelebihannya, BPNN memiliki kekurangan, seperti lambatnya waktu konvergensi, mudah konvergen ke titik minimum lokal, dan buruknya kemampuan generalisasi. ACO memiliki kelebihan dalam komputasi terdistribusi, umpan balik positif, dan sifat metaheuristik yang dapat memperbaiki kelemahan BPNN. Oleh karena itu, penelitian ini menerapkan ACO-BPNN untuk memprediksi harga saham. Penelitian ini menggunakan data *time series* harga saham Bank Rakyat Indonesia (Persero) Tbk. periode 1 Januari 2018 sampai 31 Desember 2018. ACO berfungsi untuk mengoptimalkan kombinasi nilai *learning rate*, momentum, dan jumlah *hidden node* bagi fase pelatihan BPNN. Lalu bobot, *threshold*, dan jumlah *hidden node* dari fase pelatihan digunakan untuk memprediksi harga saham harian pada suatu periode. Diperoleh kombinasi nilai parameter ACO terbaik, yaitu tetapan siklus semut sebesar 0,8, tetapan pengendali intensitas feromon sebesar 0,1, tetapan pengendali visibilitas sebesar 0,1, tetapan penguapan feromon lokal sebesar 0,5, tetapan penguapan feromon global sebesar 0,1, jumlah semut 5, dan jumlah iterasi 7. Kombinasi tersebut menghasilkan rata-rata MAPE 1,745, sedangkan BPNN hanya mencapai 3,024.

Kata kunci: prediksi, harga saham, optimasi, *ant colony optimization*, *backpropagation*, ACO-BPNN

## ABSTRACT

**David Bernhard, Stock Prices Prediction using the Backpropagation Method with Optimization of Ant Colony Optimization**

**Supervisors: Muhammad Tanzil Furqon, S.Kom, M.CompSc and Drs. Muh Arif Rahman, M.Kom**

*Stocks are a sign of a person's or party's investment contribution to a company or limited liability company. Movement of stock prices affects the profits and losses that will be obtained by the investor. The obstacle is stock prices can change in every minute on weekdays. It takes a method that is able to predict stock prices accurately and consistently, so that it can minimize the risk of stock investment. Besides its advantages, BPNN has shortcoming, such as slow convergence time, easy convergence to local minimum points, and poor generalization capabilities. ACO has advantages in distributed computing, positive feedback, and metaheuristic properties that can improve the weaknesses of BPNN. Therefore, this study applies ACO-BPNN to predict stock prices. This study uses time series data from the stock price of Bank Rakyat Indonesia (Persero) Tbk. period 1 January 2018 until 31 December 2018. ACO serves to optimize the value combination of learning rate, momentum, and number of hidden nodes for BPNN training phase. Then weights, thresholds, and number of hidden nodes from the training phase are used to predict daily stock prices for a period. Best combination of ACO parameter values was obtained, namely the ant cycle constant worth 0.8, the control constant of pheromone intensity worth 0.1, the visibility control constant worth 0.1, the local pheromone evaporation constant worth 0.5, global pheromone evaporation constant worth 0.1, number of ants 5, and number of iterations 7. That combination produces an average of MAPE 1,745, while BPNN only reached 3,024.*

**Keywords:** prediction, stock prices, optimization, ant colony optimization, backpropagation, ACO-BPNN

<b>DAFTAR ISI</b>	
PENGESAHAN .....	.ii
PERNYATAAN ORISINALITAS .....	.iii
PRAKATA.....	.iv
ABSTRAK.....	.v
ABSTRACT.....	.vi
DAFTAR ISI.....	.vii
DAFTAR TABEL.....	.x
DAFTAR GAMBAR.....	.xii
DAFTAR KODE PROGRAM .....	.xiv
DAFTAR LAMPIRAN .....	.xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan .....	2
1.4 Manfaat.....	2
1.5 Batasan Masalah.....	2
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN .....	5
2.1 Kajian Pustaka .....	5
2.2 Saham .....	8
2.2.1 Prediksi Harga Saham.....	9
2.3 Fungsi Aktivasi Sigmoid Biner .....	9
2.4 <i>Backpropagation Neural Network (BPNN)</i> .....	10
2.5 <i>Ant Colony Optimization (ACO)</i> .....	14
2.6 Min – Max Normalization .....	16
2.7 Mean Absolute Percentage Error .....	16
2.8 <i>Ant Colony Optimization – Backpropagation Neural Network</i> .....	17
BAB 3 METODOLOGI PENELITIAN .....	20
3.1 Studi Literatur .....	20
3.2 Pengumpulan Data .....	21

3.3 Penggunaan Perangkat .....	22
3.4 Perancangan .....	22
3.5 Implementasi .....	23
3.6 Pengujian dan Analisis .....	23
3.7 Penarikan Kesimpulan .....	23
BAB 4 PERANCANGAN.....	24
4.1 Definisi Masalah.....	24
4.2 Perancangan Algoritma .....	24
4.2.1 Hasilkan Parameter <i>Backpropagation Neural Network</i> .....	26
4.2.2 <i>Training Backpropagation Neural Network</i> .....	29
4.2.3 <i>Update Feromon Global</i> .....	34
4.2.4 Simpan <i>Global Best Ant Colony Optimization</i> .....	36
4.2.5 <i>Testing Backpropagation Neural Network</i> .....	37
4.3 Perhitungan Manual .....	40
4.3.1 Perhitungan <i>Ant Colony Optimization</i> .....	40
4.3.2 Perhitungan <i>Backpropagation Neural Network</i> .....	45
4.4 Perancangan Antarmuka .....	56
4.4.1 Perancangan Antarmuka Halaman Data Mentah .....	56
4.4.2 Perancangan Antarmuka Halaman Pelatihan ACO-BPNN .....	56
4.4.3 Perancangan Antarmuka Halaman Pengujian ACO-BPNN.....	59
4.5 Perancangan Pengujian .....	62
4.5.1 Perancangan Skenario Pengujian Nilai Tetapan Siklus Semut....	62
4.5.2 Perancangan Skenario Pengujian Nilai Tetapan Pengendali Intensitas Feromon .....	63
4.5.3 Perancangan Skenario Pengujian Nilai Tetapan Pengendali Visibilitas .....	64
4.5.4 Perancangan Skenario Pengujian Nilai Tetapan Penguapan Feromon Lokal.....	64
4.5.5 Perancangan Skenario Pengujian Nilai Tetapan Penguapan Feromon Global.....	65
4.5.6 Perancangan Skenario Pengujian Jumlah Semut ACO .....	66
4.5.7 Perancangan Skenario Pengujian Jumlah Iterasi ACO .....	66
4.5.8 Perancangan Skenario Pengujian ACO-BPNN dan BPNN .....	66

BAB 5 IMPLEMENTASI .....	68
5.1 Implementasi Algoritma .....	68
5.1.1 Implementasi Algoritma <i>Ant Colony Optimization</i> .....	68
5.1.2 Implementasi Algoritma <i>Backpropagation Neural Network</i> .....	73
5.2 Implementasi Antarmuka .....	77
5.2.1 Implementasi Antarmuka Halaman Data Mentah.....	78
5.2.2 Implementasi Antarmuka Halaman Pelatihan ACO-BPNN .....	78
5.2.3 Implementasi Antarmuka Halaman Pengujian ACO-BPNN.....	79
BAB 6 Pengujian dan analisis .....	82
6.1 Hasil dan Analisis Pengujian Parameter ACO .....	82
6.1.1 Pengujian Nilai Tetapan Siklus Semut.....	82
6.1.2 Pengujian Nilai Tetapan Pengendali Intensitas Feromon .....	84
6.1.3 Pengujian Nilai Tetapan Pengendali Visibilitas .....	85
6.1.4 Pengujian Nilai Tetapan Penguapan Feromon Lokal .....	86
6.1.5 Pengujian Nilai Tetapan Penguapan Feromon Global .....	88
6.1.6 Pengujian Jumlah Semut .....	89
6.1.7 Pengujian Jumlah Iterasi .....	91
6.2 Hasil dan Analisis Pengujian ACO-BPNN dan BPNN.....	92
BAB 7 PENUTUP .....	95
7.1 Kesimpulan.....	95
7.2 Saran .....	95
DAFTAR REFERENSI .....	96
LAMPIRAN A DATA MENTAH .....	98
LAMPIRAN B HASIL PELATIHAN DAN PENGUJIAN.....	108

## DAFTAR TABEL

Tabel 2.1 Kajian pustaka .....	6
Tabel 2.2 Tabel interpretasi nilai MAPE .....	17
Tabel 3.1 Sampel data mentah .....	21
Tabel 3.2 Sampel data.....	21
Tabel 4.1 Nilai parameter ACO.....	40
Tabel 4.2 Batas parameter BPNN.....	41
Tabel 4.3 Intensitas feromon awal.....	41
Tabel 4.4 Nilai probabilitas perpindahan semut.....	42
Tabel 4.5 Nilai probabilitas kumulatif <i>node 10</i> .....	42
Tabel 4.6 Intensitas feromon setelah <i>update feromon lokal</i> .....	43
Tabel 4.7 Intensitas feromon setelah <i>update feromon global</i> .....	45
Tabel 4.8 Data latih .....	45
Tabel 4.9 Nilai minimum dan maksimum data latih .....	45
Tabel 4.10 Data latih ternormalisasi .....	46
Tabel 4.11 Bobot menuju <i>hidden layer</i> .....	46
Tabel 4.12 Bobot menuju <i>output layer</i> .....	46
Tabel 4.13 <i>Threshold</i> pada <i>hidden layer</i> .....	47
Tabel 4.14 <i>Threshold</i> pada <i>output layer</i> .....	47
Tabel 4.15 Hasil perhitungan $z_{net_j}$ .....	47
Tabel 4.16 Hasil aktivasi $z_{net_j}$ .....	47
Tabel 4.17 Hasil perhitungan $y_{net_k}$ .....	48
Tabel 4.18 Hasil aktivasi $y_{net_k}$ .....	48
Tabel 4.19 Hasil perhitungan $\delta_k$ .....	49
Tabel 4.20 Hasil perhitungan $\delta_j$ .....	49
Tabel 4.21 Hasil perhitungan $\Delta w_{kj}$ .....	50
Tabel 4.22 Hasil perhitungan $\Delta v_{ji}$ .....	50
Tabel 4.23 Hasil <i>update</i> $w_{kj}$ .....	50
Tabel 4.24 Hasil <i>update</i> $v_{ji}$ .....	51
Tabel 4.25 Hasil <i>update</i> $\theta_k$ .....	51
Tabel 4.26 Hasil <i>update</i> $\theta_j$ .....	52

Tabel 4.27 Hasil perhitungan RMSE .....	52
Tabel 4.28 Data uji .....	52
Tabel 4.29 Nilai minimum dan maksimum data uji .....	53
Tabel 4.30 Data uji ternormalisasi .....	53
Tabel 4.31 Hasil perhitungan $z_{netj}$ .....	53
Tabel 4.32 Hasil aktivasi $z_{netj}$ .....	54
Tabel 4.33 Hasil perhitungan $y_{netk}$ .....	54
Tabel 4.34 Hasil aktivasi $y_{netk}$ .....	54
Tabel 4.35 Hasil denormalisasi.....	55
Tabel 4.36 Nilai MAPE .....	55
Tabel 4.37 Rancangan pengujian nilai tetapan siklus semut .....	63
Tabel 4.38 Rancangan pengujian nilai tetapan pengendali intensitas feromon ..	63
Tabel 4.39 Rancangan pengujian nilai tetapan pengendali visibilitas .....	64
Tabel 4.40 Rancangan pengujian nilai tetapan penguapan feromon lokal .....	64
Tabel 4.41 Rancangan pengujian nilai tetapan penguapan feromon global .....	65
Tabel 4.42 Rancangan pengujian jumlah semut ACO .....	66
Tabel 4.43 Rancangan pengujian jumlah iterasi ACO .....	66
Tabel 4.44 Rancangan pengujian ACO-BPNN dan BPNN .....	67
Tabel 6.1 Hasil pengujian nilai tetapan siklus semut .....	83
Tabel 6.2 Hasil pengujian nilai tetapan pengendali intensitas feromon .....	84
Tabel 6.3 Hasil pengujian nilai tetapan pengendali visibilitas .....	85
Tabel 6.4 Hasil pengujian nilai tetapan penguapan feromon lokal .....	87
Tabel 6.5 Hasil pengujian nilai tetapan penguapan feromon global .....	88
Tabel 6.6 Hasil pengujian jumlah semut .....	90
Tabel 6.7 Hasil pengujian jumlah iterasi .....	91
Tabel 6.8 Hasil pengujian MAPE ACO-BPNN dan BPNN.....	92
Tabel 6.9 Hasil pengujian waktu komputasi ACO-BPNN dan BPNN .....	94

## DAFTAR GAMBAR

Gambar 2.1 Arsitektur <i>backpropagation neural network</i> .....	10
Gambar 2.2 Ilustrasi cara kerja <i>ant colony optimization</i> .....	14
Gambar 3.1 Diagram alir metodologi penelitian .....	20
Gambar 4.1 Diagram alir algoritma ACO-BPNN .....	25
Gambar 4.2 Diagram alir optimasi ACO .....	27
Gambar 4.3 Diagram alir menentukan node selanjutnya.....	28
Gambar 4.4 Diagram alir konversi nilai parameter BPNN .....	29
Gambar 4.5 Diagram alir <i>training</i> BPNN .....	30
Gambar 4.6 Diagram alir tambah data <i>training</i> .....	31
Gambar 4.7 Diagram alir fase <i>training</i> .....	32
Gambar 4.8 Diagram alir proses <i>feedforward</i> .....	33
Gambar 4.9 Diagram alir proses <i>backpropagate</i> .....	34
Gambar 4.10 Diagram alir <i>update feromon global</i> .....	36
Gambar 4.11 Diagram alir simpan <i>global best ACO</i> .....	37
Gambar 4.12 Diagram alir <i>testing</i> BPNN.....	38
Gambar 4.13 Diagram alir tambah data <i>testing</i> .....	39
Gambar 4.14 Diagram alir fase <i>testing</i> .....	40
Gambar 4.15 Perancangan antarmuka halaman data mentah .....	56
Gambar 4.16 Perancangan antarmuka <i>tab menu</i> prapelatihan aktif .....	57
Gambar 4.17 Perancangan antarmuka <i>tab menu</i> proses pelatihan aktif.....	58
Gambar 4.18 Perancangan antarmuka <i>tab menu</i> hasil pelatihan aktif .....	59
Gambar 4.19 Perancangan antarmuka <i>tab menu</i> prapengujian aktif .....	60
Gambar 4.20 Perancangan antarmuka <i>tab menu</i> hasil pengujian aktif .....	61
Gambar 4.21 Perancangan antarmuka <i>tab menu</i> detail harga dan grafik aktif ...	62
Gambar 5.1 Implementasi antarmuka halaman data mentah .....	78
Gambar 5.2 Implementasi antarmuka <i>tab menu</i> prapelatihan aktif.....	78
Gambar 5.3 Implementasi antarmuka <i>tab menu</i> proses pelatihan aktif.....	79
Gambar 5.4 Implementasi antarmuka <i>tab menu</i> hasil pelatihan aktif.....	79
Gambar 5.5 Implementasi antarmuka <i>tab menu</i> prapengujian aktif .....	80
Gambar 5.6 Implementasi antarmuka <i>tab menu</i> hasil pengujian aktif .....	80

Gambar 5.7 Implementasi antarmuka <i>tab menu</i> detail harga dan grafik aktif ....	81
Gambar 6.1 Grafik hasil pengujian nilai tetapan siklus semut.....	83
Gambar 6.2 Grafik hasil pengujian nilai tetapan pengendali intensitas feromon	85
Gambar 6.3 Grafik hasil pengujian nilai tetapan pengendali visibilitas.....	86
Gambar 6.4 Grafik hasil pengujian nilai tetapan penguapan feromon lokal.....	88
Gambar 6.5 Grafik hasil pengujian nilai tetapan penguapan feromon global.....	89
Gambar 6.6 Grafik hasil pengujian jumlah semut .....	90
Gambar 6.7 Grafik hasil pengujian jumlah iterasi.....	92
Gambar 6.8 Grafik hasil pengujian MAPE ACO-BPNN dan BPNN .....	93



## DAFTAR KODE PROGRAM

Kode Program 5.1 Menentukan <i>node</i> awal .....	68
Kode Program 5.2 Menentukan <i>next node</i> .....	69
Kode Program 5.3 <i>Update</i> feromon lokal .....	70
Kode Program 5.4 Konversi nilai parameter BPNN.....	71
Kode Program 5.5 <i>Update</i> feromon global .....	72
Kode Program 5.6 Menyimpan <i>global best</i> ACO terakhir.....	73
Kode Program 5.7 <i>Min-max normalization</i> .....	73
Kode Program 5.8 Proses <i>feedforward</i> .....	74
Kode Program 5.9 Proses <i>backpropagate</i> .....	75
Kode Program 5.10 Menghitung <i>root mean square error</i> .....	76
Kode Program 5.11 <i>Min-max denormalization</i> .....	77
Kode Program 5.12 Menghitung <i>mean absolute percentage error</i> .....	77

## DAFTAR LAMPIRAN

LAMPIRAN A DATA MENTAH .....	98
A.1 Data.....	98
A.2 Grafik .....	107
LAMPIRAN B HASIL PELATIHAN DAN PENGUJIAN.....	108
B.1 Hasil Pelatihan .....	108
B.2 Hasil Pengujian.....	109
B.3 Grafik Hasil Pengujian.....	109



## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Saham merupakan tanda kontribusi penanaman modal seseorang atau pihak terhadap suatu perusahaan atau perseroan terbatas. Investasi saham memiliki peran dalam menambah devisa negara, pemerataan pendapatan, hingga terciptanya lapangan kerja baru. Investor saham dapat meraih 2 jenis keuntungan, antara lain dividen dan *capital gain*. *Capital gain* didapatkan saat investor menjual saham lebih mahal dari harga beli, jika sebaliknya, maka *capital loss* yang didapatkan. Sehingga investor perlu mengetahui bilamana waktunya membeli atau menjual saham, agar terhindar dari *capital loss* hingga risiko likuiditas. Kendalanya, harga saham dapat berubah dalam setiap menit pada hari kerja. Dibutuhkan metode yang optimal dalam melakukan prediksi harga saham berdasarkan data historis yang tersedia.

Terdapat berbagai penelitian dalam melakukan prediksi harga saham menggunakan metode yang bervariasi, salah satunya adalah *Backpropagation Neural Network* (BPNN). Performa metode tersebut dalam melakukan prediksi harga saham Tata Power Co Ltd memiliki akurasi 91,52%. Akurasi tersebut lebih baik daripada *Moving Average Convergence Divergence* (MACD) dan *Moving Average* (MA) yang hanya mencapai 54,17% dan 33,33% (Khan, Bandopadhyaya, & Sharma, 2008). Selain itu, dalam peramalan kebutuhan energi listrik Indonesia hingga tahun 2022, MAPE dari BPNN 2,8027% dan logika fuzzy 8,2413% (Nurkholid, Sukmadi, & Nugroho, 2014). Hasil ini disebabkan oleh kelebihan BPNN dalam hal pembelajaran adaptif, swa-organisme dan toleransi kesalahan (Wang, et al., 2010). Namun, metode tersebut memiliki beberapa kekurangan seperti lambatnya waktu konvergensi, mudah konvergen ke titik minimum lokal, dan buruknya kemampuan generalisasi (Cao, et al., 2008). Sehingga diperlukan algoritma optimasi yang mampu mengatasi kelemahan tersebut.

*Ant Colony Optimization* (ACO) telah meraih performa baik dalam pemecahan *traveling salesman problem* (TSP) dan masalah optimasi lainnya. Keunggulan metode ini terletak pada komputasi terdistribusi, umpan balik positif, dan sifat metaheuristik yang dapat memperbaiki kelemahan dari BPNN (Wang, et al., 2010). Metode tersebut memiliki efisiensi tertinggi dibandingkan dengan GA dan PSO untuk mengoptimalkan tata letak fasilitas konstruksi. Lalu mengenai efektifitas dan konsistensi, ketiga metode memiliki kualitas yang sama (Adrian, Utamima, & Wang, 2014). Metode tersebut juga cocok untuk komputasi yang besar dan/atau kompleks, masalah yang membutuhkan solusi berkualitas tinggi, dan mudah dihibridakan (Haroun, Jamal, & Hicham, 2015). Oleh karena itu, ACO dipilih untuk mengoptimalkan BPNN.

Kombinasi ACO dan BPNN telah digunakan untuk penelitian optimasi identifikasi kecepatan motor dalam kontrol torsi langsung (DTC). Penelitian ini menggunakan 2000 data sampel dan metode tersebut berhasil mencapai nilai *error* 0,0049 setelah melakukan 536 iterasi. Sebaliknya, BPNN mencapai nilai *error*

0,3986 setelah melakukan 2000 iterasi (Cao, et al., 2008). Melalui kemampuan algoritma *swarm* dalam meningkatkan akurasi prediksi sekelompok *trader*, diperlukan penelitian untuk memvalidasi manfaat algoritma tersebut di bidang keuangan. Tentunya menggunakan algoritma *swarm* yang disesuaikan dengan masalah yang ada (Rosenberg, Pescetelli, & Willcox, 2017). Pada penelitian ini, ACO berfungsi untuk mengoptimalkan nilai parameter *learning rate*, momentum, dan jumlah *hidden node* milik BPNN. Dengan demikian, diharapkan ACO dapat meningkatkan performa BPNN yang telah memiliki koreksi bobot dan *threshold* berdasarkan nilai *error*. Kemudian kedua metode digunakan untuk memprediksi harga saham Bank Rakyat Indonesia (Persero) Tbk.

## 1.2 Rumusan Masalah

Penelitian ini memiliki permasalahan yang akan diselesaikan yaitu:

1. Bagaimana menghasilkan MAPE terbaik dari metode ACO-BPNN dalam melakukan prediksi harga saham?
2. Bagaimana pengaruh ACO dalam menurunkan MAPE BPNN untuk melakukan prediksi harga saham?

## 1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini antara lain:

1. Menghasilkan MAPE terbaik dari metode ACO-BPNN dalam melakukan prediksi harga saham
2. Mengetahui pengaruh ACO dalam menurunkan MAPE BPNN untuk melakukan prediksi harga saham

## 1.4 Manfaat

Adapun manfaat yang dihasilkan dari penelitian ini meliputi:

1. Memberikan informasi mengenai harga saham di masa yang akan datang, sehingga membantu investor dalam menentukan waktu menjual maupun membeli saham
2. Memberikan prediksi yang optimal dalam hal ketelitian dan konsistensi, sehingga mampu meminimalkan risiko kerugian investasi saham, bahkan saat perubahan harga fluktuatif
3. Memberikan alternatif baru bagi para investor dalam melakukan prediksi harga saham pada masa yang akan datang

## 1.5 Batasan Masalah

Terdapat batasan dalam ruang lingkup masalah penelitian sebagai berikut:

1. Data yang digunakan untuk pelatihan dan pengujian dalam penelitian ini adalah data harga saham milik Bank Rakyat Indonesia (Persero) Tbk. pada periode 1 Januari 2018 sampai 31 Desember 2018

2. Data pelatihan merupakan harga saham yang merupakan *close price* pada empat bulan berurutan dengan data pengujian *close price* pada satu bulan berikutnya
3. Parameter *Backpropagation Neural Network* yang dioptimasi adalah *learning rate*, momentum, dan jumlah *hidden node*
4. Pengujian dilakukan menggunakan evaluasi MAPE
5. Periode prediksi harga saham yang mencakup 1 bulan berikutnya tersusun dari hasil prediksi *close price* harian

## 1.6 Sistematika Pembahasan

Bagian ini menjelaskan struktur penulisan dari bab pendahuluan hingga bab penutup. Dengan demikian, diharapkan dapat membantu pembaca untuk memahami sistematika pembahasan pada laporan penelitian ini.

### BAB 1

#### PENDAHULUAN

Menjelaskan latar belakang, rumusan masalah, tujuan, manfaat, dan batasan masalah penelitian ini. Kemudian tentunya penjelasan mengenai sistematika pembahasan laporan penelitian.

### BAB 2

#### LANDASAN KEPUSTAKAAN

Menguraikan kajian pustaka dan beberapa landasan teori, seperti saham, fungsi aktivasi sigmoid biner, BPNN, ACO, *min-max normalization*, MAPE, dan ACO-BPNN.

### BAB 3

#### METODOLOGI

Menjelaskan metodologi yang digunakan guna mencapai tujuan penelitian. Metodologi tersusun dari studi literatur, pengumpulan data, penggunaan perangkat, perancangan, implementasi, pengujian dan analisis, serta penarikan kesimpulan.

### BAB 4

#### PERANCANGAN

Memaparkan definisi masalah, perancangan algoritma, perhitungan manual, perancangan antarmuka, dan perancangan pengujian sebagai pedoman untuk pengerjaan bab selanjutnya.

### BAB 5

#### IMPLEMENTASI

Menyatakan implementasi algoritma ACO-BPNN dalam kode program dan menampilkan implementasi antarmukanya.

### BAB 6

#### PENGUJIAN DAN ANALISIS

Memperlihatkan dan menganalisis hasil pengujian guna menjawab rumusan masalah penelitian. Pengujian dilakukan dua kali, yaitu pengujian parameter ACO dan pengujian ACO-BPNN dan BPNN.

## BAB 7

## PENUTUP

Pada bagian akhir, seluruh hasil penelitian diringkas berdasarkan rumusan masalah penelitian guna menjawabnya. Kemudian disampaikan saran peneliti untuk perbaikan dan pengembangan penelitian selanjutnya.



## BAB 2 LANDASAN KEPUSTAKAAN

Dalam melakukan penelitian dibutuhkan landasan teori dan kajian pustaka sebagai referensi, sehingga setiap teori yang dipakai dapat dipertanggungjawabkan.

### 2.1 Kajian Pustaka

Penelitian ini dilakukan berdasarkan referensi dari penelitian sebelumnya yang memiliki persamaan. Persamaan-persamaan tersebut menjadi kerangka teoritik pada penelitian berjudul “Prediksi Harga Saham menggunakan Metode *Backpropagation* dengan Optimasi *Ant Colony Optimization*” ini. Selain persamaan tentunya penelitian ini memiliki perbedaan dari penelitian sebelumnya, sehingga keunikan penelitian ini tetap terjaga. Perbedaan tersebut juga membuktikan bahwa penelitian ini tidak menjiplak penelitian sebelumnya. Persamaan dan perbedaan penelitian ini dibandingkan penelitian sebelumnya secara lengkap dapat dilihat pada Tabel 2.1.

Salah satu penelitian prediksi harga saham yang telah dilakukan terdapat pada Konferensi Nasional Sistem dan Informatika di Bali tahun 2008. Penelitian tersebut menggunakan data yang berasal dari Bursa Efek Surabaya. Arsitektur JST pada penelitian ini menggunakan 1 *input layer*, 1 *hidden layer*, dan 1 *output layer*. Konfigurasi parameter terbaik untuk pelatihan menghasilkan jumlah *hidden node* 50, *learning rate* 0,95, nilai momentum 0,95, dan toleransi MAPE 3,97%. Dengan tahun 1998 sebagai data pelatihan dan tahun 1999 sebagai data uji, BPNN berhasil mencapai nilai MAPE terkecil. Masing-masing MAPE dari BPNN, SMA, WMA, EMA, MACD, dan StochRSI adalah 2,7%, 9,45%, 18,32%, 12,03%, 12,03%, dan 9,04% (Setiawan, 2008). Berikutnya penelitian mengenai prediksi harga saham Tata Power Co Ltd menggunakan data pada periode 1 Januari 2004 sampai 29 Desember 2006. Arsitektur BPNN penelitian tersebut terdiri dari 2 *node input layer*, 2 *node hidden layer*, dan 1 *node output layer*. Dengan harga saham periode 2 Januari 2007 sampai 30 Maret 2007 sebagai data uji, BPNN memiliki tingkat akurasi 91,52%. Sedangkan MACD dan MA memiliki akurasi masing-masing 54,17% dan 33,33% (Khan, Bandopadhyaya, & Sharma, 2008).

Selain untuk melakukan prediksi harga saham, BPNN juga dipakai dalam memprediksi kebutuhan listrik jangka panjang di Indonesia. Penelitian ini menggunakan data *time series* 2004 sampai 2012 sebagai data pelatihan. Arsitekturnya terdiri dari 1 *input layer*, 2 *hidden layer*, dan 1 *output layer*. Terdapat 9 *node input* yang terdiri dari pelanggan PLN jenis rumah tangga, pelanggan PLN jenis industri, pelanggan PLN jenis bisnis, pelanggan PLN jenis sosial, pelanggan PLN jenis gedung kantor pemerintah, pelanggan PLN jenis penerangan jalan umum, jumlah penduduk/populasi, rasio elektrifikasi, dan pertumbuhan ekonomi. BPNN memprediksi 242.120 Gwh dan Logika *Fuzzy* memprediksi 215.203 Gwh untuk kebutuhan listrik tahun 2022. Dalam melakukan prediksi untuk tahun 2013 sampai 2022, hasil nilai *error* rata-rata BPNN sebesar 2,8027% dan Logika *Fuzzy* sebesar 8,2413% (Nurkholid, Sukmadi, & Nugroho, 2014).

ACO sendiri pernah digunakan dalam penelitian yang bertujuan untuk mengoptimalkan penempatan dan tata letak fasilitas di area konstruksi demi meningkatkan efisiensi, produktivitas, dan keselamatan kerja. Tiga metode metaheuristik dipakai untuk meminimalkan biaya konstruksi dan biaya interaktif yang disebabkan oleh tata letak fasilitas di area konstruksi. Teknik *craziness concept*, *cross-mutate*, dan *scramble mutation* digunakan untuk meningkatkan keragaman solusi dan menjaga agoritma terjebak pada *local optima*. Parameter optimal dari setiap algoritma ditentukan melalui *design of experiment approach* (DOE). Terdapat dua studi kasus yang digunakan untuk membandingkan efektifitas, efisiensi, dan konsistensi ketiga algoritma menggunakan *one way ANOVA test*. Data yang digunakan berasal dari jurnal *American Society of Civil Engineering*. Dalam hal efektifitas dan konsistensi, GA, PSO, dan ACO memiliki performa yang sama. Namun, dalam hal efisiensi, ACO memiliki waktu konvergensi tercepat diikuti GA lalu PSO (Adrian, Utamima, & Wang, 2014).

Metode ACO-BPNN telah diimplementasikan dalam salah satu penelitian pada kongres dunia mengenai kontrol cerdas dan otomatisasi di Chongqing, China. Penelitian ini memperkenalkan pemeringkatan bobot pada algoritma ACO dasar, modifikasi ACO lalu digunakan untuk mengoptimasi bobot dan *threshold* dari BPNN. Pada fase pelatihan BPNN, ACOrw dapat mengatasi kekurangan BPNN dalam hal mudah jatuh di *local optima* dan mempercepat waktu konvergensi. Kemudian metode ACOrw-BP digunakan untuk mengidentifikasi kecepatan motor dalam kontrol torsi langsung (DTC). Dalam pelatihan BPNN, nilai *error* dapat dikendalikan lewat jumlah node dalam *hidden layer*. Paramater BPNN yang dipakai, antara lain 30 *node hidden layer*, *learning rate* sebesar 0,5, momentum sebesar 0,2, dan toleransi kesalahan minimum 0,005. Dengan 2000 data latih, waktu iterasi dan nilai *error* BPNN 2000 kali dan 0,3986, ACO-BPNN 536 dan 0,0049, serta ACOrw-BPNN 129 dan 0,0048 (Cao, et al., 2008).

**Tabel 2.1 Kajian pustaka**

No.	Penulis	Objek	Metode	Hasil
1	Li Wang, Dong-qing Wang, Jia-jun Zhu, Xiang Zhao	Kadar silikon dalam logam panas	<i>Ant Colony Algorithm,</i> <i>Backpropagation</i>	Perbandingan akurasi dan <i>error</i> antara ACA-BPNN dengan ACA serta parameter optimal dari ACA
2	Asif Ullah Khan, T. K. Bandopadhyay, Sudhir Sharma	Harga saham Tata Power Co Ltd	<i>Moving Average,</i> <i>Moving Average</i> <i>Convergence</i> <i>Divergence,</i> <i>Backpropagation</i> , <i>Genetic</i> <i>Algorithm-</i> <i>Backpropagation</i>	Perbandingan akurasi antara MA, MACD, BPNN, dan GA-BPNN

3	Wahyudi Setiawan	Harga saham yang tercatat pada Bursa Efek Surabaya	<i>Backpropagation, Simple Moving Average, Weighted Moving Average, Exponential Moving Average, Moving Average Convergence Divergence, StochRSI</i>	Parameter optimal BPNN dan perbandingan MAPE antara BPNN, SMA, WMA, EMA, MACD, dan StochRSI
4	Nahar Nurkholiq, Tejo Sukamdi, Agung Nugroho	Kebutuhan energi listrik jangka panjang Indonesia	<i>Logika Fuzzy, Backpropagation</i>	Perbandingan error antara Logika Fuzzy dan BPNN
5	Angelia Melani Adrian, Amalia Utamina, Kung-Jeng Wang	Tata letak fasilitas di area konstruksi	<i>Genetic Algorithm, Particle Swarm Optimization, Ant Colony Optimization</i>	Perbandingan efektifitas, efesiensi, dan konsistensi antara GA, PSO, ACO
6	Louis Rosenberg, Niccolo Pescetelli, Gregg Willcox	Tren mingguan SPX, GLD, GDX, Crude Oil	<i>Individual, Crowd, Swarm Based Prediction</i>	Perbandingan akurasi prediksi ketiga metode serta signifikansi pemakaian metode Crowd dan Swarm
7	Sabry Ahmed Haroun, Benhra Jamal, El Hassani Hicham	Travelling salesman problem	<i>Genetic Algorithm, Ant Colony Optimization</i>	Perbandingan jarak, error rate, dan waktu tempuh antara GA dengan ACO
8	Chengzhi Cao, Yifan Wang, Lichao Jia, Yang Liu	Kecepatan motor dalam kontrol torsi langsung (DTC)	<i>Ant Colony Optimization, Backpropagation</i>	Perbandingan efesiensi iterasi dan nilai error antara BPNN, ACO-BPNN, dan ACOrw-BPNN

9	Penulis	Harga saham Bank Rakyat Indonesia (Persero) Tbk.	<i>Ant Colony Optimization, Backpropagation</i>	Parameter dan MAPE optimal ACO-BPNN serta perbandingan MAPE dengan BPNN
---	---------	---	---	--

## 2.2 Saham

Saham merupakan surat berharga sebagai tanda kepemilikan seseorang atau badan atas suatu perusahaan. Pembelian saham atau berinvestasi saham mempunyai arti membeli sebagian kepemilikan atas perusahaan tersebut. Porsi kepemilikan ditentukan oleh besarnya nilai modal yang ditanamkan di perusahaan tersebut. Saham sebagai produk investasi mempunyai keunggulan yaitu, likuiditas tinggi, tanpa pajak rutin, tanpa biaya perawatan, dan informasi harga secara *real-time*. Dalam melakukan investasi saham terdapat beberapa keuntungan yang dapat diraih yaitu:

1. Dividen

Dividen adalah keuntungan yang dibagikan sebuah perusahaan kepada para pemegang sahamnya berdasarkan banyaknya saham yang dimiliki.

2. *Capital Gain*

*Capital Gain* adalah keuntungan yang didapat investor saat harga jual saham lebih mahal daripada harga saat membeli saham.

Selain keuntungan di atas, investasi saham juga memiliki risiko kerugian antara lain:

1. *Capital Loss*

*Capital Loss* adalah kebalikan dari *Capital Gain*, yaitu saat harga jual saham lebih murah daripada harga saat membeli saham.

2. Risiko Likuiditas

Risiko Likuiditas adalah kerugian yang dimiliki investor ketika perusahaan dimana ia berinvestasi mengalami kebangkrutan atau dibubarkan. Sehingga aset investor akan dapat dikembalikan setelah perusahaan menyelesaikan seluruh kewajibannya.

Pergerakan harga saham yang dapat berubah sewaktu-waktu menjadi salah satu penyebab keuntungan dan kerugian investor. Terdapat 3 macam pergerakan harga saham antara lain:

1. *Bullish*, yaitu kondisi dimana harga saham mengalami kenaikan dari waktu ke waktu
2. *Bearish*, yaitu kondisi dimana harga saham mengalami penurunan dari waktu ke waktu

3. *Sideways*, yaitu kondisi dimana harga saham mengalami kenaikan dan penurunan yang sepadan dari waktu ke waktu. Pergerakan ini mencerminkan perubahan harga yang stabil karena membentuk pola grafik mendatar

### 2.2.1 Prediksi Harga Saham

Prediksi harga saham merupakan kegiatan menganalisa faktor-faktor yang mempengaruhi pergerakan harga saham di masa yang akan datang. Terdapat 2 teknik analisis dalam melakukan prediksi harga saham, yaitu analisis fundamental dan analisis teknikal. Analisis teknikal meperhatikan pergerakan harga saham dari waktu ke waktu. Asumsi dalam analisis ini adalah harga suatu saham hanya ditentukan oleh elastisitas penawaran dan permintaan atas saham tersebut (WBBA, & Pratomo, 2013). Dalam perkembangannya para analis teknikal modern membuat grafik batang yang dapat menampilkan *open price*, *high price*, *low price*, dan *close price*. Sebaliknya, analisis fundamental mengidentifikasi apakah harga saham lebih murah atau lebih mahal dibandingkan harga yang seharusnya pada suatu waktu tertentu. Analisis ini mementukan apakah harga saham sudah mencerminkan nilai intrinsiknya atau belum (Halim, 2005, p.21 disitasi dalam WBBA, & Pratomo, 2013).

### 2.3 Fungsi Aktivasi Sigmoid Biner

Fungsi aktivasi merupakan *node* yang ditambahkan untuk memproses lebih lanjut hasil penjumlahan dari *layer* sebelumnya. Ada beberapa jenis fungsi aktivasi, antara lain logsig, tansig, dan purelin. Penelitian ini memakai fungsi aktivasi logsig, yaitu fungsi aktivasi sigmoid biner. Fungsi aktivasi tersebut dipilih karena kemudahan implementasi dan diferensiasi (Jumarwanto, 2009). Fungsi aktivasi dan turunannya dinyatakan pada Persamaan 2.1 dan 2.2.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.1)$$

$$f'(x) = f(x)(1 - f(x)) \quad (2.2)$$

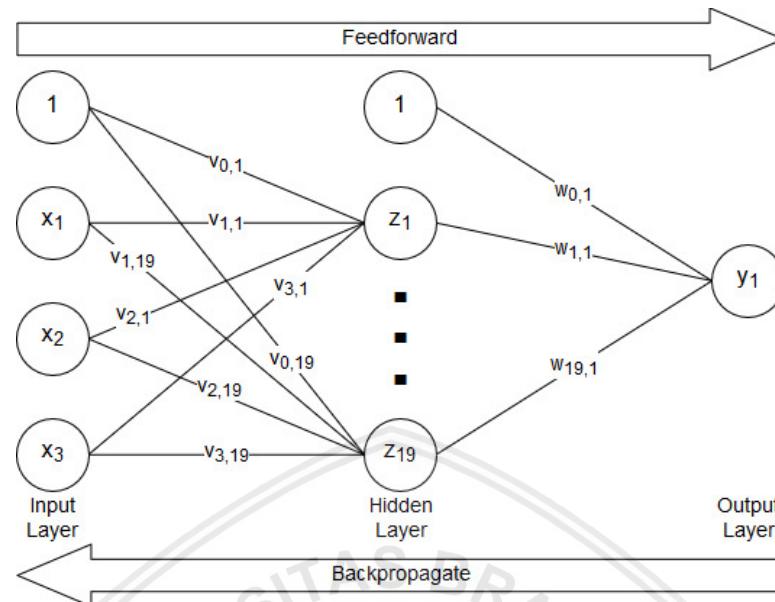
Keterangan:

$f(x)$  = nilai hasil aktivasi

$f'(x)$  = nilai hasil deaktivasi

$e$  = logaritma alami

## 2.4 Backpropagation Neural Network (BPNN)



Gambar 2.1 Arsitektur backpropagation neural network

*Backpropagation neural network* (BPNN) adalah salah satu metode JST yang mempunyai beberapa lapisan koneksi yang terdiri dari beberapa *node*. Setiap *node* di lapisan sebelumnya dapat terhubung dengan satu dari setiap *node* di lapisan selanjutnya. BPNN melakukan propagasi mundur setelah propagasi maju untuk memperbaiki bobot, proses ini berulang hingga mencapai iterasi maksimum atau *error* minimum. BPNN mempunyai kelebihan dalam hal pembelajaran adaptif, swa-organisme, dan toleransi kesalahan yang membuat algoritma ini sukses di berbagai bidang (Wang, et al., 2010). Metode tersebut juga relatif mudah untuk diterapkan, meskipun dalam menentukan konfigurasi jaringan dan parameter optimalnya tidaklah mudah (Khan, Bandopadhyaya, & Sharma, 2008). Selain itu, metode ini memiliki kekurangan, seperti lambatnya waktu konvergensi, mudah konvergen ke *local optima*, dan kemampuan generalisasi yang buruk (Cao, et al., 2008).

Dalam implementasinya BPNN memiliki langkah-langkah sebagai berikut (Negnevitsky, 2005):

1. Inisialisasi nilai bobot dan *threshold* dengan nilai acak beserta inisialisasi jumlah *layer*, jumlah *node* setiap *layer*, *learning rate*, momentum, iterasi maksimal, dan toleransi *error* maksimal
2. Lakukan langkah 3 sampai 14 apabila kondisi *stop* belum terpenuhi
3. Lakukan langkah 4 sampai 13 untuk setiap pasangan data latih

**Langkah untuk propagasi maju:**

4. Menghitung hasil pada unit tersembunyi,  $z_j$  ( $j=1,2,\dots,n$ )

$$z_{netj} = v_{j0} + \sum_{i=1}^n x_i v_{ji} - \theta_j \quad (2.3)$$

Menghitung aktivasi dengan fungsi sigmoid biner berdasarkan Persamaan 2.1

$$z_j = f(z_{netj}) = \frac{1}{1 + e^{-z_{netj}}} \quad (2.4)$$

Keluaran dari fungsi aktivasi tersebut dikirim ke unit lapisan tersembunyi

Keterangan:

$z_{netj}$  = hasil penjumlahan *input layer*

$x_i$  = nilai *node input layer*

$v_{ji}$  = bobot antara *input layer* dan *hidden layer*

$\theta_j$  = *threshold node hidden layer*

$z_j$  = nilai *node hidden layer*

5. Menghitung hasil pada unit keluaran  $y_k$  ( $k=1,2,\dots,n$ )

$$y_{netk} = w_{k0} + \sum_{j=1}^n z_j w_{kj} - \theta_k \quad (2.5)$$

Menghitung aktivasi dengan fungsi sigmoid biner berdasarkan Persamaan 2.1

$$y_k = f(y_{netk}) = \frac{1}{1 + e^{-y_{netk}}} \quad (2.6)$$

Keterangan:

$y_{netk}$  = hasil penjumlahan *hidden layer*

$w_{kj}$  = bobot antara *hidden layer* dan *output layer*

$\theta_k$  = *threshold node output layer*

$y_k$  = nilai *node output layer*

**Langkah untuk propagasi mundur:**

6. Menghitung nilai  $\delta$  pada unit keluaran berdasarkan kesalahan di setiap unit keluaran  $y_k$

$$\begin{aligned} \delta_k &= (t_k - y_k) f'(y_{netk}) \\ \delta_k &= (t_k - y_k) y_k (1 - y_k) \end{aligned} \quad (2.7)$$

$\delta_k$  digunakan untuk mengubah bobot lapisan sebelumnya

7. Menghitung nilai  $\delta$  pada unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi  $z_j$

$$\delta_{netj} = \sum_{k=1}^n \delta_k w_{kj} \quad (2.8)$$

Nilai  $\delta$  untuk unit tersembunyi:

$$\begin{aligned}\delta_j &= (\delta_{netj}) f'(z_{netj}) \\ \delta_j &= (\delta_{netj}) z_j(1 - z_j)\end{aligned} \quad (2.9)$$

Keterangan:

$\delta_k$  = faktor kesalahan unit *output layer*

$t_k$  = target *output layer*

$\delta_{netj}$  = hasil penjumlahan kesalahan unit *output layer*

$\delta_j$  = faktor kesalahan unit *hidden layer*

**Langkah untuk perubahan bobot dan threshold:**

8. Perubahan bobot yang menuju unit keluaran

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (2.10)$$

Perubahan bobot yang menuju unit tersembunyi

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

9. Memperbarui bobot yang menuju unit keluaran

$$w_{kj}(baru) = w_{kj} + \Delta w_{kj} + m \Delta w_{kj}(lama) \quad (2.12)$$

Memperbarui bobot yang menuju unit tersembunyi

$$v_{ji}(baru) = v_{ji} + \Delta v_{ji} + m \Delta v_{ji}(lama) \quad (2.13)$$

Keterangan:

$\Delta w_{kj}$  = koreksi bobot antara *hidden layer* dan *output layer*

$\alpha$  = *learning rate*

$\Delta v_{ji}$  = koreksi bobot antara *input layer* dan *hidden layer*

$w_{kj}(baru)$  = bobot antara *hidden layer* dan *output layer* baru

$m$  = momentum

$\Delta w_{kj}(lama)$  = koreksi bobot antara *hidden layer* dan *output layer* lama

$v_{ji}(baru)$  = bobot antara *input layer* dan *hidden layer* baru

$\Delta v_{ji}(lama)$  = koreksi bobot antara *input layer* dan *hidden layer* lama

10. Perubahan *threshold* pada unit keluaran

$$\Delta \theta_k = \alpha(-1) \delta_k \quad (2.14)$$

11. Perubahan *threshold* pada unit tersembunyi

$$\Delta\theta_j = \alpha(-1)\delta_j \quad (2.15)$$

12. Memperbarui *threshold* pada unit keluaran

$$\theta_k(\text{baru}) = \theta_k + \Delta\theta_k \quad (2.16)$$

13. Memperbarui *threshold* pada unit tersembunyi

$$\theta_j(\text{baru}) = \theta_j + \Delta\theta_j \quad (2.17)$$

Keterangan:

$\Delta\theta_k$  = koreksi *threshold node output layer*

$\Delta\theta_j$  = koreksi *threshold node hidden layer*

$\theta_k(\text{baru})$  = *threshold node output layer* baru

$\theta_j(\text{baru})$  = *threshold node hidden layer* baru

14. Menghitung nilai *error* menggunakan RMSE

$$RMSE = \sqrt{\left( \frac{\sum_n \sum_k (t_{kn} - o_{kn})^2}{nk} \right)} \quad (2.18)$$

Keterangan:

$n$  = jumlah data

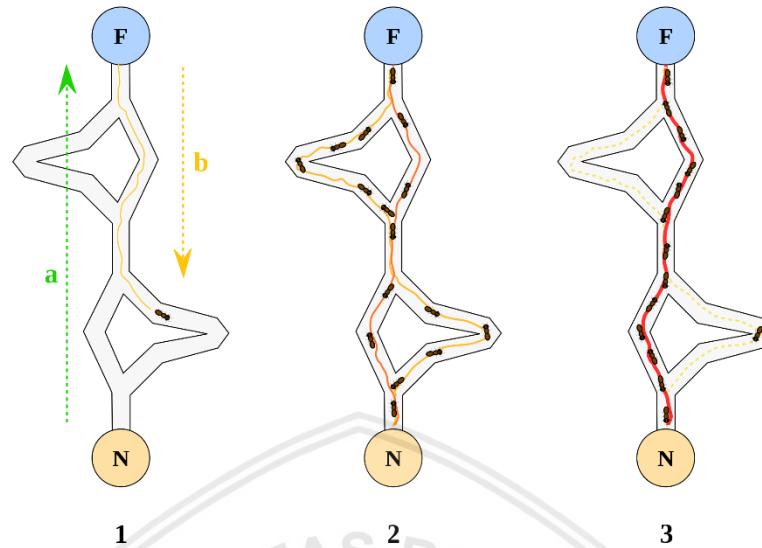
$k$  = jumlah pola *output*

$t_{kn}$  = target pada pola *output* ke-k dan data ke-n

$o_{kn}$  = *output* pada pola *output* ke-k dan data ke-n

15. Menguji kondisi *stop*, yaitu apabila sudah mencapai iterasi maksimal ataupun RMSE telah lebih kecil dari toleransi *error* maksimal

## 2.5 Ant Colony Optimization (ACO)



**Gambar 2.2 Ilustrasi cara kerja ant colony optimization**

Sumber: Dreo (2006)

*Ant colony optimization* (ACO) adalah agen berbasis metaheuristik yang terinspirasi dari perilaku semut dalam mencari makanannya. Semut memiliki perilaku kolektif dalam mempertahankan kehidupan koloninya. Setiap semut bersifat independen dan berkomunikasi dengan semut lain melalui zat kimia yang sangat mudah menguap bernama feromon. Pada perjalanan semut untuk menemukan sumber makanan, zat feromon dikeluarkan dari kelenjar perut dan diletakkan di atas tanah. Semut sangat sensitif pada zat feromon, sehingga zat feromon berfungsi sebagai pemandu jalan saat mereka menuju sumber makanan dan sebaliknya (Haroun, Jamal, & Hicham, 2015). Rute terbaik pilihan koloni semut akan memperoleh intensitas zat feromon tertinggi dan menjadi solusi permasalahan yang dihasilkan ACO. Keunggulan metode ini terletak pada komputasi terdistribusi, umpan balik positif, dan sifat metaheuristik yang dapat memperbaiki kelemahan BPNN (Wang, et al., 2010).

Algoritma ACO melakukan optimasi melalui langkah-langkah sebagai berikut (Dorigo, Birattari, & Stutzle, 2006):

1. Inisialisasi parameter dan lintasan ACO:
  - a. Jumlah semut ( $m$ )
  - b. Node dengan nilai diskrit 0 sampai 10
  - c. Iterasi maksimum ( $NCmax$ )
  - d. Tetapan siklus semut ( $q_0$ ) dengan nilai  $0 < q_0 < 1$
  - e. Tetapan pengendali intensitas feromon ( $\alpha$ ) dengan nilai  $\alpha > 0$
  - f. Tetapan pengendali visibilitas ( $\beta$ ) dengan nilai  $\beta > 0$

- g. Intensitas feromon awal ( $\tau_0$ )
  - h. Tetapan penguapan feromon lokal ( $p$ ) dengan nilai  $0 < p < 1$
  - i. Tetapan penguapan feromon global ( $\delta$ ) dengan nilai  $0 < \delta < 1$
2. Melakukan perjalanan semut:
- a. Meletakkan semut pada *node* awal ( $r$ ) yang bernilai 10
  - b. Menentukan nilai ( $q$ ) secara acak antara 0 sampai 1
  - c. Menentukan *node* selanjutnya ( $u$ ) dengan menghitung peluang maksimal perpindahan semut dari *node* awal ( $r$ ) menggunakan Persamaan 2.17 atau menggunakan seleksi *Roulette Wheel* pada Persamaan 2.18 berikut

Jika  $q \leq q_0$ ,

$$P(r, u) = \left\{ \arg \max_{u \in J(r)} \left\{ [\tau(r, u)]^\alpha \times [\tau(r, u)]^\beta \right\} \right\}_S \quad (2.19)$$

Jika  $q > q_0$ ,

$$P(r, s) = \begin{cases} \frac{[\tau(r, u)]^\alpha \times [\tau(r, u)]^\beta}{\sum_{u \in J(r)} [\tau(r, u)]^\alpha \times [\tau(r, u)]^\beta}, & \text{if } S \in J(r) \\ 0, & \text{lainnya} \end{cases} \quad (2.20)$$

Keterangan:

- $q$  = nilai acak antara 0 sampai 1
- $q_0$  = tetapan siklus semut
- $r$  = *node* awal
- $u$  = *node* selanjutnya
- $J(r)$  = kumpulan *node* yang masih tersedia
- $\tau$  = intensitas feromon
- $\alpha$  = tetapan pengendali intensitas feromon
- $\beta$  = tetapan pengendali visibilitas
- $S$  = nilai acak antara 0 sampai 9

- d. Update feromon lokal  $\tau(r, u)$  dengan persamaan berikut

$$\tau(r, u) = (1 - p) \times (\tau(r, u)) + \rho \tau_0 \quad (2.21)$$

Keterangan:

- $\rho$  = tetapan penguapan feromon lokal
- $\tau_0$  = intensitas feromon awal

- e. Membarui *node* awal ( $r$ ) dengan *node* yang didapatkan ( $u$ ). Ulangi dari langkah 2b sampai seluruh *node* dilewati semut. Setelah semua *node* terlewati, maka semut berikutnya memulai perjalanan dari langkah 2a

3. *Update feromon global pada seluruh lintasan*

$$\tau(r, u) = (1 - \delta) \times (\tau(r, u)) + \delta \Delta \tau(r, u) \quad (2.22)$$

$\Delta \tau(r, u)$  digunakan untuk meningkatkan intensitas feromon pada rute terbaik. Oleh karena itu, selain *global best* nilai  $\Delta \tau(r, u)$  adalah 0.

$$\Delta \tau(r, u) = \begin{cases} \frac{1}{L}, & \text{if } (r, u) \in \text{global best route} \\ 0, & \text{lainnya} \end{cases} \quad (2.23)$$

Keterangan:

$\Delta \tau(r, u)$  = matriks perubahan intensitas feromon global

$L$  = RMSE *global best route*

4. Kosongkan rute perjalanan setiap semut

5. Apabila belum mencapai iterasi maksimal, maka mengulang dari langkah 2

## 2.6 Min – Max Normalization

Normalisasi dilakukan untuk mengubah interval data sesuai dengan interval *output* yang dimiliki oleh BPNN di penelitian ini. Normalisasi yang dilakukan menggunakan *min-max normalization* pada range 0 sampai 1.

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (2.24)$$

Ketika proses perhitungan ACO-BPNN selesai, maka diperlukan proses denormalisasi untuk mengembalikan nilai data menjadi nilai aktualnya. Rumus denormalisasi ditunjukkan pada bagian di bawah ini.

$$x = x'(x_{\max} - x_{\min}) + x_{\min} \quad (2.25)$$

Keterangan:

$x'$  = nilai data hasil normalisasi

$x$  = nilai data aktual

$x_{\max}$  = nilai maksimal *dataset*

$x_{\min}$  = nilai minimal *dataset*

## 2.7 Mean Absolute Percentage Error

MAPE merupakan persentase rata-rata kesalahan absolut antara hasil prediksi dan nilai aktual. Sehingga hasil prediksi dapat dikatakan semakin baik ketika nilai MAPE semakin kecil. Interpretasi dari nilai MAPE ditunjukkan pada tabel 2.2.

**Tabel 2.2 Tabel interpretasi nilai MAPE**

MAPE	Arti
< 10%	Peramalan sangat bagus
10% - 20%	Peramalan bagus
20% - 50%	Peramalan cukup bagus
> 50%	Peramalan tidak bagus

**Sumber:** Chang, et al. (2007) disitasi dalam Rof'i'ah, Cholissodin, & Dewi (2018)

Rumus perhitungan untuk memperoleh nilai MAPE ditunjukkan pada Persamaan 2.26.

$$MAPE = \frac{\sum_{t=1}^n \left| \frac{x_t - f_t}{x_t} \right|}{n} \times 100\% \quad (2.26)$$

Keterangan:

$n$  = jumlah data

$x_t$  = nilai aktual pada indeks ke-t

$f_t$  = nilai prediksi pada indeks ke-t

## 2.8 Ant Colony Optimization – Backpropagation Neural Network

Langkah-langkah ACO untuk mengoptimalkan parameter *learning rate*, momentum, dan jumlah *hidden node* dari BPNN adalah sebagai berikut (Lixing, et al., 2010):

1. Inisialisasi parameter dan lintasan ACO
2. Melakukan perjalanan semut
  - a. Meletakkan semut pada *node* awal ( $r$ )
  - b. Menentukan nilai ( $q$ )
  - c. Menentukan *node* selanjutnya ( $u$ ) menggunakan Persamaan 2.19 atau 2.20 berdasarkan kondisi nilai ( $q$ ) terhadap nilai ( $q_0$ )
  - d. Membarui feromon lokal  $\tau(r, u)$  dengan Persamaan 2.21
  - e. Membarui *node* awal ( $r$ ) dengan *node* yang didapatkan ( $u$ ). Ulangi dari langkah 2b sampai seluruh *node* dilewati semut. Setelah semua *node* terlewati, maka semut berikutnya memulai perjalanan dari langkah 2a
3. Konversi nilai parameter *learning rate*, momentum, dan jumlah *hidden node* BPNN dari kumpulan *node* kunjungan setiap semut

$$A = \frac{(J_i \times 100) + (J_{i+1} \times 10) + (J_{i+2} \times 1)}{1000} \quad (2.27)$$

$$Norma = \left[ \left( \frac{A - A_{min}}{A_{max} - A_{min}} \right) \times (\alpha_{max} - \alpha_{min}) \right] + \alpha_{min} \quad (2.28)$$

$$Normm = \left[ \left( \frac{A - A_{min}}{A_{max} - A_{min}} \right) \times (mmax - mmin) \right] + mmin \quad (2.29)$$

$$A = (J_i \times 10) + (J_{i+1} \times 1) \quad (2.30)$$

$$cap = \sum_{i=j_{min}+1}^{j_{max}} 5i \quad (2.31)$$

$$Normj = \begin{cases} \frac{cap}{5}, & \text{if } A \leq cap \\ \text{lainnya} & \end{cases} \quad (2.32)$$

Keterangan:

$A$  = hasil optimasi ACO

$J_i$  = node kunjungan semut pada indeks ke- $i$

$Norma$  = hasil konversi nilai parameter *learning rate*

$A_{max}$  = nilai maksimal optimasi ACO

$A_{min}$  = nilai minimal optimasi ACO

$\alpha_{max}$  = batas atas parameter *learning rate*

$\alpha_{min}$  = batas bawah parameter *learning rate*

$Normm$  = hasil konversi nilai parameter momentum

$mmax$  = batas atas parameter momentum

$mmin$  = batas bawah parameter momentum

$cap$  = batas penentuan nilai parameter jumlah *hidden node*

$jmax$  = batas atas parameter jumlah *hidden node*

$jmin$  = batas bawah parameter jumlah *hidden node*

$Normj$  = hasil konversi nilai parameter jumlah *hidden node*

4. Melakukan normalisasi data latih menggunakan Persamaan 2.24. Selanjutnya melatih jaringan BPNN sebanyak hasil konversi parameter semua semut. Lalu menjadikan RMSE terkecil dari seluruh pelatihan sebagai *global best*
5. Membarui feromon global pada seluruh lintasan memakai Persamaan 2.22. Nilai  $\Delta\tau(r, u)$  rute *global best* dihitung memakai Persamaan 2.23, sehingga nilai  $\Delta\tau(r, u)$  rute selain *global best* adalah 0
6. Mengosongkan rute perjalanan setiap semut
7. Memeriksa kondisi *stop*, apakah sudah mencapai iterasi maksimal. Jika belum mencapai iterasi maksimal, maka tahapan diulang dari langkah 2. Sebaliknya, tahapan dilanjutkan menuju pengujian BPNN

- repository.ub.ac.id
8. Melakukan normalisasi data uji menggunakan Persamaan 2.24. Kemudian melakukan proses *feedforward* memakai jumlah *hidden node*, bobot, dan *threshold* milik *global best* pada iterasi ACO terakhir. Keluaran proses tersebut didenormalisasi memakai Persamaan 2.25 agar dapat dihitung MAPE-nya menggunakan Persamaan 2.26



## BAB 3 METODOLOGI PENELITIAN

Dibutuhkan tahapan kegiatan penelitian yang terstruktur dan tepat agar mampu mencapai tujuan penelitian. Bagian metodologi penelitian membahas beberapa hal, seperti metode, teknik, dan langkah-langkah yang digunakan dalam penelitian non-implementatif analitik berjudul “Prediksi Harga Saham menggunakan Metode *Backpropagation* dengan Optimasi *Ant Colony Optimization*”. Tahapan kegiatan penelitian meliputi studi literatur, pengumpulan data, penggunaan perangkat, perancangan, implementasi, pengujian dan analisis, serta penarikan kesimpulan. Diagram dari tahapan tersebut dapat dilihat pada Gambar 3.1 di bawah ini:



Gambar 3.1 Diagram alir metodologi penelitian

### 3.1 Studi Literatur

Tahapan pertama yang dilakukan tentunya mengumpulkan dan mempelajari literatur untuk memahami konsep-konsep dalam bidang ilmu yang berkaitan dengan penelitian ini. Literatur tersebut adalah *e-book*, jurnal, serta laporan penelitian sebelumnya, baik skala nasional maupun internasional. Referensi yang dibutuhkan untuk menunjang penelitian ini antara lain mengenai:

1. Saham
2. *Ant Colony Optimization*
3. *Backpropagation Neural Network*
4. *Ant Colony Optimization – Backpropagation Neural Network*

### 3.2 Pengumpulan Data

Penelitian ini menggunakan data *time series* harga saham Bank Rakyat Indonesia (Persero) Tbk. periode 1 Januari 2018 sampai 31 Desember 2018. Data diunduh langsung dari situs *Yahoo Finance* dengan format *xlsx*. Data tersebut merupakan *close price* harian di hari kerja sejumlah 261 data. Setiap data memiliki tambahan informasi berupa tanggal dari *close price* yang bersangkutan dengan format *mm/dd/yyyy*. Data mentah tersebut dapat dilihat pada LAMPIRAN A.1, sedangkan sampel datanya ditampilkan pada Tabel 3.1.

**Tabel 3.1 Sampel data mentah**

Tanggal	Harga	Tanggal	Harga
12/03/2018	2970	12/12/2018	3040
12/04/2018	2970	12/13/2018	3010
12/05/2018	2970	12/14/2018	2990
12/06/2018	2870	12/17/2018	3040
12/07/2018	2850	12/18/2018	3090
12/10/2018	2890	12/19/2018	3080
12/11/2018	2980	12/20/2018	3070

Selanjutnya menyusun setiap 4 data mentah berurutan menjadi 1 data dengan 4 fitur, yaitu X1, X2, X3, dan X4. Fitur X1-X4 pertama merupakan data mentah indeks 1-4, lalu X1-X4 kedua merupakan data mentah indeks 2-5, dan seterusnya. Data untuk pelatihan dan pengujian memiliki jumlah fitur yang sama dengan 3 fitur pertama sebagai *input* dan fitur terakhir sebagai target. Sehingga dalam melakukan prediksi pada hari tertentu memperhitungkan *close price* pada 3 hari sebelumnya. Sampel data yang telah disesuaikan dengan kebutuhan penelitian ditampilkan pada Tabel 3.2.

**Tabel 3.2 Sampel data**

X1	X2	X3	X4
2970	2970	2970	2870
2970	2970	2870	2850
2970	2870	2850	2890
2870	2850	2890	2980
2850	2890	2980	3040
2890	2980	3040	3010
2980	3040	3010	2990

3040	3010	2990	3040
3010	2990	3040	3090
2990	3040	3090	3080
3040	3090	3080	3070

### 3.3 Penggunaan Perangkat

Melakukan penguraian atas perangkat yang digunakan dalam implementasi prediksi harga saham. Perangkat yang dimaksud meliputi perangkat keras dan perangkat lunak.

#### 1. Penggunaan Perangkat Keras

- Intel Core™ i7-4510U CPU @ 2.00 GHz 2.60 GHz
- Memori 4GB RAM
- Intel® HD Graphics
- Harddisk 1 TB

#### 2. Penggunaan Perangkat Lunak

- Sistem Operasi Windows 8.1 64-bit
- Sublime Text 3.1.1
- XAMPP 7.2.12
- Mozilla Firefox 63.0.3 64-bit

### 3.4 Perancangan

Bagian ini menjelaskan rancangan tahapan kerja dalam melakukan prediksi harga saham menggunakan metode BPNN yang dioptimasi ACO secara detail. Pada bagian pertama menjelaskan perancangan algoritma menggunakan diagram alir yang mewakili proses-proses dalam sistem. Lalu menjabarkan perhitungan manual pada metode ACO dan BPNN. Kemudian menjelaskan perancangan antarmuka melalui cetak biru antarmuka beserta menyebutkan komponen di dalamnya. Bagian akhir adalah perancangan pengujian yang menyebutkan apa saja pengujian yang dilakukan beserta tujuan dan desain tabelnya. Tahapan perancangan bertujuan untuk memudahkan proses implementasi dan pengujian dalam penelitian ini.

### 3.5 Implementasi

Implementasi adalah tahapan untuk membangun sistem berdasarkan perancangan yang telah dilakukan. Beberapa hal yang dilakukan dalam tahapan ini antara lain:

1. Implementasi algoritma ACO-BPNN ke dalam bahasa pemrograman
2. Implementasi antarmuka berbasis *web*

### 3.6 Pengujian dan Analisis

Pada tahap ini dilakukan pengujian menggunakan MAPE terhadap selisih keluaran program dengan nilai aktual. Nilai MAPE dihasilkan dari setiap pengujian pada parameter ACO yaitu:

1. Tetapan siklus semut ( $q_0$ )
2. Tetapan pengendali intensitas feromon ( $\alpha$ )
3. Tetapan pengendali visibilitas ( $\beta$ )
4. Tetapan penguapan feromon lokal ( $\rho$ )
5. Tetapan penguapan feromon global ( $\delta$ )
6. Jumlah semut ( $m$ )
7. Jumlah iterasi ( $NCmax$ )

Masing-masing pengujian dilakukan dengan cara memberikan nilai yang berbeda pada suatu parameter. Nilai parameter dengan MAPE terkecil di setiap pengujian disimpan dan dianalisis penyebabnya hingga mendapatkan kombinasi parameter ACO terbaik. Berikutnya melakukan perbandingan MAPE antara ACO-BPNN dengan kombinasi parameter terbaik dan kombinasi parameter acak untuk BPNN beserta waktu komputasinya.

### 3.7 Penarikan Kesimpulan

Tahapan akhir ini dilakukan guna menjawab rumusan masalah penelitian. Kesimpulan dibuat berdasarkan hasil analisis pada tahap pengujian. Selain itu, terdapat evaluasi dengan memberikan saran pada penelitian ini. Hal tersebut diperlukan untuk perbaikan kekurangan penelitian ini dan sebagai bahan pertimbangan pada perkembangan penelitian selanjutnya.

## BAB 4 PERANCANGAN

Pada bab ini akan dijelaskan mengenai perancangan dari penelitian “Prediksi Harga Saham menggunakan Metode *Backpropagation* dengan Optimasi *Ant Colony Optimization*”. Tahap perancangan dalam penelitian ini terdiri dari tiga tahapan, yaitu perancangan algoritma, perancangan antarmuka, dan perancangan pengujian.

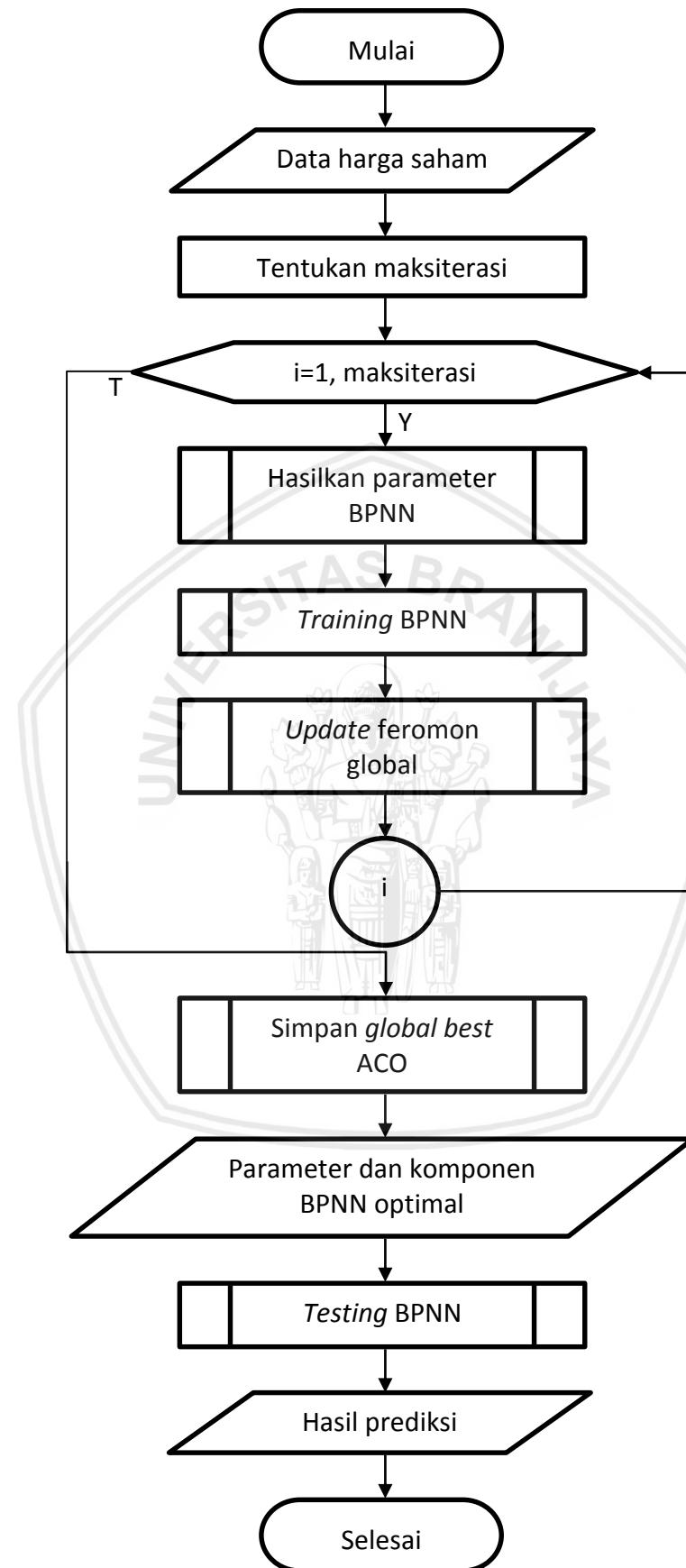
### 4.1 Definisi Masalah

Permasalahan yang akan diselesaikan adalah melakukan “Prediksi Harga Saham menggunakan Metode *Backpropagation* dengan Optimasi *Ant Colony Optimization*”. Metode BPNN digunakan untuk melakukan prediksi harga saham pada periode berikutnya, sedangkan metode ACO digunakan untuk mengoptimasi parameter *learning rate*, momentum, dan jumlah *hidden node* dari BPNN.

Proses prediksi ini menggunakan harga saham milik Bank Rakyat Indonesia (Persero) Tbk. pada periode 1 Januari 2018 sampai 31 Desember 2018 sebagai masukan bagi sistem. Pada penelitian ini metode BPNN dibagi dalam dua tahap, yaitu fase *training* dan *testing*. Sebelum melakukan fase *training* BPNN, akan dilakukan optimasi dengan metode ACO seperti yang telah dijelaskan sebelumnya.

### 4.2 Perancangan Algoritma

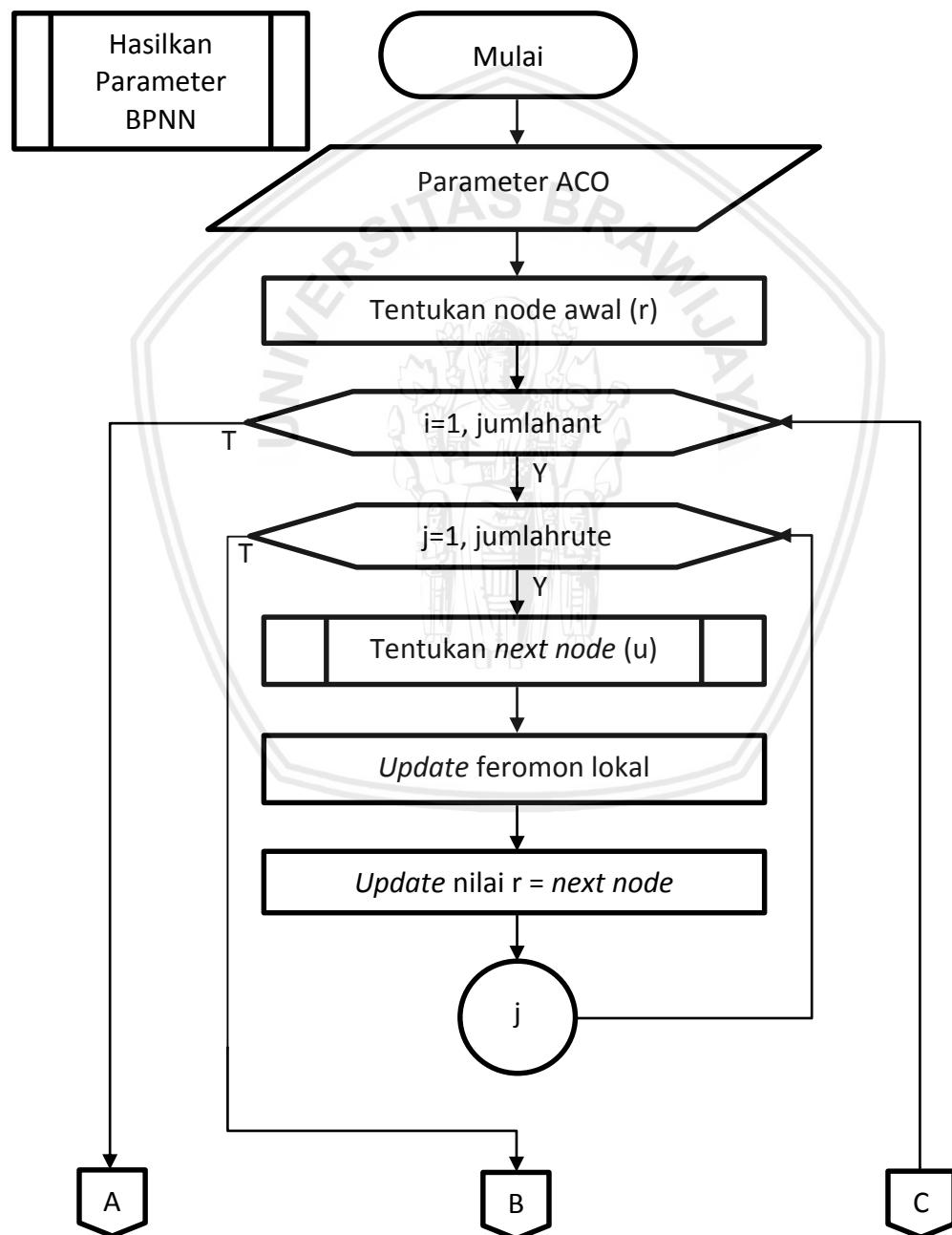
Diagram alir merupakan gambaran umum dari tahapan kerja pada sistem yang akan dikembangkan. Dalam diagram alir sistem terdapat tahapan-tahapan yang dimulai dari *input* data harga saham milik Bank Rakyat Indonesia (Persero) Tbk. sebagai masukan bagi sistem. Lalu menentukan iterasi ACO maksimum. Selanjutnya menghasilkan kombinasi nilai parameter BPNN, fase *training* BPNN yang menghasilkan RMSE, dan *update* feromon global hingga iterasi maksimal tercapai. Kemudian menyimpan *global best* ACO terakhir, yaitu *learning rate*, momentum, jumlah *hidden node*, bobot, dan *threshold* BPNN. Jumlah *hidden node*, bobot, dan *threshold* merupakan komponen BPNN yang akan dimuat pada tahap *testing* BPNN. Melalui tahap pengujian tersebut dihasilkanlah prediksi harga saham pada suatu periode. Diagram alir metode ACO-BPNN untuk melakukan prediksi harga saham dapat dilihat pada Gambar 4.1.

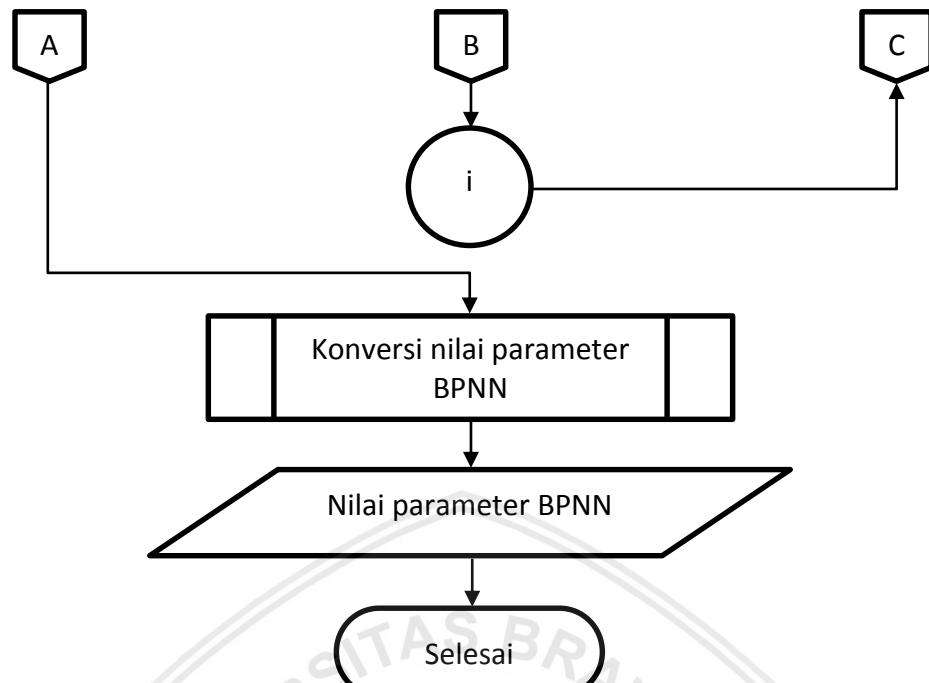


Gambar 4.1 Diagram alir algoritma ACO-BPNN

#### 4.2.1 Hasilkan Parameter Backpropagation Neural Network

Dalam diagram alir ini terdapat tahapan-tahapan yang dimulai dari menerima masukan berupa parameter ACO untuk melakukan optimasi. Lalu menentukan *node* awal untuk setiap semut. *Node* awal adalah *node* pasif yang tidak dihitung pada proses konversi parameter BPNN. Sebanyak rute yang harus ditempuh, tentukan *node* selanjutnya, *update feromon lokal*, dan *update current node*, hal ini dilakukan sejumlah semut yang telah diinisialisasi. Selanjutnya melakukan konversi nilai parameter dari *node* kunjungan semut ACO menjadi *learning rate*, momentum, dan jumlah *hidden node* BPNN. Diagram alir optimasi ACO untuk menghasilkan parameter BPNN dapat dilihat pada Gambar 4.2.

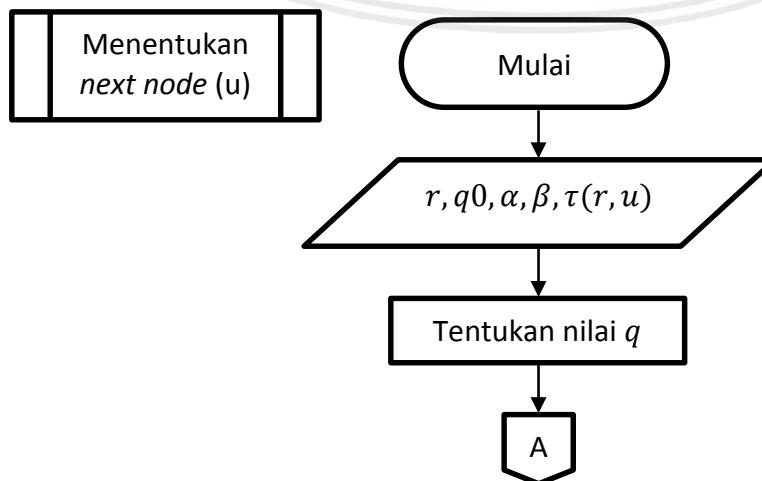


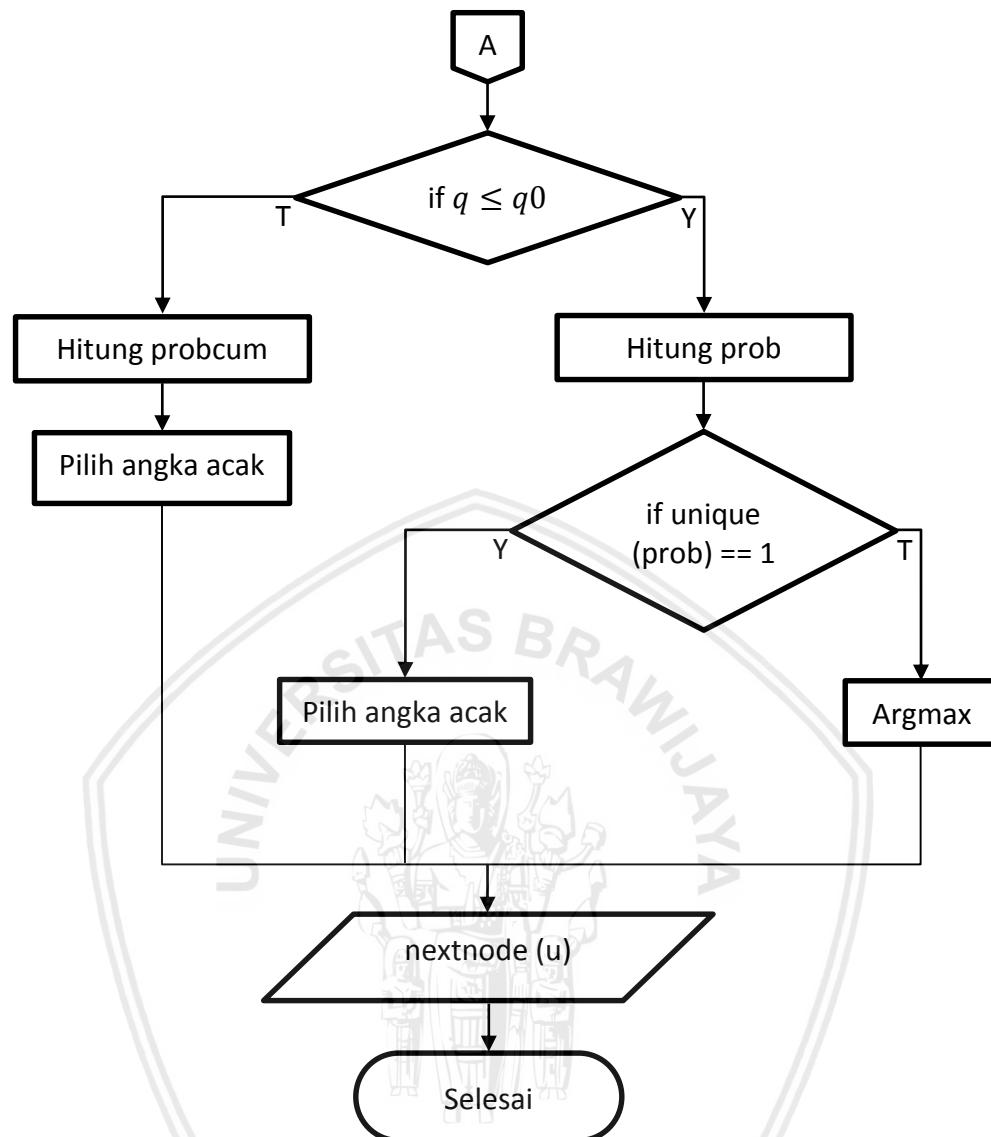


Gambar 4.2 Diagram alir optimasi ACO

#### 4.2.1.1 Menentukan *Next Node*

Terdapat tahapan-tahapan dalam menentukan *node* selanjutnya. Pertama-tama menerima masukan *current node*, tetapan siklus semut, tetapan pengendali intensitas feromon, tetapan pengendali visibilitas, dan intensitas feromon pada semua rute. Setelah itu menentukan nilai  $q$  secara acak. Selanjutnya memasuki tahap seleksi kondisi, apabila nilai  $q$  kurang dari atau sama dengan nilai  $q_0$ , maka menghitung probabilitas. Ketika semua nilai probabilitas sama, maka menentukan node selanjutnya secara acak, sebaliknya, menentukan node selanjutnya dengan Argmax. Apabila nilai  $q$  lebih dari nilai  $q_0$ , maka menghitung probabilitas kumulatif dan menentukan node selanjutnya secara acak. Diagram alir menentukan *next node* dapat dilihat pada Gambar 4.3.

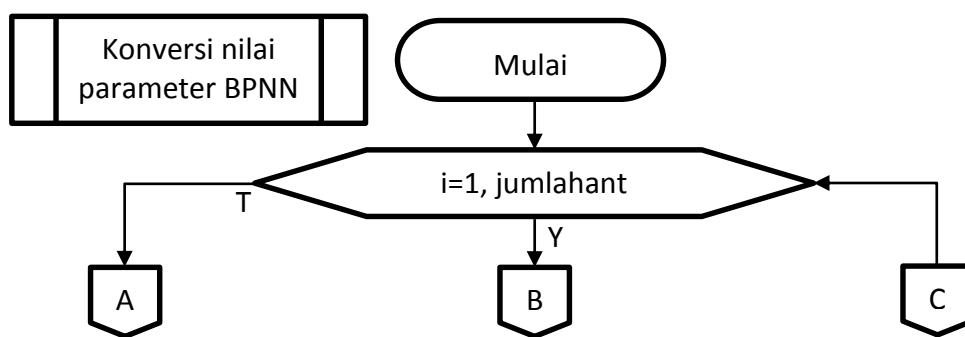


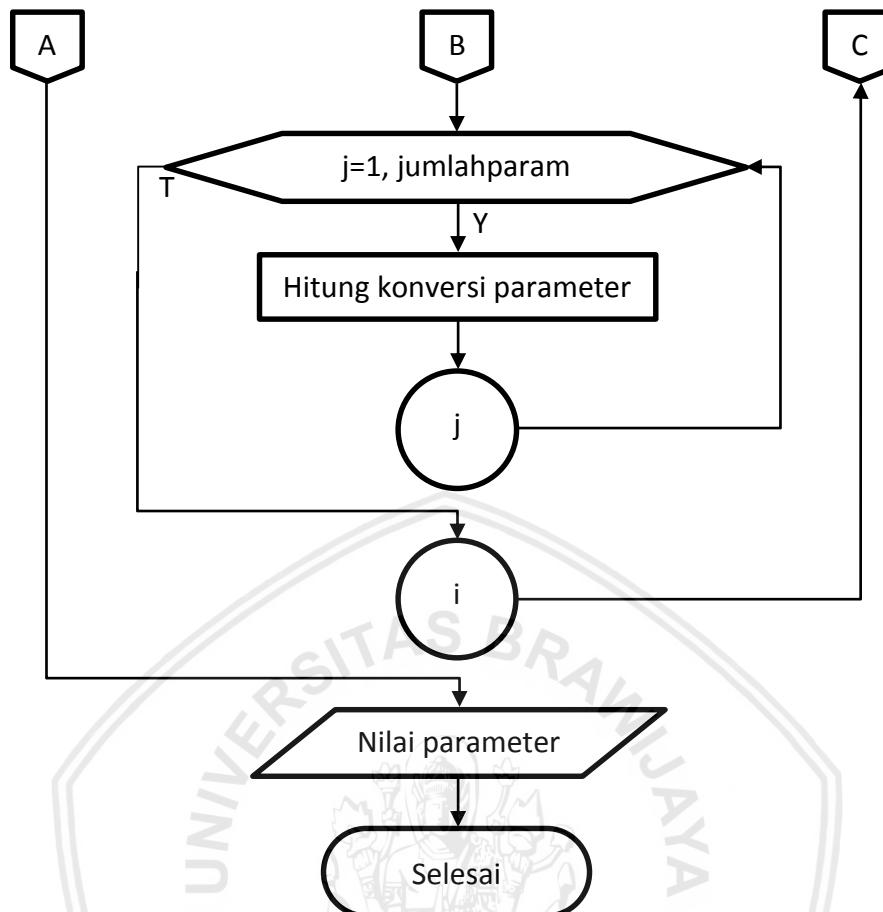


Gambar 4.3 Diagram alir menentukan node selanjutnya

#### 4.2.1.2 Konversi Nilai Parameter BPNN

Tahapan untuk menghasilkan parameter BPNN cukup sederhana. Lakukan perhitungan konversi parameter sebanyak jumlah parameter yang dioptimasi oleh ACO, langkah ini dieksekusi sejumlah semut yang telah diinisialisasi. Diagram alir konversi parameter BPNN dapat dilihat pada Gambar 4.4.

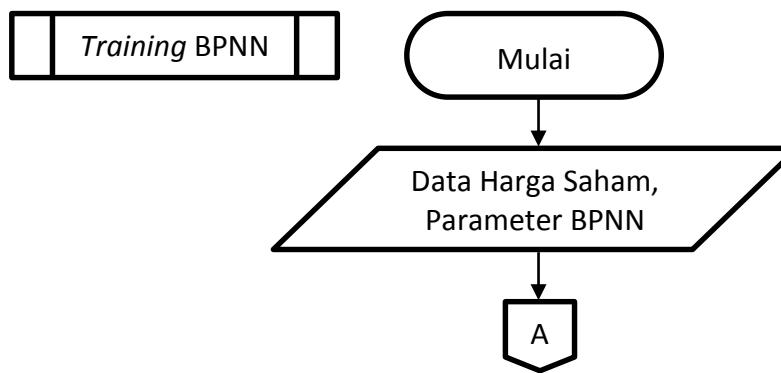


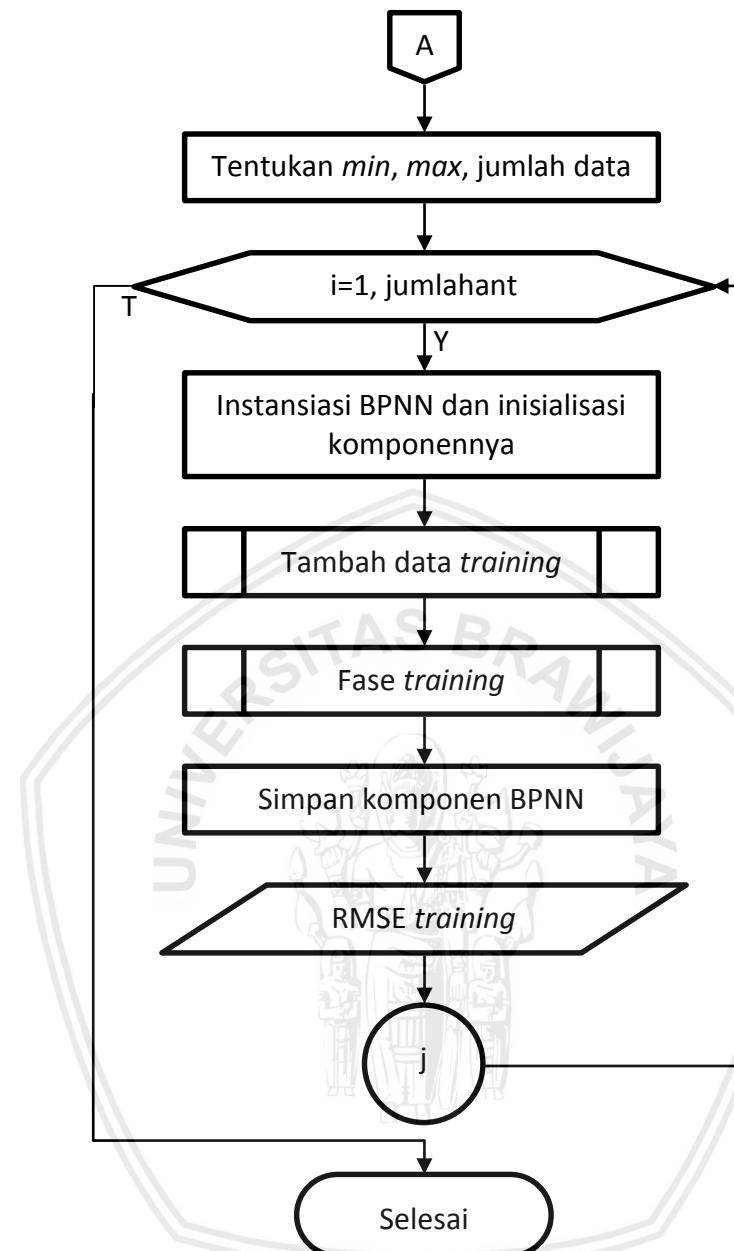


Gambar 4.4 Diagram alir konversi nilai parameter BPNN

#### 4.2.2 Training Backpropagation Neural Network

Diagram alir *training* BPNN memiliki tahapan-tahapan yang dimulai dari menerima masukan berupa data harga saham dan parameter BPNN. Berikutnya menentukan nilai *min*, nilai *max*, dan jumlah data pelatihan. Kemudian melakukan perulangan sejumlah semut yang telah diinisialisasi. Perulangan tersebut berisi proses instansiasi dan inisialisasi BPNN, menambah data *training*, fase *training*, menyimpan komponen BPNN, dan menghasilkan RMSE. Nilai RMSE setiap semut digunakan untuk seleksi penentuan *global best* dalam satu iterasi. Diagram alir *training* BPNN dapat dilihat pada Gambar 4.5.

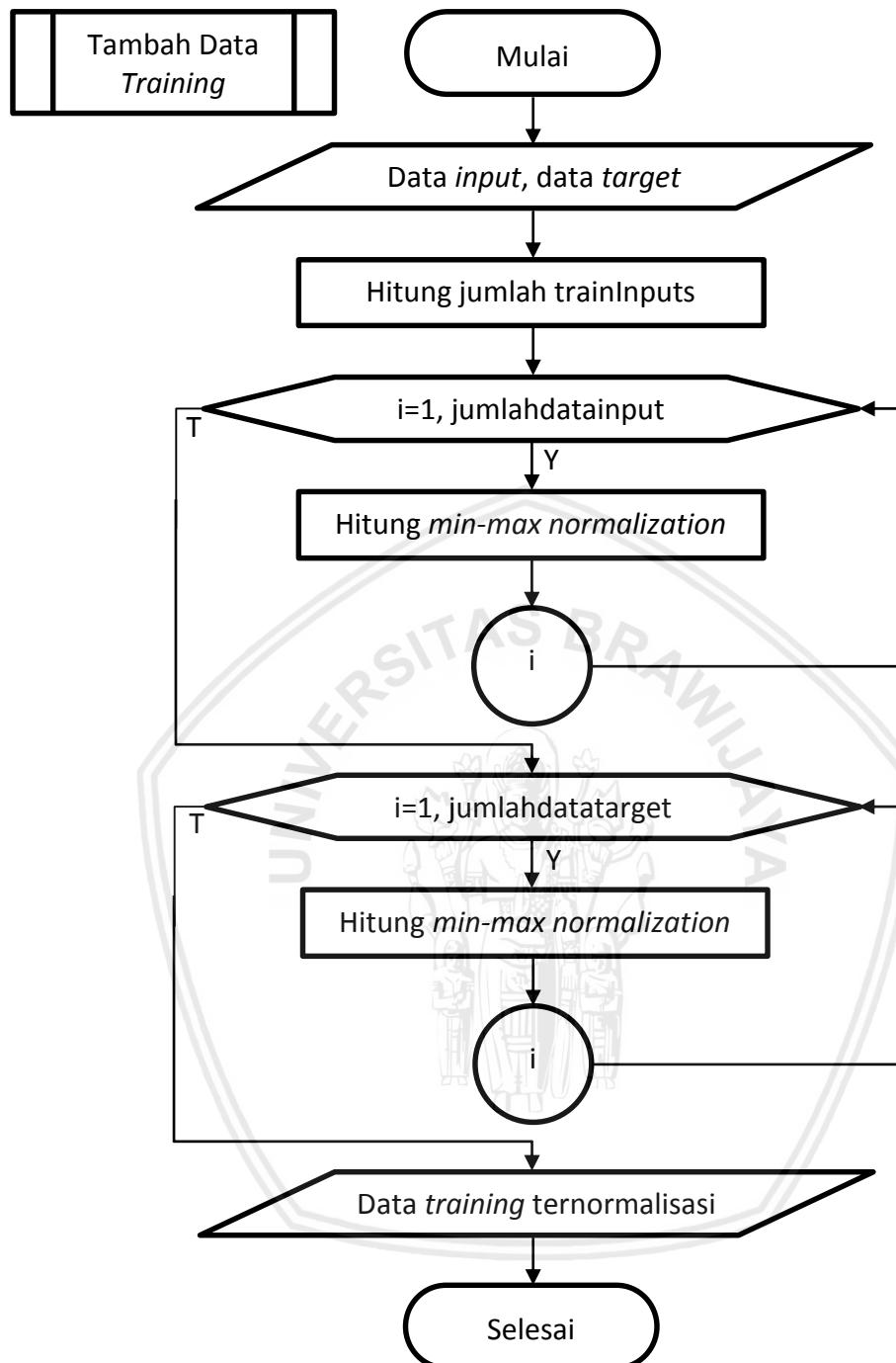




Gambar 4.5 Diagram alir *training* BPNN

#### 4.2.2.1 Tambah Data *Training*

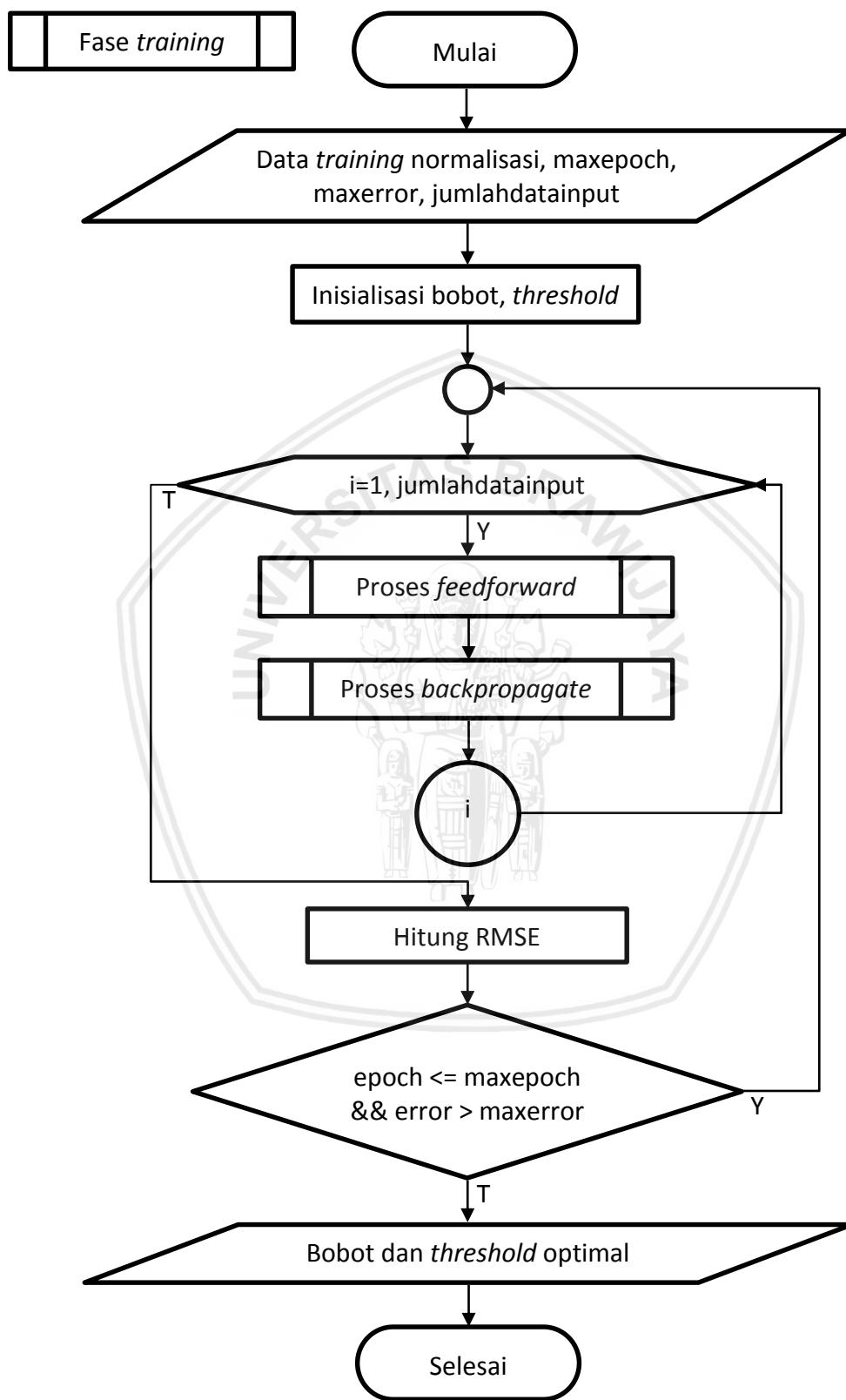
Terdapat tahapan-tahapan dalam menambahkan data *training*. Pada tahap awal menerima masukan data *input* dan data *target*. Setelah itu menghitung jumlah trainInputs untuk menentukan *index* data saat ini. Kemudian melakukan perulangan sejumlah nilai variabel jumlahdatainput untuk menghitung *min-max normalization* data *input*. Lalu melakukan perulangan sejumlah nilai variabel jumlahdatatarget untuk menghitung *min-max normalization* data *target*. Hasil yang diberikan berupa data *training* ternormalisasi. Diagram alir menambahkan data *training* BPNN dapat dilihat pada Gambar 4.6.

Gambar 4.6 Diagram alir tambah data *training*

#### 4.2.2.2 Fase *Training*

Terdapat serangkaian tahapan dalam fase *training* BPNN. Pertama-tama menerima masukan data *training* normalisasi dan nilai variabel maxepoch, maxerror, serta jumlahdatainput. Selanjutnya melakukan inisialisasi bobot dan *threshold* dengan nilai acak. Lakukan proses *feedforward* dan *backpropagate* sebanyak nilai variabel jumlahdatainput, lalu hitung nilai RMSE. Proses dari *feedforward-backpropagate* hingga menghitung RMSE dilakukan selama nilai epoch kurang dari atau sama dengan nilai maxepoch dan nilai error lebih dari nilai

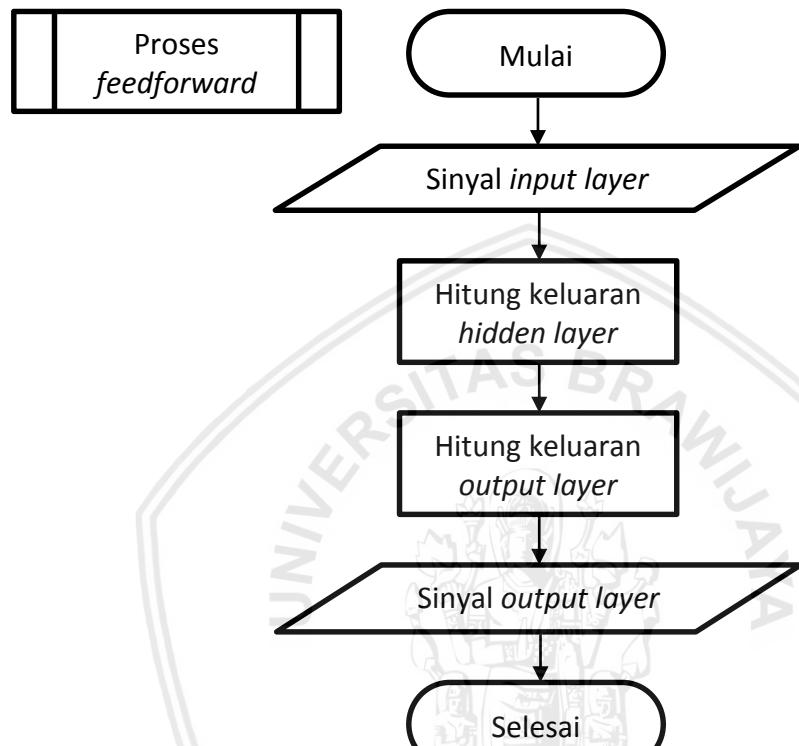
maxerror. Fase *training* ini menghasilkan bobot dan *threshold* optimal untuk fase *testing*. Diagram alir fase *training* BPNN dapat dilihat pada Gambar 4.7.



Gambar 4.7 Diagram alir fase *training*

## 1. Proses *Feedforward*

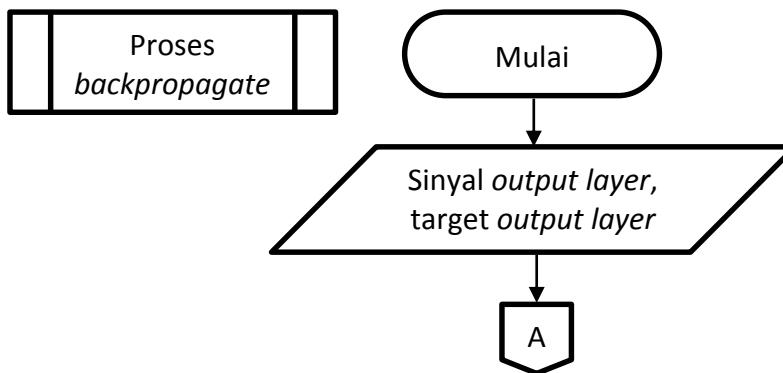
Terdapat beberapa tahapan dalam proses *feedforward* yang dimulai dari menerima masukan sinyal *input layer*. Kemudian melakukan perhitungan keluaran *hidden layer* dan *output layer*. Keluaran pada *output layer* adalah sinyal *output layer* atau *output* yang dihasilkan BPNN. Diagram alir proses *feedforward* dapat dilihat pada Gambar 4.8.

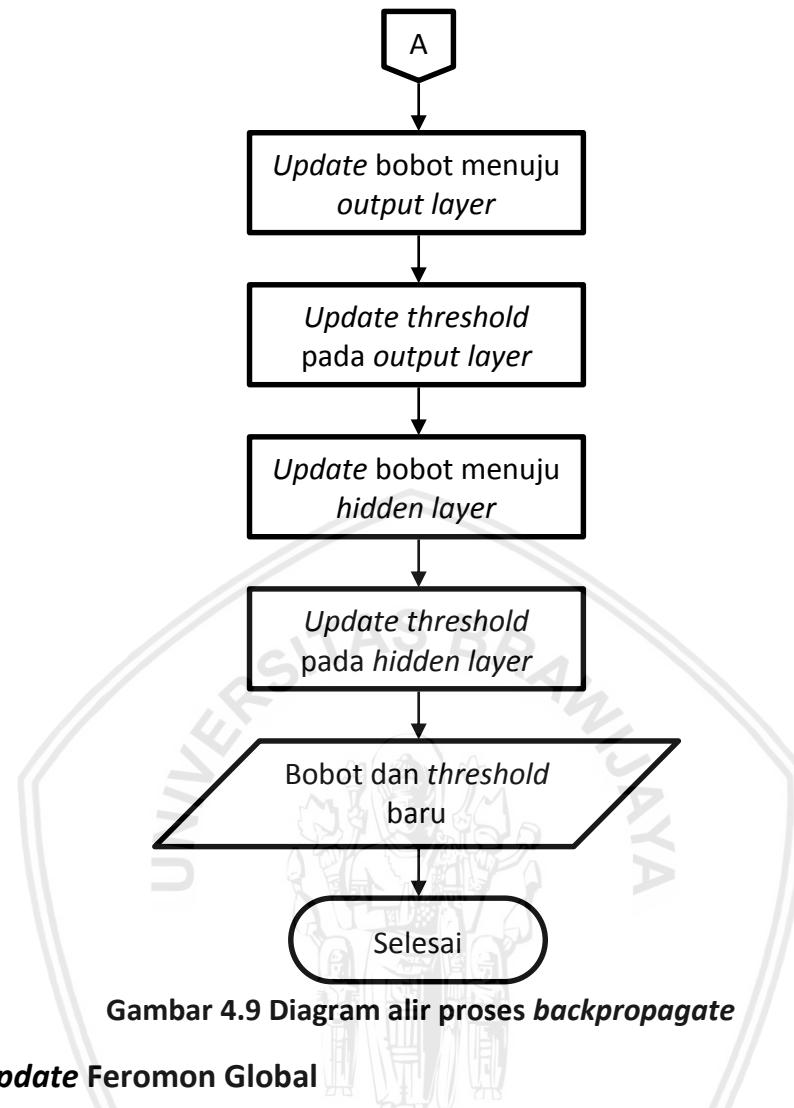


Gambar 4.8 Diagram alir proses *feedforward*

## 2. Proses *Backpropagate*

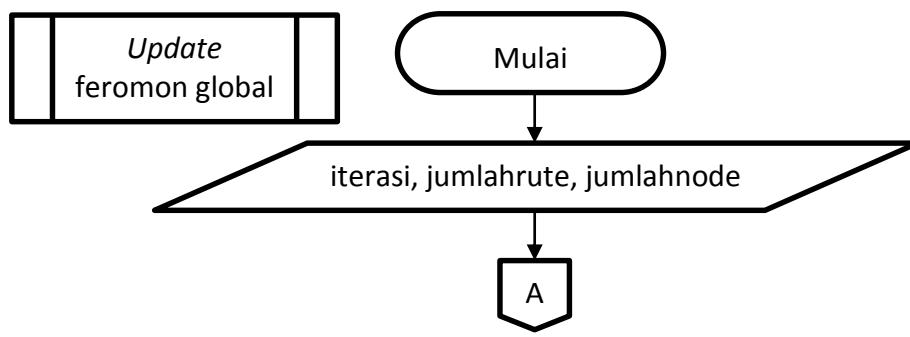
Terdapat beberapa tahapan dalam proses *backpropagate* yang dimulai dari menerima masukan sinyal dan target dari *output layer*. Kemudian melakukan *update* bobot menuju *output layer* dan *threshold* pada *output layer*. Lalu melakukan *update* bobot menuju *hidden layer* dan *threshold* pada *hidden layer*. Hasil dari proses ini adalah bobot dan *threshold* baru. Diagram alir proses *backpropagate* dapat dilihat pada Gambar 4.9.

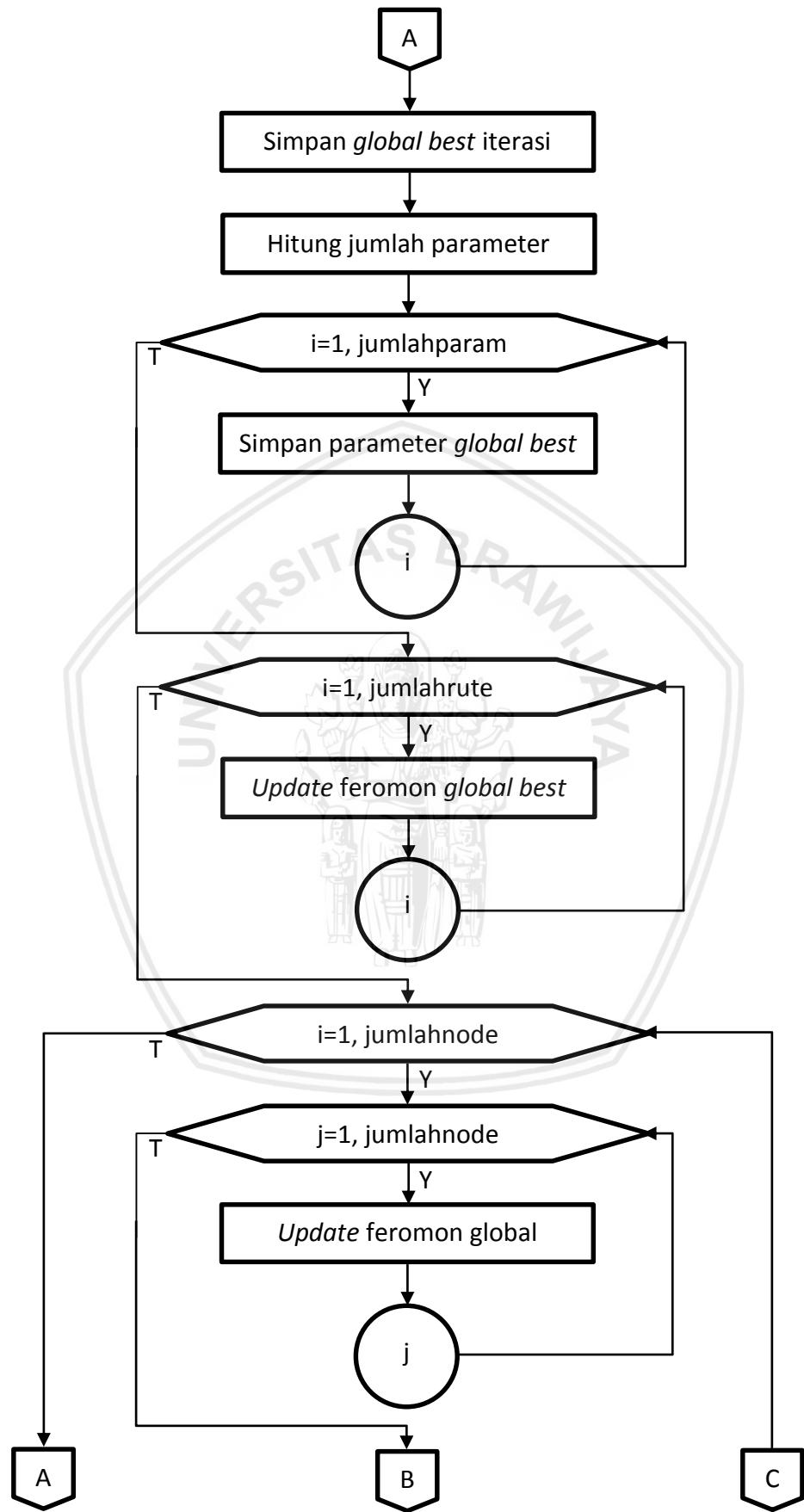


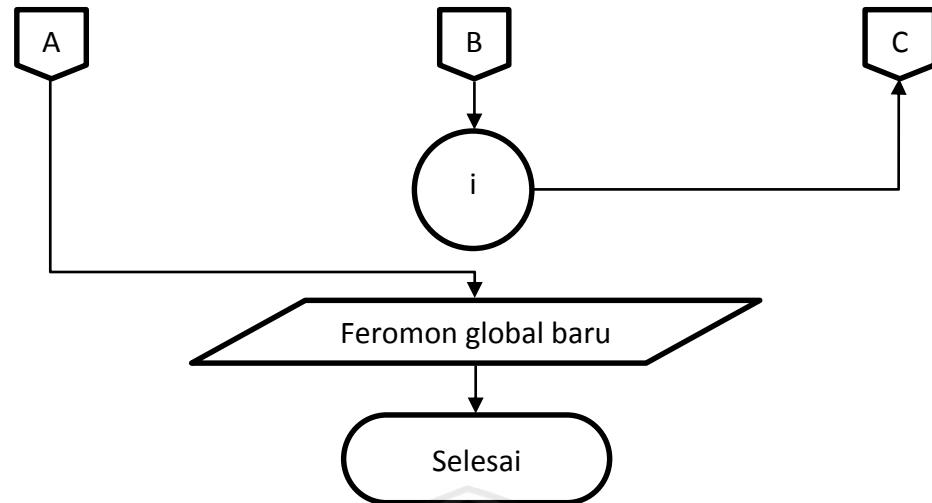


#### 4.2.3 Update Feromon Global

Diagram alir *update feromon global* dimulai dari menerima masukan nilai iterasi, jumlahrute, dan jumlahnode. Kemudian menyimpan RMSE dan komponen BPNN *global best* iterasi dan menghitung jumlah parameter. Lalu melakukan perulangan untuk menyimpan parameter *global best* sebanyak nilai jumlahparam. Melakukan perulangan lagi untuk *update* feromon pada rute *global best* sebanyak nilai jumlahrute. Selanjutnya melakukan *nested loop* masing-masing sebanyak nilai jumlahnode untuk *update* feromon pada semua rute. Melalui rangkaian tahapan tersebut maka dihasilkanlah intensitas feromon baru bagi seluruh rute. Diagram alir *update feromon global* dapat dilihat pada Gambar 4.10.

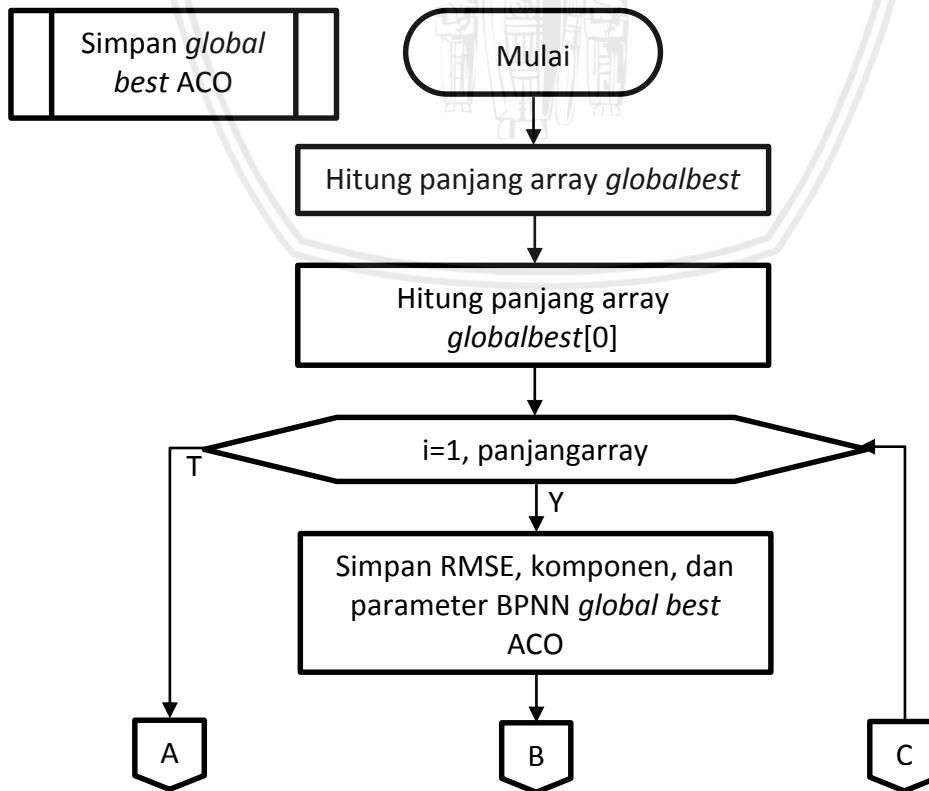


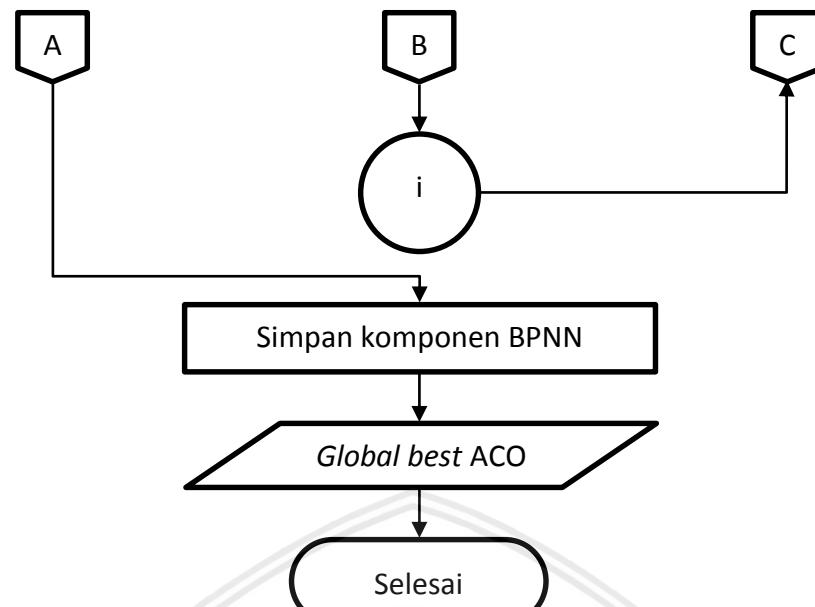


Gambar 4.10 Diagram alir *update feromon global*

#### 4.2.4 Simpan *Global Best Ant Colony Optimization*

Diagram alir menyimpan *global best* ACO dimulai dari menghitung panjang array *globalbest* untuk mengetahui *index* terakhirnya. Kemudian menghitung panjang array *globalbest[0]* sebagai batas perulangan pada proses berikutnya. Lalu melakukan perulangan sejumlah nilai panjangarray untuk menyimpan RMSE, komponen, dan parameter BPNN *global best* ACO. Setelah itu komponen BPNN juga disimpan dalam sebuah file berformat .txt. Rangkaian tahapan tersebut akan menghasilkan *global best* ACO, dimana komponen BPNN-nya akan digunakan untuk fase *testing* BPNN. Diagram alir menyimpan *global best* ACO dapat dilihat pada Gambar 4.11.

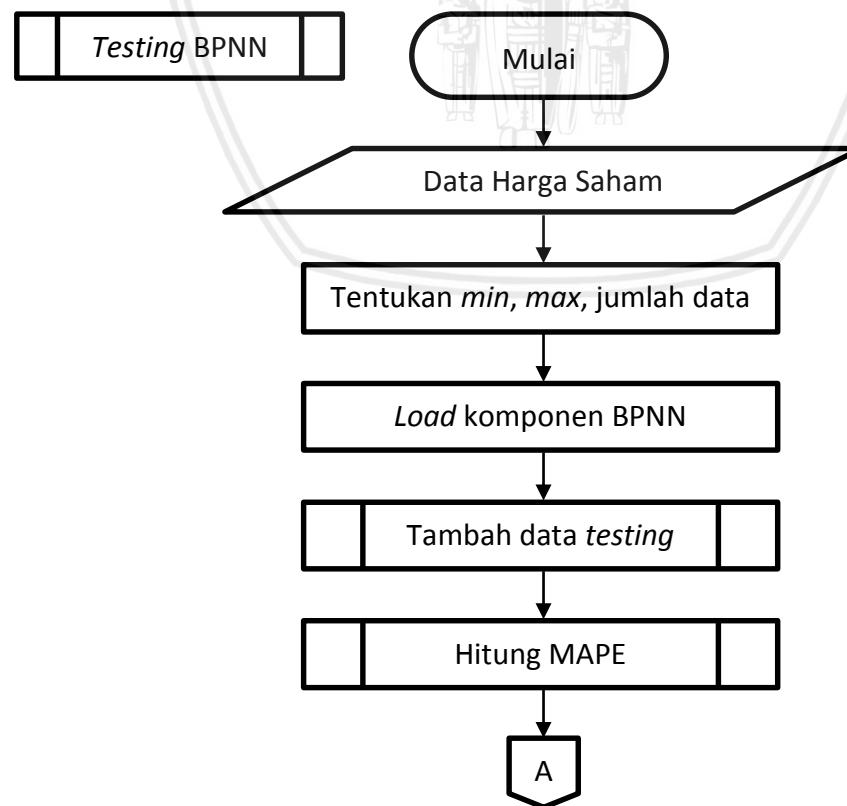


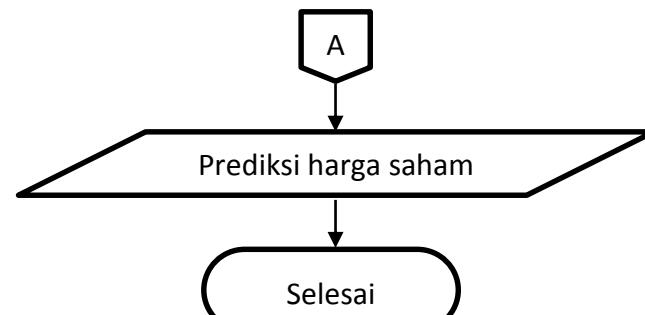


Gambar 4.11 Diagram alir simpan *global best ACO*

#### 4.2.5 Testing Backpropagation Neural Network

Diagram alir *testing* BPNN berisi tahapan-tahapan yang dimulai dari menerima masukan data harga saham. Kemudian menentukan nilai *min*, *max*, dan jumlah data. Lalu memuatkan komponen BPNN dari file berformat .txt untuk proses berikutnya. Selanjutnya menambahkan data *testing* dan melakukan perhitungan MAPE. Rangkaian tahapan tersebut akan menghasilkan harga saham hasil prediksi. Diagram alir *testing* BPNN dapat dilihat pada Gambar 4.12.

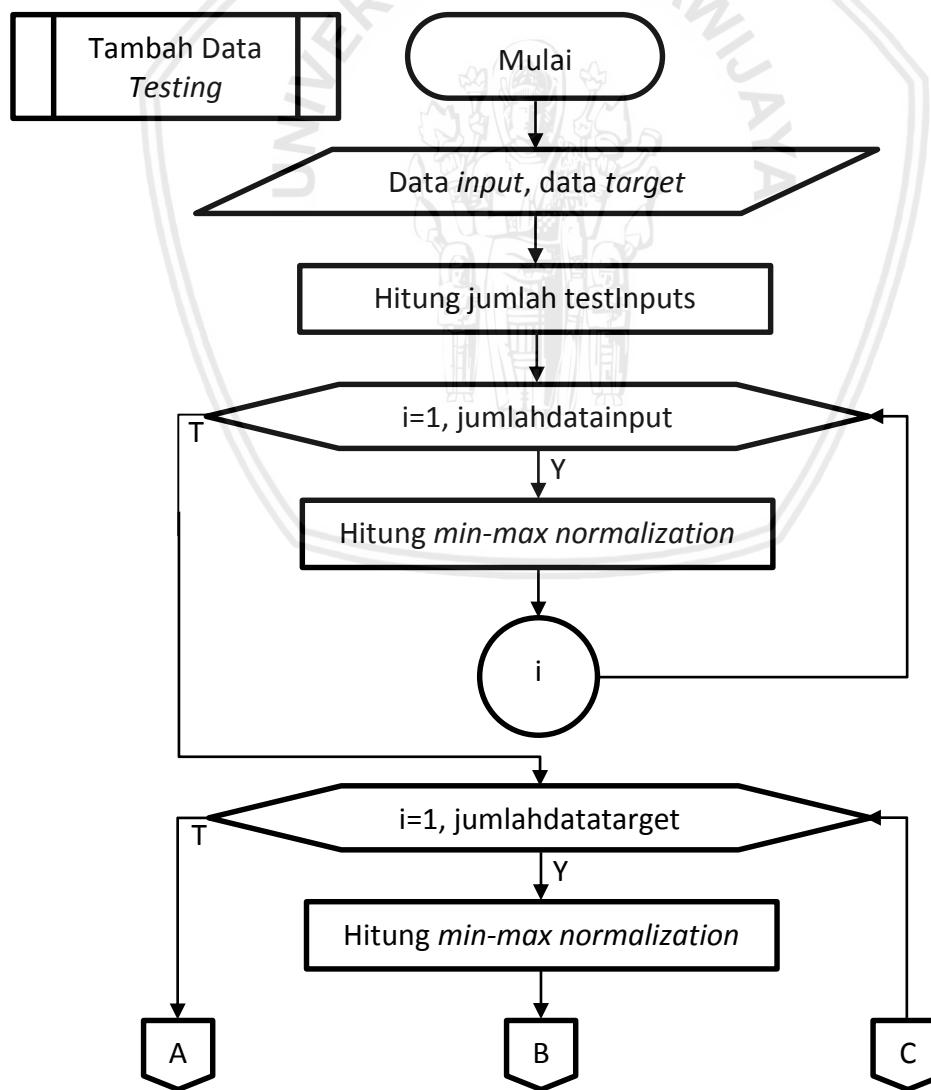


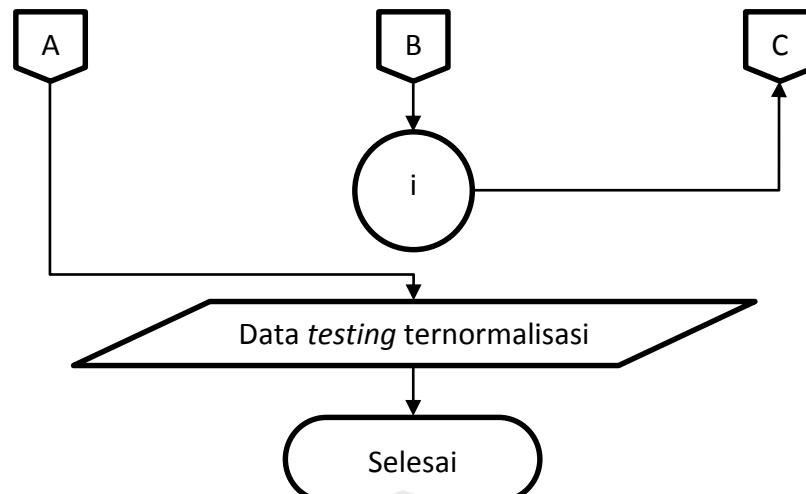


Gambar 4.12 Diagram alir *testing* BPNN

#### 4.2.5.1 Tambah Data *Testing*

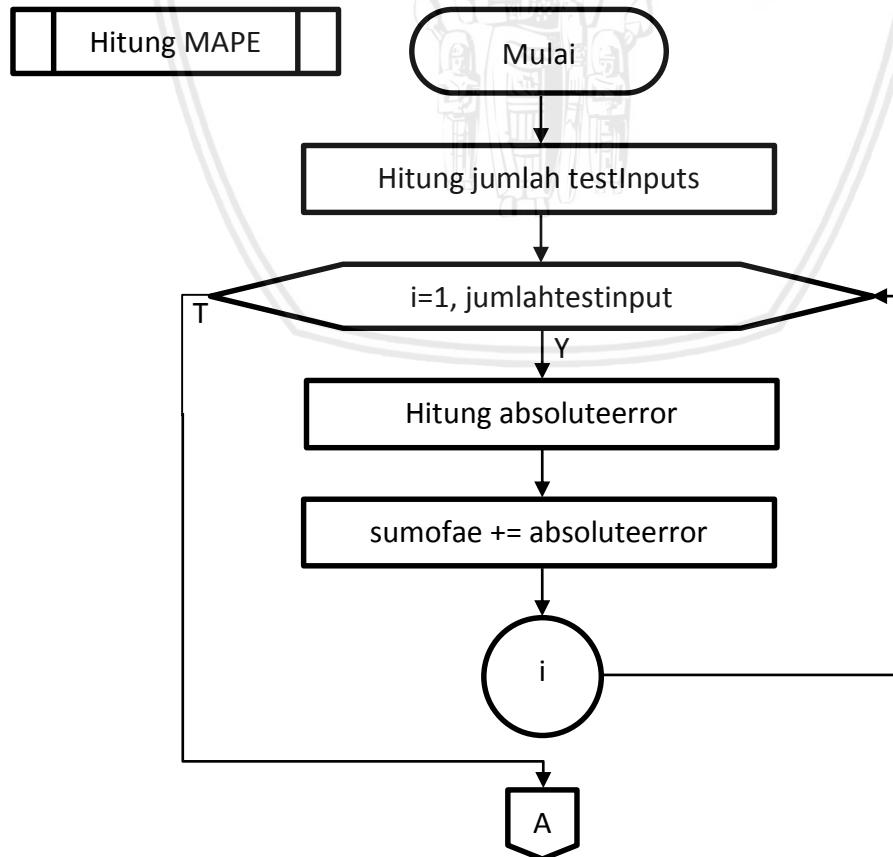
Tahap awal dalam menambahkan data *testing* adalah menerima masukan data *input* dan data *target*. Selanjutnya menghitung jumlah *testInputs* untuk menentukan *index* data saat ini. Kemudian melakukan dua kali perulangan sebanyak nilai jumlahdatainput dan nilai jumlahdatatarget untuk menghitung *min-max normalization*. Hasil yang diberikan berupa data *testing* ternormalisasi. Diagram alir tambah data *testing* dapat dilihat pada Gambar 4.13.

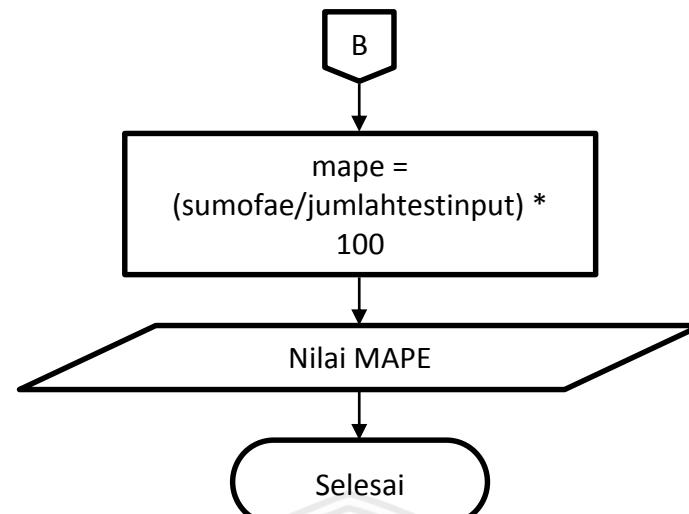


Gambar 4.13 Diagram alir tambah data *testing*

#### 4.2.5.2 Hitung MAPE

Tahapan pertama diagram alir perhitungan MAPE adalah menghitung jumlah testInputs untuk mengetahui banyaknya data *testing*. Kedua, melakukan perulangan sebanyak nilai jumlahtestinput untuk menghitung nilai absoluteError dan menjumlahkannya dengan nilai sumofae saat ini. Ketiga, membagi nilai sumofae dengan nilai jumlahtestinput, lalu dikalikan 100. MAPE yang dihasilkan merepresentasikan nilai *error* pada pengujian suatu kelompok data. Diagram alir perhitungan MAPE dapat dilihat pada Gambar 4.14.



Gambar 4.14 Diagram alir fase *testing*

### 4.3 Perhitungan Manual

#### 4.3.1 Perhitungan Ant Colony Optimization

##### Tahap 1: Inisialisasi parameter ACO

Inisialisasi parameter ACO dengan nilai acak sesuai intervalnya masing-masing. Parameter ACO tersebut adalah jumlah semut ( $m$ ), jumlah *node* yang harus dilewati setiap semut, iterasi maksimum ( $NCmax$ ), tetapan siklus semut ( $q0$ ), tetapan pengendali intensitas feromon ( $\alpha$ ), tetapan pengendali visibilitas ( $\beta$ ), tetapan penguapan feromon lokal ( $\rho$ ), dan tetapan penguapan feromon global ( $\delta$ ). Inisialisasi parameter ACO ditunjukkan pada Tabel 4.1.

Tabel 4.1 Nilai parameter ACO

Jumlah Semut (m)	Node	Iterasi	$q0$	$\alpha$	$\beta$	$\tau0$	$\rho$	$\delta$
2	8	1	0,5	1	1	0,5	0,3	0,4

Kemudian menentukan parameter BPNN yang akan dioptimasi dan batas-batas nilainya. Karena interval dari nilai kunjungan semut dan parameter BPNN berbeda, maka diperlukan representatif kelompok *node* bagi setiap parameter. Oleh sebab itu, parameter *learning rate*, momentum, dan jumlah *hidden node* dioptimasi melalui 8 *node* dalam ACO. Batas nilai parameter ditunjukkan pada Tabel 4.2, sedangkan pembagian kelompok *node* untuk setiap parameter dinyatakan di bawah ini.

- Node 1-3 untuk parameter *learning rate*
- Node 4-6 untuk parameter momentum
- Node 7-8 untuk parameter jumlah *hidden node*

**Tabel 4.2 Batas parameter BPNN**

<i>Learning Rate (<math>\alpha</math>)</i>	<i>momentum (<math>m</math>)</i>	Jumlah <i>hidden node</i> ( $j$ )
0,001 – 0,999	0,001 – 0,999	1 - 20

**Tahap 2: Menentukan *node* awal ( $r$ )**

Posisi awal ( $r$ ) sebelum setiap semut memulai perjalannya berada pada *node* 10. *Node* tersebut tidak termasuk dalam 8 *node* representatif parameter BPNN dan tidak dapat menjadi *node* tujuan ( $u$ ) pada kunjungan semut. *Node* pertama untuk optimasi parameter BPNN ditentukan di tahap 3.

**Tahap 3: Menentukan *node* selanjutnya ( $u$ )**

- Menentukan nilai ( $q$ )

Nilai ( $q$ ) ditentukan secara acak dengan nilai antara 0 sampai 1, dimisalkan ( $q$ ) = 0,7. Berikutnya melalui seleksi kondisi untuk penentuan *node* selanjutnya. Kondisi pertama, jika nilai  $q \leq q_0$ , maka menghitung *argument max*. Kondisi kedua, jika  $q > q_0$ , maka melakukan seleksi *roulette wheel*.

- Menentukan *node* selanjutnya ( $u$ )

Pemilihan *node* selanjutnya ( $u$ ) dihitung berdasarkan probabilitas perpindahan dari *node* saat ini menuju beberapa *node* selanjutnya. Probabilitas perpindahan tersebut nilainya bergantung pada intensitas feromon di rute yang bersangkutan. Intensitas feromon di seluruh rute beserta *node* 10 ditunjukkan pada Tabel 4.3.

**Tabel 4.3 Intensitas feromon awal**

$\tau_{ru}$	0	1	2	3	4	5	6	7	8	9
0	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
1	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
2	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
3	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
4	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
6	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
7	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
8	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
9	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
10	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5

Dimisalkan  $q > q_0$ , sehingga melakukan seleksi *roulette wheel* untuk memilih *node* selanjutnya. Nilai probabilitas perpindahan  $P(r, u)$  dihitung menggunakan Persamaan 2.20. Hasil perhitungan probabilitas perpindahan semut ditunjukkan pada Tabel 4.4.

**Tabel 4.4 Nilai probabilitas perpindahan semut**

$\tau_{ru}$	0	1	2	3	4	5	6	7	8	9
0	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
2	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
3	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
4	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
5	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
6	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
7	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
8	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
9	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
10	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1

Berikut adalah contoh perhitungan probabilitas perpindahan semut dengan  $(r) = 10$  menuju  $(u) = 0$

$$P(10,0) = \frac{0,1}{0,1 + 0,1 + 0,1 + 0,1 + 0,1 + 0,1 + 0,1 + 0,1 + 0,1 + 0,1} = 0,1$$

- Menghitung probabilitas kumulatif ( $P_k$ )

Nilai ( $P_k$ ) suatu *node* terhadap *node* tujuan bergantung pada nilai probabilitas perpindahan menuju *node* 0 hingga *node* tujuan. Berikut adalah contoh perhitungan probabilitas kumulatif *node* 10 menuju *node* 2. Hasil perhitungan probabilitas kumulatif *node* 10 terhadap setiap *node* tujuan ditunjukkan pada Tabel 4.5.

$$P_k(10,2) = 0,1 + 0,1 + 0,1 = 0,3$$

**Tabel 4.5 Nilai probabilitas kumulatif *node* 10**

$\tau(r, u)$	0	1	2	3	4	5	6	7	8	9
10	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
Pk10	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1

Setelah itu menentukan nilai acak dalam interval 0 sampai 1. Dimisalkan nilai acaknya adalah  $(k) = 0,2197$ , maka  $(Pk)$  yang terpilih adalah 0,3 dan *node* selanjutnya adalah *node* 2.

#### Tahap 4: Memperbarui feromon lokal

Memperbarui feromon pada jalur yang telah dilewati semut, yaitu antara *node* 10 dan *node* 2 dengan Persamaan 2.21. Berikut adalah contoh perhitungan pembaruan feromon lokal untuk  $\tau(10,2)$ . Intensitas feromon di seluruh jalur setelah pembaruan feromon lokal ditunjukkan pada Tabel 4.6.

$$\tau(10,2) = (1 - 0,3) \times 0,5 + (0,3 \times 0,5) = 0,5$$

**Tabel 4.6 Intensitas feromon setelah *update* feromon lokal**

$\tau_{ru}$	0	1	2	3	4	5	6	7	8	9
0	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
1	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
2	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
3	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
4	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
6	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
7	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
8	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
9	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
10	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5

#### Tahap 5: *Update* nilai (*r*) dengan nilai (*u*)

Mengganti nilai (*r*) yang semula 10 dengan nilai (*u*) yaitu 2. Lalu kembali ke tahap 3, ulangi proses ini hingga seluruh *node* terlewati. Jika seluruh *node* telah dilewati, maka semut berikutnya memulai perjalanannya.

#### Tahap 6: Konversi parameter BPNN

Ketika semua semut telah menyelesaikan perjalanannya, rute perjalanan setiap semut akan dikonversi menjadi parameter BPNN. *Node-node* kunjungan semut dikonversi berdasarkan pembagian representatif dan interval parameter BPNN pada Tabel 4.2. Perhitungan konversi parameter BPNN untuk *node* 1-6 menggunakan Persamaan 2.27, 2.28, dan 2.29. Lalu untuk *node* 7-8 menggunakan Persamaan 2.30, 2.31, dan 2.32.

Contoh perhitungan konversi parameter *learning rate*:

$$A = ((2 \times 100) + (1 \times 10) + (2 \times 1)) \div 1000$$

$$A = 0,212$$

$$Norm\alpha = \left[ \left( \frac{0,212 - 0,001}{0,999 - 0,001} \right) \times (0,999 - 0,001) \right] + 0,001$$

$$Norm\alpha = 0,212$$

Contoh perhitungan konversi parameter momentum:

$$A = ((7 \times 100) + (6 \times 10) + (7 \times 1)) \div 1000$$

$$A = 0,767$$

$$Normm = \left[ \left( \frac{0,767 - 0,001}{0,999 - 0,001} \right) \times (0,999 - 0,001) \right] + 0,001$$

$$Normm = 0,767$$

Contoh perhitungan konversi parameter jumlah *hidden node*:

$$A = ((5 \times 10) + (8 \times 1))$$

$$A = 58$$

$$cap \geq 58, \quad cap = 5 \times i$$

$$cap = 60, \quad i = 12$$

$$Normj = 60 \div 5$$

$$Normj = 12$$

### Tahap 7: Memperbarui feromon global

Ketika kombinasi parameter setiap semut telah diketahui nilai RMSE-nya, maka intensitas feromon di seluruh rute dapat diperbarui. Pembaruan intensitas feromon tersebut dipengaruhi oleh nilai RMSE dari masing-masing rute perjalanan semut. Dipilih satu rute terbaik sebagai *global best* dengan nilai RMSE terkecil. Dimisalkan selain RMSE semut pertama pada Tabel 4.27, semut kedua memiliki RMSE bernilai 0,1. Sehingga semut kedua menjadi *global best* dengan rute perjalanan *node* 10-1-0-0-6-2-0-1-2. Selanjutnya menghitung matriks perubahan intensitas feromon global dengan Persamaan 2.23.

$$\Delta\tau(1,0) = \frac{1}{0,1} = 10$$

Lalu menghitung pembaruan feromon global pada jalur (1,0) menggunakan Persamaan 2.22.

$$\tau(1,0) = (1 - 0,4) \times (0,5) + (0,4 \times 10) = 4,3$$

Hasil *update* feromon global di seluruh rute ditunjukkan pada Tabel 4.7.

**Tabel 4.7 Intensitas feromon setelah *update feromon global***

$\tau_{ru}$	0	1	2	3	4	5	6	7	8	9
0	4,3	4,3	0,3	0,3	0,3	0,3	4,3	0,3	0,3	0,3
1	4,3	0,3	4,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3
2	4,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3
3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3
4	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3
5	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3
6	0,3	0,3	4,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3
7	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3
8	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3
9	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3
10	0,3	4,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3

Setelah pembaruan feromon global selesai, kondisi *stop* pada iterasi ACO diperiksa. Jika kondisi *stop* belum terpenuhi, maka iterasi diulang dari tahap 2, yaitu meletakkan setiap semut pada node awal. Apabila kondisi *stop* terpenuhi, maka tahapan dilanjutkan menuju fase *testing* BPNN.

#### 4.3.2 Perhitungan *Backpropagation Neural Network*

##### Tahap 1: *Input* data pelatihan

Memasukkan data pelatihan yang merupakan *close price* saham harian Bank Rakyat Indonesia (Persero) Tbk. periode 1 Januari 2018 sampai 31 Desember 2018. Sampel data yang sesuai dengan visualisasi Tabel 3.2 ditunjukkan pada Tabel 4.8.

**Tabel 4.8 Data latih**

Fitur	X1	X2	X3	Target
Data	2990	3040	3090	3080

##### Tahap 2: Normalisasi data pelatihan

Mencari nilai minimum dan maksimum data *input* seperti yang ditunjukkan pada Tabel 4.9.

**Tabel 4.9 Nilai minimum dan maksimum data latih**

Nilai minimum	2990
Nilai maksimum	3090

Lalu melakukan normalisasi data latih menggunakan Persamaan 2.24. Hasil perhitungan ditunjukkan pada Tabel 4.10.

$$x' = \frac{(x - x_{min})}{(x_{max} - x_{min})}$$

$$x' = \frac{(3040 - 2990)}{(3090 - 2990)}$$

$$x' = \frac{50}{100}$$

$$x' = 0,5$$

**Tabel 4.10 Data latih ternormalisasi**

Fitur	X1	X2	X3	Target
Data	0	0,5	1	0,9

### Tahap 3: Fase *training*

Data yang telah dinormalisasi selanjutnya melalui tahapan *training* atau pelatihan jaringan. Dalam tahapan *training* terdapat proses *feedforward* dan *backpropagate*. Pelatihan jaringan dilakukan terus-menerus hingga memenuhi kondisi berhenti. Kondisi berhenti dapat berupa jumlah iterasi maksimal atau toleransi *error* maksimal.

#### 1. Inisialisasi parameter BPNN

Dimisalkan tahapan optimasi ACO sebelumnya menghasilkan nilai *learning rate* sebesar 0,1 dan momentum sebesar 0,1. Kemudian untuk arsitekturnya memiliki 4 *node input*, 2 *node hidden*, dan 1 *node output*.

#### 2. Inisialisasi bobot dan *threshold*

Inisialisasi dengan nilai acak pada interval -0,25 sampai 0,25. Nilai bobot dan *threshold* ditunjukkan pada Tabel 4.11, 4.12, 4.13, dan 4.14.

**Tabel 4.11 Bobot menuju *hidden layer***

$v_{ji}$	$v_{10}$	$v_{11}$	$v_{12}$	$v_{13}$
Nilai	0,25	-0,25	0,25	-0,25

**Tabel 4.12 Bobot menuju *output layer***

$w_{kj}$	$w_{10}$	$w_{11}$
Nilai	-0,25	0,25

**Tabel 4.13 Threshold pada hidden layer**

$\theta_j$	$\theta_1$
Nilai	0,25

**Tabel 4.14 Threshold pada output layer**

$\theta_k$	$\theta_1$
Nilai	-0,25

### 3. Proses feedforward

Tujuan dalam tahapan ini adalah untuk memperoleh nilai yang akan digunakan pada lapisan selanjutnya. Dimulai dari menghitung keluaran pada *hidden layer* menggunakan Persamaan 2.3. Hasil dari perhitungan tersebut ditampilkan pada Tabel 4.15.

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji} - \theta_j$$

$$z_{net_1} = ((1 \times 0,25) + (0 \times -0,25) + (0,5 \times 0,25) + (1 \times -0,25)) \\ - 0,25$$

$$z_{net_1} = -0,125$$

**Tabel 4.15 Hasil perhitungan  $z_{net_j}$** 

$z_{net_j}$	$z_{net_1}$
Nilai	-0,125

Melakukan aktivasi menggunakan fungsi aktivasi sigmoid biner pada Persamaan 2.4. Hasil aktivasi tersebut ditunjukkan pada Tabel 4.16.

$$f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}}$$

$$f(z_{net_1}) = \frac{1}{1 + e^{-0,125}}$$

$$f(z_{net_1}) = \frac{1}{1,8824969}$$

$$f(z_{net_1}) = 0,5312$$

**Tabel 4.16 Hasil aktivasi  $z_{net_j}$** 

$z_j$	$z_1$
Nilai	0,5312

Kemudian menghitung nilai keluaran pada *output layer* menggunakan Persamaan 2.5. Hasil dari perhitungan tersebut ditunjukkan pada Tabel 4.17.

$$y_{net_k} = w_{k0} + \sum_{j=1}^n z_j w_{kj} - \theta_k$$

$$y_{net_1} = ((1 \times -0,25) + (0,5312 \times 0,25)) - (-0,25)$$

$$y_{net_1} = 0,1328$$

**Tabel 4.17 Hasil perhitungan  $y_{net_k}$**

$y_{net_k}$	$y_{net_1}$
Nilai	0,1328

Melakukan aktivasi menggunakan fungsi aktivasi sigmoid biner pada Persamaan 2.6. Hasil aktivasi tersebut ditunjukkan pada Tabel 4.18.

$$f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}}$$

$$f(y_{net_1}) = \frac{1}{1 + e^{0,1328}}$$

$$f(y_{net_1}) = \frac{1}{2,142022}$$

$$f(y_{net_1}) = 0,4668486$$

**Tabel 4.18 Hasil aktivasi  $y_{net_k}$**

$y_k$	$y_1$
Nilai	0,4668486

#### 4. Proses *backpropagate*

Propagasi balik bertujuan untuk memperbaiki bobot dan *threshold* berdasarkan *error* pada lapisan *output*. Langkah pertama adalah menghitung faktor kesalahan unit *output layer* menggunakan Persamaan 2.7. Hasil dari perhitungan tersebut ditunjukkan pada Tabel 4.19.

$$\delta_k = (t_k - y_k)y_k(1 - y_k)$$

$$\delta_1 = (t_1 - y_1)y_1(1 - y_1)$$

$$\delta_1 = (0,9 - 0,4668)0,4668(1 - 0,4668)$$

$$\delta_1 = 0,4332 \times 0,4668 \times 0,5332$$

$$\delta_1 = 0,10782251$$

**Tabel 4.19 Hasil perhitungan  $\delta_k$** 

$\delta_k$	$\delta_1$
Nilai	0,10782251

Kemudian dilanjutkan dengan menghitung faktor kesalahan unit *hidden layer* menggunakan Persamaan 2.8 dan 2.9. Pengecualian untuk ( $\delta_0$ ) tidak dihitung karena *node* bias selalu bernilai 1. Hasil perhitungan tersebut ditunjukkan pada Tabel 4.20.

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

$$\delta_{net_1} = \delta_1 w_{11}$$

$$\delta_{net_1} = (0,1078 \times 0,25)$$

$$\delta_{net_1} = 0,02695$$

$$\delta_j = (\delta_{net_j})z_j(1 - z_j)$$

$$\delta_1 = (\delta_{net_1})z_1(1 - z_1)$$

$$\delta_1 = (0,02695)0,5312(1 - 0,5312)$$

$$\delta_1 = (0,02695) \times 0,5312 \times (0,4688)$$

$$\delta_1 = 0,006711266$$

**Tabel 4.20 Hasil perhitungan  $\delta_j$** 

$\delta_j$	$\delta_1$
Nilai	0,006711266

Faktor kesalahan unit *output layer* digunakan untuk menghasilkan koreksi  $w_{kj}$ . Perhitungan ini melibatkan parameter *learning rate* dan nilai *node* yang bersangkutan sesuai dengan Persamaan 2.10. Hasil perhitungan tersebut ditampilkan pada Tabel 4.21.

$$\Delta w_{kj} = \alpha \delta_k z_j$$

$$\Delta w_{10} = \alpha \delta_1 z_0$$

$$\Delta w_{10} = 0,1 \times 0,1078 \times 1$$

$$\Delta w_{10} = 0,01078$$

**Tabel 4.21 Hasil perhitungan  $\Delta w_{kj}$** 

$\Delta w_{kj}$	$\Delta w_{10}$	$\Delta w_{11}$
Nilai	0,01078	0,00572

Berikutnya menghitung koreksi  $v_{ji}$  berdasarkan faktor kesalahan unit *hidden layer* menggunakan Persamaan 2.11. Hasil dari perhitungan dapat dilihat pada Tabel 4.22.

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

$$\Delta v_{10} = \alpha \delta_1 x_0$$

$$\Delta v_{10} = 0,1 \times 0,0067 \times 1$$

$$\Delta v_{11} = 0,00067$$

**Tabel 4.22 Hasil perhitungan  $\Delta v_{ji}$** 

$\Delta v_{ji}$	$\Delta v_{10}$	$\Delta v_{11}$	$\Delta v_{12}$	$\Delta v_{13}$
Nilai	0,00067	0	0,000335	0,00067

Setelah menghitung koreksi setiap bobot, maka proses dilanjutkan pada bagian memperbarui seluruh bobot. Dimulai dari bobot antara lapisan *hidden* dan lapisan *output*. Hasil dari perhitungan yang menggunakan Persamaan 2.12 ini ditampilkan pada Tabel 4.23.

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

$$w_{10}(\text{baru}) = w_{10}(\text{lama}) + \Delta w_{10}(\text{baru}) + m \times \Delta w_{10}(\text{lama})$$

$$w_{10}(\text{baru}) = -0,25 + 0,01078 + 0,1 \times 0$$

$$w_{10}(\text{baru}) = -0,23922$$

**Tabel 4.23 Hasil update  $w_{kj}$** 

$w_{kj}$	$w_{10}$	$w_{11}$
Nilai	-0,23922	0,25572

Berikutnya memperbarui bobot antara lapisan *input* dan lapisan *hidden* menggunakan Persamaan 2.13. Hasil perhitungan ditampilkan pada Tabel 4.24.

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

$$v_{10}(\text{baru}) = v_{10}(\text{lama}) + \Delta v_{10}(\text{baru}) + m \times \Delta v_{10}(\text{lama})$$

$$v_{10}(\text{baru}) = 0,25 + 0,00067 + 0,1 \times 0$$

$$v_{10}(\text{baru}) = 0,25067$$

**Tabel 4.24 Hasil update  $v_{ji}$** 

$v_{ji}$	$v_{10}$	$v_{11}$	$v_{12}$	$v_{13}$
Nilai	0,25067	-0,25	0,250335	-0,24933

Selanjutnya memperbarui *threshold* menggunakan parameter *learning rate* dan faktor kesalahan pada unit yang bersangkutan. Diawali dengan menghitung koreksi *threshold* dan memperbarui *threshold* di unit *output layer* menggunakan Persamaan 2.14 dan 2.16. Hasil proses tersebut ditunjukkan pada Tabel 4.25.

$$\Delta\theta_k = \alpha(-1)\delta_k$$

$$\Delta\theta_1 = \alpha(-1)\delta_1$$

$$\Delta\theta_1 = 0,1 \times (-1) \times -0,1078$$

$$\Delta\theta_1 = 0,01078$$

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

$$\theta_1(\text{baru}) = \theta_1(\text{lama}) + \Delta\theta_1$$

$$\theta_1(\text{baru}) = -0,25 + 0,01078$$

$$\theta_1(\text{baru}) = -0,23922$$

**Tabel 4.25 Hasil update  $\theta_k$** 

$\theta_k$	$\theta_1$
Nilai	-0,23922

Kemudian menghitung koreksi *threshold* dan memperbarui *threshold* di unit *hidden layer* menggunakan Persamaan 2.15 dan 2.17. Hasil perhitungan tersebut ditampilkan pada Tabel 4.26.

$$\Delta\theta_j = \alpha(-1)\delta_j$$

$$\Delta\theta_j = 0,1 \times (-1) \times 0,0067$$

$$\Delta\theta_j = -0,00067$$

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

$$\theta_j(\text{baru}) = 0,25 + (-0,00067)$$

$$\theta_j(\text{baru}) = 0,24933$$

**Tabel 4.26 Hasil update  $\theta_j$** 

$\theta_j$	$\theta_1$
Nilai	0,24933

### 5. Menghitung RMSE

Pada setiap satu kali iterasi pelatihan jaringan akan dilakukan perhitungan RMSE. RMSE merupakan salah satu syarat selesainya pelatihan ketika nilainya telah lebih kecil atau sama dengan batas maksimal. Jika nilai RMSE masih lebih besar dari batas maksimal, maka mengulang dari proses *feedforward*. Perhitungan RMSE menggunakan Persamaan 2.18 dan hasilnya dapat dilihat pada Tabel 4.27.

$$RMSE = \sqrt{\left( \frac{\sum_n \sum_k (t_{kn} - o_{kn})^2}{nk} \right)}$$

$$RMSE = \sqrt{\left( \frac{(t_{11} - o_{11})^2}{1 \times 1} \right)}$$

$$RMSE = \sqrt{(0,4668 - 0,9)^2}$$

$$RMSE = \sqrt{(-0,4332)^2}$$

$$RMSE = 0,18766224$$

**Tabel 4.27 Hasil perhitungan RMSE**

Nilai RMSE	0,18766224
------------	------------

### Tahap 4: Normalisasi data pengujian

Pengujian ini menggunakan bobot, *threshold*, dan arsitektur jaringan yang berasal dari fase *training* BPNN. Fase *training* tersebut dilakukan dengan parameter hasil konversi rute semut *global best* pada iterasi terakhir ACO. Sebelum melakukan pengujian, terlebih dahulu memasukkan data pengujian yang sampel datanya ditampilkan pada Tabel 4.28.

**Tabel 4.28 Data uji**

Fitur	X1	X2	X3	Target
Data	3040	3090	3080	3070

Mencari nilai minimum dan maksimum data *input* seperti yang ditunjukkan pada Tabel 4.29.

**Tabel 4.29 Nilai minimum dan maksimum data uji**

Nilai minimum	3040
Nilai maksimum	3090

Lalu melakukan normalisasi data uji menggunakan Persamaan 2.24. Hasil perhitungan ditunjukkan pada Tabel 4.30.

$$x' = \frac{(x - x_{min})}{(x_{max} - x_{min})}$$

$$x' = \frac{(3080 - 3040)}{(3090 - 3040)}$$

$$x' = \frac{40}{50}$$

$$x' = 0,8$$

**Tabel 4.30 Data uji ternormalisasi**

Fitur	X1	X2	X3	Target
Data	0	1	0,8	0,6

#### Tahap 5: Fase testing

Fase pengujian dimulai dari melakukan satu kali proses *feedforward* menggunakan data uji ternormalisasi. Proses *feedforward* diawali dengan menghitung keluaran pada *hidden layer* berdasarkan Persamaan 2.3. Hasil perhitungan tersebut ditunjukkan pada Tabel 4.31.

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji} - \theta_j$$

$$z_{net_1} = ((1 \times 0,25067) + (0 \times -0,25) + (1 \times 0,250335) \\ + (0,8 \times -0,24933)) - 0,24933$$

$$z_{net_1} = 0,052211$$

**Tabel 4.31 Hasil perhitungan  $z_{net_j}$** 

$z_{net_j}$	$z_{net_1}$
Nilai	0,052211

Setelah itu melakukan aktivasi menggunakan Persamaan 2.4 dan hasilnya ditampilkan pada Tabel 4.32.

$$f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}}$$

$$f(z_{net_1}) = \frac{1}{1 + e^{-0,052211}}$$

$$f(z_{net_1}) = \frac{1}{1,94912858}$$

$$f(z_{net_1}) = 0,5130498$$

**Tabel 4.32 Hasil aktivasi  $z_{net_j}$** 

$z_j$	$z_1$
Nilai	0,5130498

Kemudian menghitung nilai keluaran pada *output layer* menggunakan Persamaan 2.5. Hasil dari perhitungan tersebut ditunjukkan pada Tabel 4.33.

$$y_{net_k} = w_{k0} + \sum_{j=1}^n z_j w_{kj} - \theta_k$$

$$y_{net_1} = ((1 \times -0,23922) + (0,5130498 \times 0,25572)) - (-0,23922)$$

$$y_{net_1} = 0,1311971$$

**Tabel 4.33 Hasil perhitungan  $y_{net_k}$** 

$y_{net_k}$	$y_{net_1}$
Nilai	0,1311971

Melakukan aktivasi menggunakan Persamaan 2.6 dan hasilnya ditampilkan pada Tabel 4.34.

$$f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}}$$

$$f(y_{net_1}) = \frac{1}{1 + e^{-0,1311971}}$$

$$f(y_{net_1}) = \frac{1}{1,877045}$$

$$f(y_{net_1}) = 0,5327523$$

**Tabel 4.34 Hasil aktivasi  $y_{net_k}$** 

$y_k$	$y_1$
Nilai	0,5327523

### Tahap 6: Denormalisasi data pengujian

Sebelum menghitung MAPE fase pengujian, perlu dilakukan denormalisasi data menggunakan Persamaan 2.25. Hasil denormalisasi ditunjukkan pada Tabel 4.35.

$$\begin{aligned}x &= x'(x_{max} - x_{min}) + x_{min} \\x &= 0,5327523(3090 - 3040) + 3040 \\x &= 0,5327523(50) + 3040 \\x &= 26,637615 + 3040 \\x &= 3066,63 \\x &= 3067\end{aligned}$$

**Tabel 4.35 Hasil denormalisasi**

Fitur	X1	X2	X3	Target	Output
Data	3040	3090	3080	3070	3067

### Tahap 7: Menghitung MAPE

Menghitung MAPE dari hasil prediksi dan target data uji menggunakan Persamaan 2.26. Hasil dari perhitungan tersebut ditunjukkan pada Tabel 4.36.

$$MAPE = \frac{\sum_{t=1}^n \left| \frac{x_t - f_t}{x_t} \right|}{n} \times 100\%$$

$$MAPE = \frac{\sum_{t=1}^1 \left| \frac{x_t - f_t}{x_t} \right|}{1} \times 100\%$$

$$MAPE = \frac{\left| \frac{x_1 - f_1}{x_1} \right|}{1} \times 100\%$$

$$MAPE = \frac{\left| \frac{3070 - 3067}{3070} \right|}{1} \times 100\%$$

$$MAPE = \frac{0,0009771987}{1} \times 100\%$$

$$MAPE = 0,000977 \times 100\%$$

$$MAPE = 0,0977\%$$

**Tabel 4.36 Nilai MAPE**

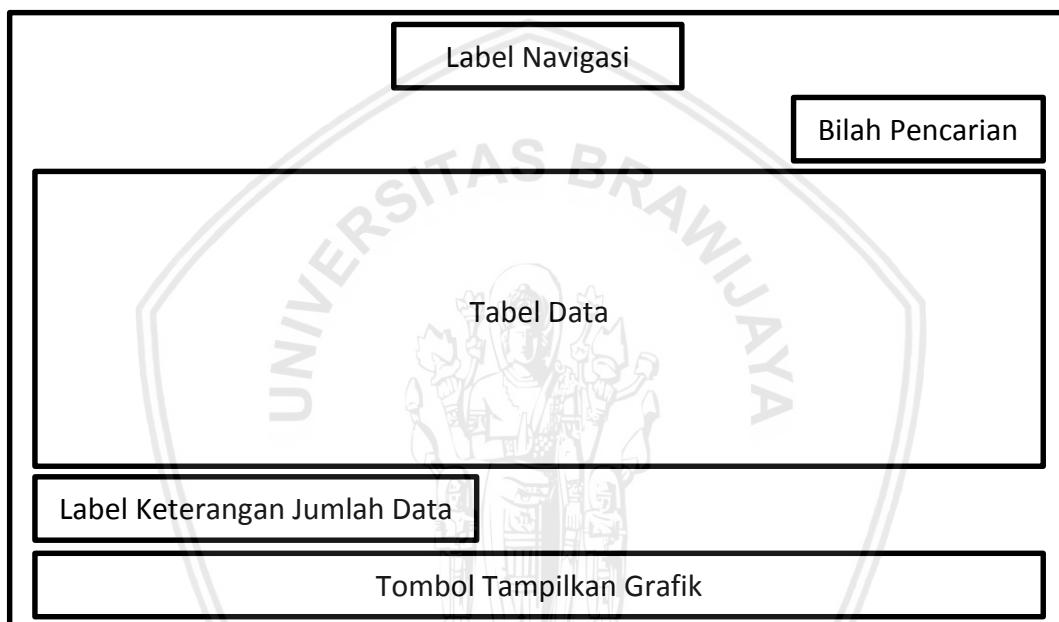
MAPE	0,0977%
------	---------

## 4.4 Perancangan Antarmuka

Terdapat 3 perancangan antarmuka yang akan ditampilkan, yaitu halaman data mentah, halaman pelatihan ACO-BPNN, dan halaman pengujian ACO-BPNN. Pada masing-masing gambar perancangan antarmuka tersebut juga dicantumkan nama dari komponen-komponennya.

### 4.4.1 Perancangan Antarmuka Halaman Data Mentah

Fungsi halaman ini adalah menampilkan data mentah yang telah diunduh dari situs *Yahoo Finance*. Data disajikan dalam sebuah tabel yang dibawahnya tersedia tombol untuk melihat grafik pergerakan *close price* dari data tersebut. Perancangan antarmuka ditunjukkan pada Gambar 4.15.



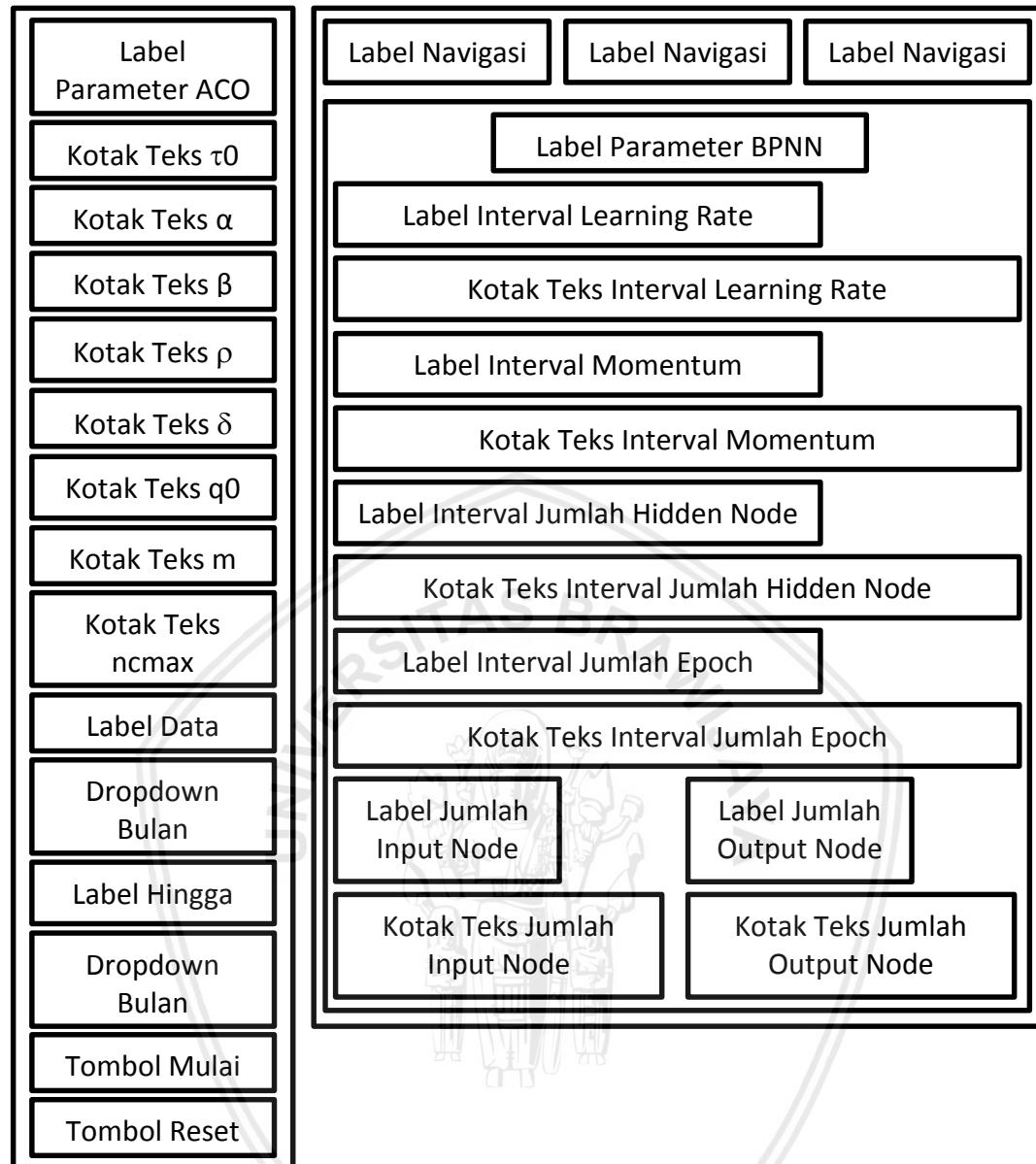
Gambar 4.15 Perancangan antarmuka halaman data mentah

### 4.4.2 Perancangan Antarmuka Halaman Pelatihan ACO-BPNN

Fungsi halaman ini adalah menerima masukan berupa parameter ACO dan data latih untuk proses pelatihan. Selain itu, halaman ini menampilkan informasi pada saat sebelum pelatihan, saat pelatihan berlangsung, dan hasil pelatihan. Salah satu dari ketiga informasi tersebut ditampilkan ketika *tab menu* prapelatihan, proses pelatihan, atau hasil pelatihan sedang aktif. Sehingga ketika salah satu *tab menu* sedang aktif, maka *tab menu* lainnya tidak aktif.

#### 4.4.2.1 Tab Menu Prapelatihan Aktif

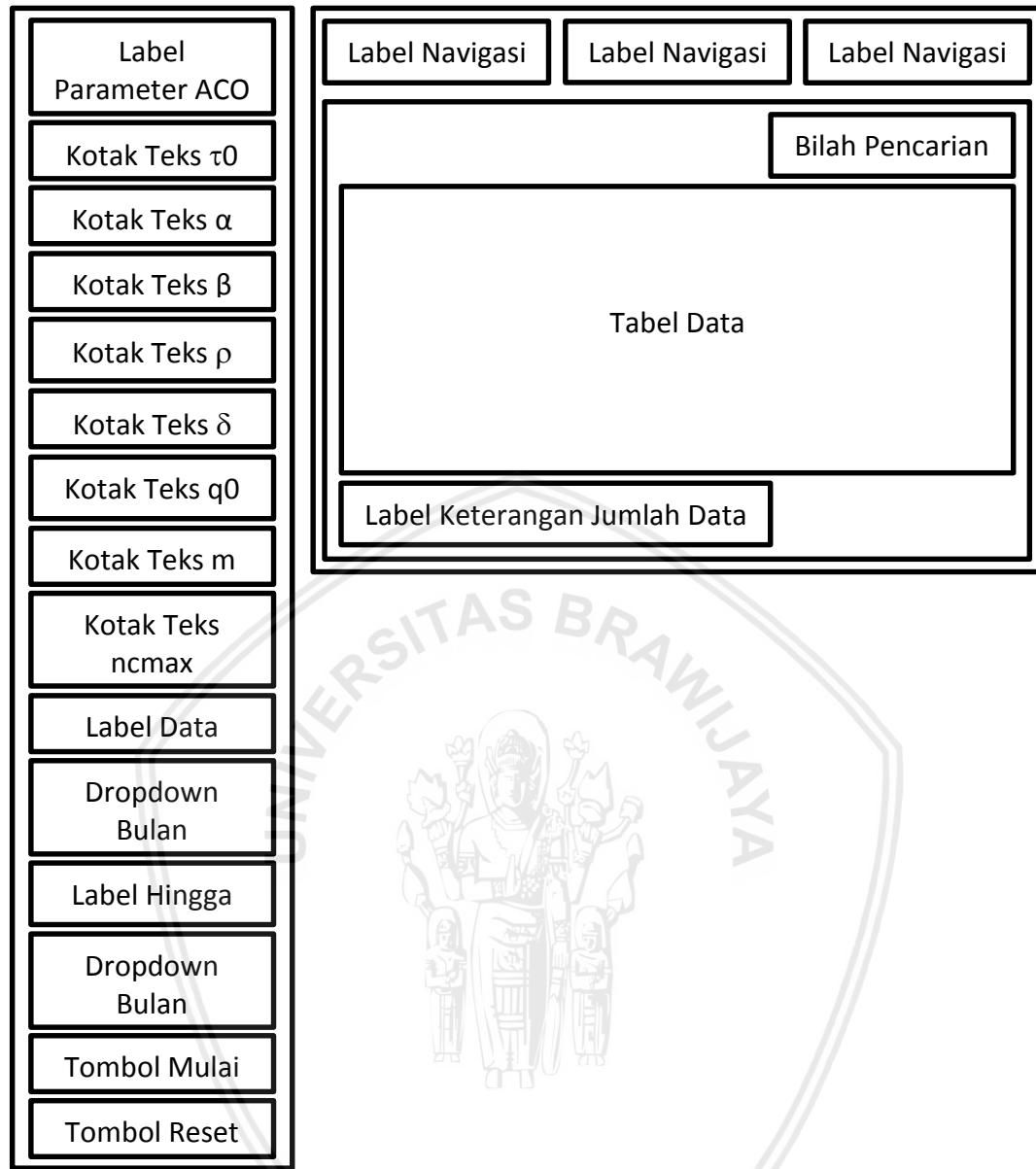
Ketika *tab menu* ini aktif, maka halaman akan menampilkan 2 *card*. *Card* di sebelah kiri berfungsi untuk menerima masukan parameter ACO. *Card* di sebelah kanan berfungsi untuk menampilkan beberapa parameter BPNN yang digunakan untuk proses pelatihan. Perancangan antarmuka ditunjukkan pada Gambar 4.16.



Gambar 4.16 Perancangan antarmuka *tab menu* prapelatihan aktif

#### 4.4.2.2 Tab Menu Proses Pelatihan Aktif

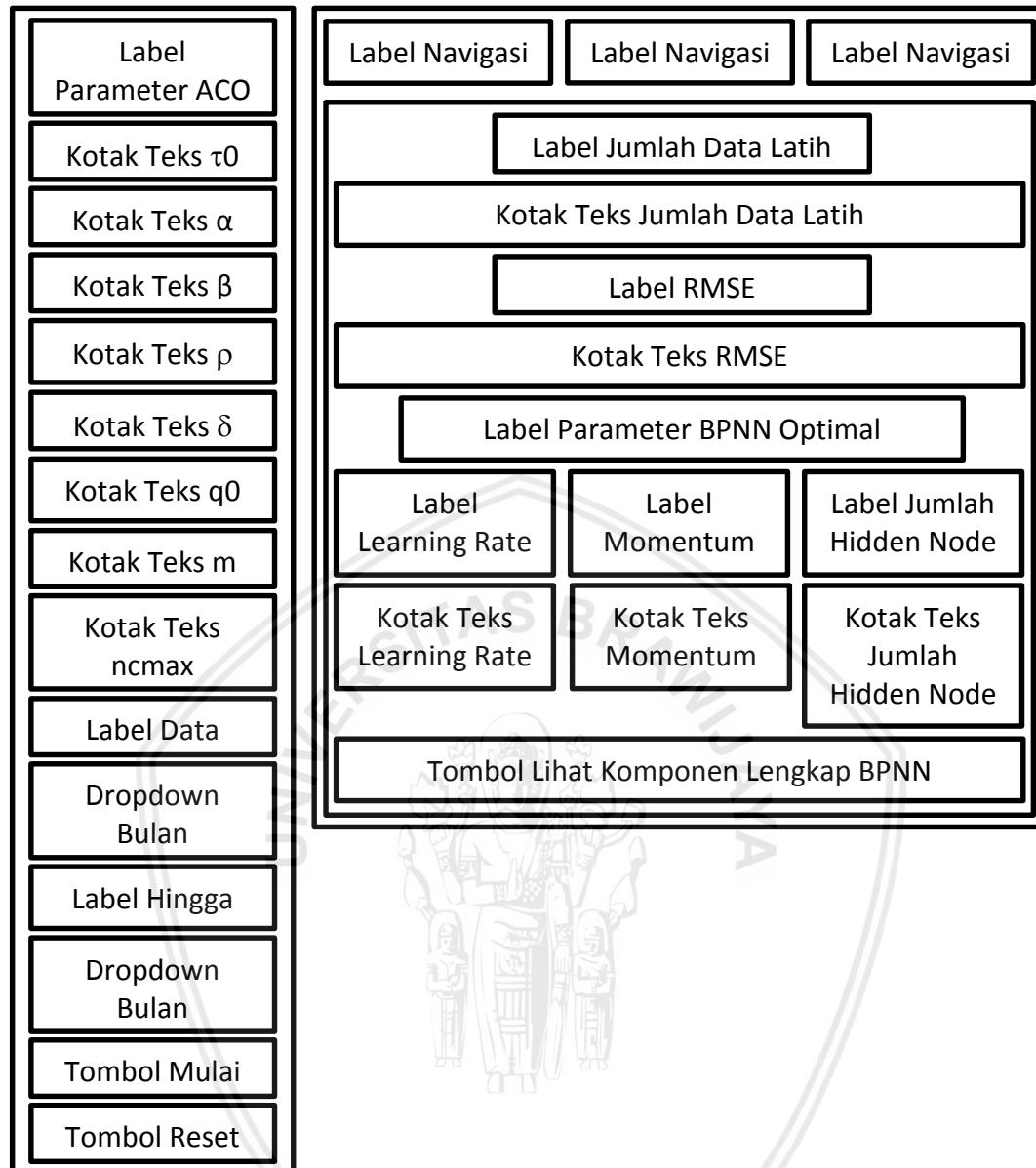
Ketika *tab menu* ini aktif, maka halaman akan menampilkan 2 *card*. *Card* di sebelah kiri berfungsi untuk menampilkan parameter ACO yang telah dimasukkan sebelumnya. *Card* di sebelah kanan berfungsi untuk menampilkan *global best ACO* di setiap iterasi. Perancangan antarmuka ditunjukkan pada Gambar 4.17.



Gambar 4.17 Perancangan antarmuka *tab menu* proses pelatihan aktif

#### 4.4.2.3 Tab Menu Hasil Pelatihan Aktif

Ketika *tab menu* ini aktif, maka halaman akan menampilkan 2 *card*. *Card* di sebelah kiri berfungsi untuk menampilkan parameter ACO yang telah dimasukkan sebelumnya. *Card* di sebelah kanan berfungsi untuk menampilkan hasil pelatihan ACO-BPNN beserta komponen lengkap BPNN. Perancangan antarmuka tersebut ditunjukkan pada Gambar 4.18.



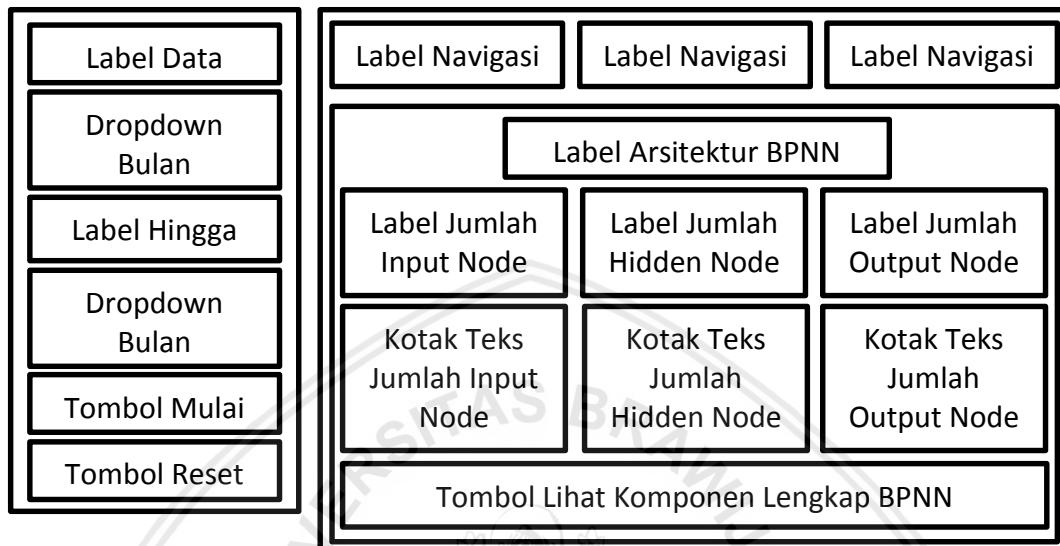
**Gambar 4.18 Perancangan antarmuka *tab menu hasil pelatihan aktif***

#### 4.4.3 Perancangan Antarmuka Halaman Pengujian ACO-BPNN

Fungsi halaman ini adalah menerima masukan berupa data uji untuk proses pengujian. Selain itu, halaman ini menampilkan informasi pada saat sebelum pengujian, hasil pengujian, serta detail harga dan grafik pengujian. Salah satu dari ketiga informasi tersebut ditampilkan ketika *tab menu* prapengujian, hasil pengujian, atau detail harga dan grafik sedang aktif. Sehingga ketika salah satu *tab menu* sedang aktif, maka *tab menu* lainnya tidak aktif.

#### 4.4.3.1 Tab Menu Prapengujian Aktif

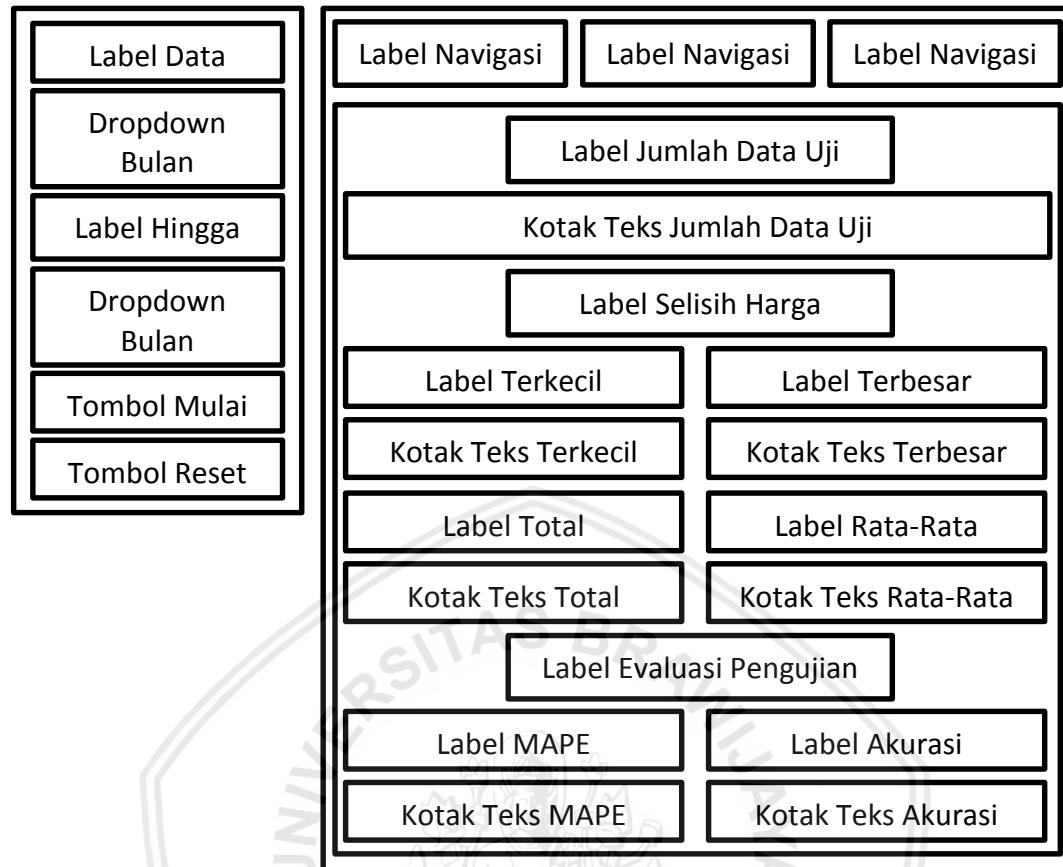
Ketika *tab menu* ini aktif, maka halaman akan menampilkan 2 *card*. *Card* di sebelah kiri berfungsi untuk menerima masukan data uji. *Card* di sebelah kanan berfungsi untuk menampilkan arsitektur BPNN yang akan digunakan untuk proses pengujian. Perancangan antarmuka ditunjukkan pada Gambar 4.19.



Gambar 4.19 Perancangan antarmuka *tab menu* prapengujian aktif

#### 4.4.3.2 Tab Menu Hasil Pengujian Aktif

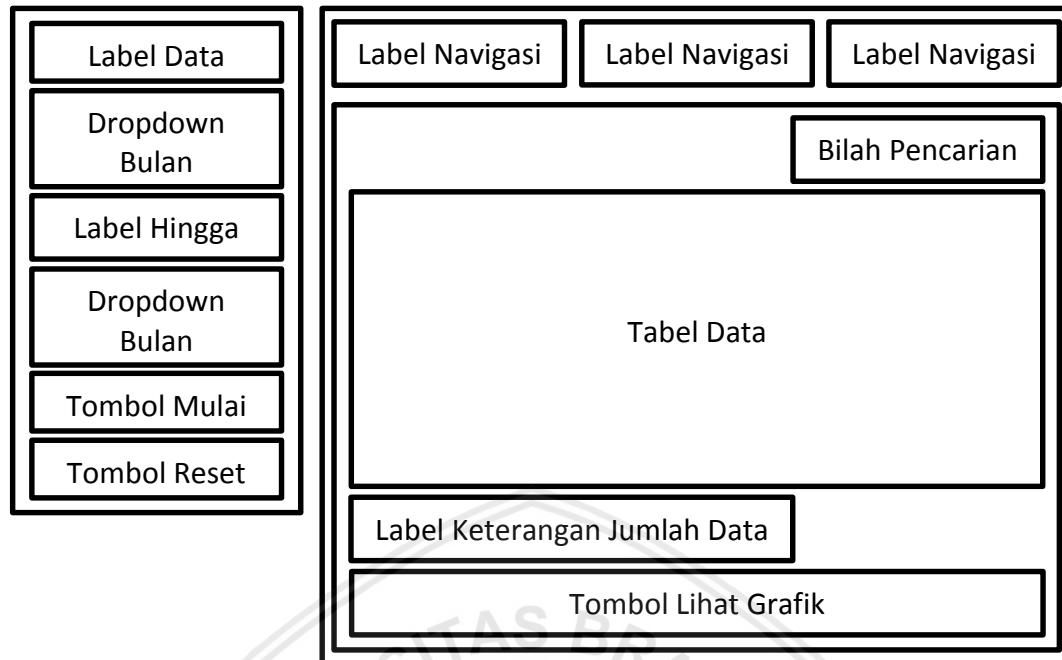
Ketika *tab menu* ini aktif, maka halaman akan menampilkan 2 *card*. *Card* di sebelah kiri berfungsi untuk menampilkan interval bulan data uji yang telah dimasukkan sebelumnya. *Card* di sebelah kanan berfungsi untuk menampilkan hasil pengujian. Perancangan antarmuka ditunjukkan pada Gambar 4.20.



Gambar 4.20 Perancangan antarmuka *tab menu hasil pengujian aktif*

#### 4.4.3.3 Tab Menu Detail Harga dan Grafik

Ketika *tab menu* ini aktif, maka halaman akan menampilkan 2 *card*. *Card* di sebelah kiri berfungsi untuk menampilkan interval bulan data uji yang telah dimasukkan sebelumnya. *Card* di sebelah kanan berfungsi untuk menampilkan detail harga prediksi berdasarkan harga aktual beserta grafiknya. Perancangan antarmuka ditunjukkan pada Gambar 4.21.



**Gambar 4.21 Perancangan antarmuka *tab menu detail harga dan grafik aktif***

## 4.5 Perancangan Pengujian

Perancangan skenario pengujian diperlukan untuk menghasilkan MAPE terbaik dari implementasi ACO-BPNN dalam memprediksi harga saham. Skenario tersebut berisi pengujian terhadap nilai parameter ACO seperti:

1. Tetapan siklus semut ( $q_0$ )
2. Tetapan pengendali intensitas feromon ( $\alpha$ )
3. Tetapan pengendali visibilitas ( $\beta$ )
4. Tetapan penguapan feromon lokal ( $\rho$ )
5. Tetapan penguapan feromon global ( $\delta$ )
6. Jumlah semut ( $m$ )
7. Jumlah iterasi ( $NCmax$ )

### 4.5.1 Perancangan Skenario Pengujian Nilai Tetapan Siklus Semut

Pengujian ini bertujuan untuk memperoleh nilai ( $q_0$ ) yang menghasilkan MAPE terkecil. Uji coba dilakukan dengan memberikan nilai berbeda-beda pada ( $q_0$ ) dalam intervalnya. Setiap nilai diuji 4 kali, kemudian diambil rata-rata MAPE-nya. Rancangan pengujian tersebut ditunjukkan pada Tabel 4.37.

Nilai $q_0$	Nilai MAPE Percobaan ke-i				Rata-Rata MAPE
	1	2	3	4	
0,1					
0,2					
0,3					
0,4					
0,5					
0,6					
0,7					
0,8					
0,9					

#### 4.5.2 Perancangan Skenario Pengujian Nilai Tetapan Pengendali Intensitas Feromon

Pengujian ini bertujuan untuk memperoleh nilai ( $\alpha$ ) yang menghasilkan MAPE terkecil. Uji coba dilakukan dengan memberikan nilai berbeda-beda pada ( $\alpha$ ) dalam intervalnya. Setiap nilai diuji 4 kali, kemudian diambil rata-rata MAPE-nya. Rancangan pengujian tersebut ditunjukkan pada Tabel 4.38.

Tabel 4.38 Rancangan pengujian nilai tetapan pengendali intensitas feromon

Nilai $\alpha$	Nilai MAPE Percobaan ke-i				Rata-Rata MAPE
	1	2	3	4	
0,1					
0,2					
0,3					
0,4					
0,5					
0,6					
0,7					
0,8					
0,9					

#### 4.5.3 Perancangan Skenario Pengujian Nilai Tetapan Pengendali Visibilitas

Pengujian ini bertujuan untuk memperoleh nilai ( $\beta$ ) yang menghasilkan MAPE terkecil. Uji coba dilakukan dengan memberikan nilai berbeda-beda pada ( $\beta$ ) dalam intervalnya. Setiap nilai diuji 4 kali, kemudian diambil rata-rata MAPE-nya. Rancangan pengujian tersebut ditunjukkan pada Tabel 4.39.

**Tabel 4.39 Rancangan pengujian nilai tetapan pengendali visibilitas**

Nilai $\beta$	Nilai MAPE Percobaan ke-i				Rata-Rata MAPE
	1	2	3	4	
0,1					
0,2					
0,3					
0,4					
0,5					
0,6					
0,7					
0,8					
0,9					

#### 4.5.4 Perancangan Skenario Pengujian Nilai Tetapan Penguapan Feromon Lokal

Pengujian ini bertujuan untuk memperoleh nilai ( $\rho$ ) yang menghasilkan MAPE terkecil. Uji coba dilakukan dengan memberikan nilai berbeda-beda pada ( $\rho$ ) dalam intervalnya. Setiap nilai diuji 4 kali, kemudian diambil rata-rata MAPE-nya. Rancangan pengujian tersebut ditunjukkan pada Tabel 4.40.

**Tabel 4.40 Rancangan pengujian nilai tetapan penguapan feromon lokal**

Nilai $\rho$	Nilai MAPE Percobaan ke-i				Rata-Rata MAPE
	1	2	3	4	
0,1					
0,2					
0,3					
0,4					
0,5					

0,6					
0,7					
0,8					
0,9					

#### 4.5.5 Perancangan Skenario Pengujian Nilai Tetapan Penguapan Feromon Global

Pengujian ini bertujuan untuk memperoleh nilai ( $\delta$ ) yang menghasilkan MAPE terkecil. Uji coba dilakukan dengan memberikan nilai berbeda-beda pada ( $\delta$ ) dalam intervalnya. Setiap nilai diuji 4 kali, kemudian diambil rata-rata MAPE-nya. Rancangan pengujian tersebut ditunjukkan pada Tabel 4.41.

**Tabel 4.41 Rancangan pengujian nilai tetapan penguapan feromon global**

Nilai $\delta$	Nilai MAPE Percobaan ke-i				Rata-Rata MAPE
	1	2	3	4	
0,1					
0,2					
0,3					
0,4					
0,5					
0,6					
0,7					
0,8					
0,9					

#### 4.5.6 Perancangan Skenario Pengujian Jumlah Semut ACO

Pengujian ini bertujuan untuk memperoleh nilai ( $m$ ) yang menghasilkan MAPE terkecil. Uji coba dilakukan dengan memberikan nilai berbeda-beda pada ( $m$ ) dalam intervalnya. Setiap nilai diuji 4 kali, kemudian diambil rata-rata MAPE-nya. Rancangan pengujian tersebut ditunjukkan pada Tabel 4.42.

**Tabel 4.42 Rancangan pengujian jumlah semut ACO**

Nilai $m$	Nilai MAPE Percobaan ke-i				Rata-Rata MAPE
	1	2	3	4	
3					
5					
7					
9					
11					

#### 4.5.7 Perancangan Skenario Pengujian Jumlah Iterasi ACO

Pengujian ini bertujuan untuk memperoleh nilai ( $NCmax$ ) yang menghasilkan MAPE terkecil. Uji coba dilakukan dengan memberikan nilai berbeda-beda pada ( $NCmax$ ) dalam intervalnya. Setiap nilai diuji 4 kali, kemudian diambil rata-rata MAPE-nya. Rancangan pengujian tersebut ditunjukkan pada Tabel 4.43.

**Tabel 4.43 Rancangan pengujian jumlah iterasi ACO**

Nilai $NCmax$	Nilai MAPE Percobaan ke-i				Rata-Rata MAPE
	1	2	3	4	
3					
4					
5					
6					
7					

#### 4.5.8 Perancangan Skenario Pengujian ACO-BPNN dan BPNN

Pengujian ini bertujuan untuk mengetahui pengaruh ACO dalam menurunkan MAPE BPNN. Uji coba dilakukan dengan memprediksi harga saham pada 4 bulan yang berbeda. Setiap bulan diuji 7 kali, kemudian diambil rata-rata MAPE-nya. Sehingga BPNN secara total diuji 28 kali, lalu diambil rata-rata MAPE keseluruhan. Kemudian MAPE ACO-BPNN diambil dari hasil terbaik pada pengujian jumlah iterasi ACO. Rancangan pengujian tersebut ditunjukkan pada Tabel 4.44.

**Tabel 4.44 Rancangan pengujian ACO-BPNN dan BPNN**

Percobaan ke-i	Nilai MAPE Percobaan ke-j				Rata-Rata MAPE	MAPE ACO-BPNN
	1	2	3	4		
1						
2						
3						
4						
5						
6						
7						
Rata-Rata Seluruh Percobaan						

## BAB 5 IMPLEMENTASI

Bagian ini membahas tentang implementasi sistem berdasarkan perancangan sebelumnya. Implementasi sistem berisi mengenai implementasi algoritma dan implementasi antarmuka untuk menghasilkan keluaran yang akan diuji dan dianalisis di bab selanjutnya.

### 5.1 Implementasi Algoritma

Algoritma ACO-BPNN yang telah dirancang, selanjutnya diimplementasikan ke dalam bahasa pemrograman. Proses implementasi dijelaskan dalam dua bagian, yaitu implementasi algoritma ACO dan implementasi algoritma BPNN. Setiap algoritma memiliki beberapa proses yang akan diuraikan fungsi dan kode programnya. Bahasa pemrograman yang dipakai adalah PHP dengan *text editor* Sublime Text dan Apache dari XAMPP sebagai servernya.

#### 5.1.1 Implementasi Algoritma *Ant Colony Optimization*

##### 5.1.1.1 Menentukan *Node* Awal

Algoritma ini berfungsi untuk meletakkan semua semut di *node* 10, yaitu titik awal bagi semut sebelum memulai perjalannya. Implementasi algoritma tersebut ke dalam sebuah fungsi ditunjukkan oleh Kode Program 5.1.

Algoritma 1: Fungsi Setupants

```

1 private function setupAnts() {
2     $city = 0;
3     for ($ant = 0; $ant < $this->numberofants; $ant++) {
4         $this->routes[$ant][$city] = 10;
5     }
6 }
```

Kode Program 5.1 Menentukan *node* awal

Keterangan Kode Program 5.1:

1. Baris 1 deklarasi fungsi tanpa parameter
2. Baris 2 deklarasi dan inisialisasi variabel lokal
3. Baris 3-5 perulangan sebanyak jumlah semut untuk meletakkan semua semut di *node* 10

##### 5.1.1.2 Menentukan *next node*

Algoritma ini berfungsi untuk memilih *node* selanjutnya yang akan dikunjungi oleh semut. Implementasi algoritma tersebut ke dalam sebuah fungsi ditunjukkan oleh Kode Program 5.2.

Algoritma 2: Fungsi Nextcity

```

1 private function nextCity($city) {
2     $q = $this->getRandomValue();
3     $probability = array();
```

```

4      $sumofprobability = 0;
5      $cumulativeprobability = 0;
6      $maxpossibility = 0;
7      if ($q <= $this->q0) {
8          for ($citydestination = 0; $citydestination < $this-
>numberofcities; $citydestination++) {
9              $probability[$citydestination] = pow($this-
>trails[$city][$citydestination], $this->alpha) * pow($this-
>trails[$city][$citydestination], $this->beta);
10         }
11         if (count(array_unique($probability)) == 1) {
12             return $this->getRandomCity();
13         }
14         $maxpossibility = max($probability);
15         $maxpossibility = array_search($maxpossibility,
$probability);
16         return $maxpossibility;
17     }
18     else {
19         for ($citydestination = 0; $citydestination < $this-
>numberofcities; $citydestination++) {
20             $probability[$citydestination] = pow($this-
>trails[$city][$citydestination], $this->alpha) * pow($this-
>trails[$city][$citydestination], $this->beta);
21             $sumofprobability +=
$probability[$citydestination];
22         }
23         for ($citydestination = 0; $citydestination < $this-
>numberofcities; $citydestination++) {
24             $cumulativeprobability +=
$probability[$citydestination] / $sumofprobability;
25             $probability[$citydestination] =
$cumulativeprobability;
26         }
27         $limit = $this->getRandomValue();
28         for ($citydestination = 0; $citydestination < $this-
>numberofcities; $citydestination++) {
29             if ($limit <= $probability[$citydestination]) {
30                 $maxpossibility = $citydestination;
31                 break;
32             }
33         }
34         return $maxpossibility;
35     }
36 }

```

### Kode Program 5.2 Menentukan *next node*

Keterangan Kode Program 5.2:

1. Baris 1 deklarasi fungsi beserta parameternya
2. Baris 2-6 deklarasi dan inisialisasi variabel lokal
3. Baris 7-17 seleksi kondisi ketika  $q \leq q0$ , maka melakukan pemilihan *node* selanjutnya dengan ArgMax. Ketika semua nilai probabilitas menuju *node* selanjutnya sama, maka pemilihan dilakukan secara acak

4. Baris 18-35 seleksi kondisi ketika  $q > q_0$ , maka melakukan pemilihan *node* selanjutnya dengan *Roulette Wheel*
5. Baris 36 mengembalikan nilai *node* selanjutnya

#### 5.1.1.3 Update Feromon Lokal

Algoritma ini berfungsi untuk memperbarui feromon lokal pada rute yang dilewati semut. Implementasi algoritma tersebut ke dalam sebuah fungsi ditunjukkan oleh Kode Program 5.3.

Algoritma 3: Fungsi Localpheromoneupdate

```

1 private function localPheromoneUpdate($city,
2   $citydestination) {
3     $this->trails[$city][$citydestination] = (1 - $this-
>rho) * $this->trails[$city][$citydestination] + ($this-
>rho * $this->tau0);
4 }
```

#### Kode Program 5.3 Update feromon lokal

Keterangan Kode Program 5.3:

1. Baris 1 deklarasi fungsi beserta parameternya
2. Baris 2 melakukan perhitungan *update* feromon lokal dengan Persamaan 2.21

#### 5.1.1.4 Konversi Nilai Parameter BPNN

Algoritma ini berfungsi untuk melakukan konversi terhadap kumpulan *node* kunjungan semut menjadi setiap parameter BPNN. Implementasi algoritma tersebut ke dalam sebuah fungsi ditunjukkan oleh Kode Program 5.4.

Algoritma 4: Fungsi Parameterconversion

```

1 private function parameterConversion() {
2   for ($ant = 0; $ant < $this->numberofants; $ant++) {
3     $parameter = 0;
4     for ($route = 1; $route <= $this->numberofnodes - 1;
5       $route += 3) {
6       if ($parameter < 2) {
7         $this->parameters[$ant][$parameter] =
8           ((($this->routess[$ant][$route] * 100) + ($this-
>routess[$ant][$route + 1] * 10) + ($this-
>routess[$ant][$route + 2])) / 1000;
9         if ($this->parameters[$ant][$parameter]
10          == 0) {
11           $this->parameters[$ant][$parameter]
12           = 0.001;
13           }
14           $this->parameters[$ant][$parameter] =
15             (((($this->parameters[$ant][$parameter] - 0,001) / (0,999 -
16               0,001)) * (0,999 - 0,001)) + 0,001;
17             $parameter++;
18           }
19           else {
20             $this->parameters[$ant][$parameter] =
21               ($this->routess[$ant][$route] * 10) + ($this-
>routess[$ant][$route + 1]);
22           }
23         }
24       }
25     }
26   }
27 }
```

```

15           for ($value = 10; $value <= 100; $value
16             += 5) {
17               if ($this-
18 >parameters[$ant][$parameter] <= $value) {
19                 $this-
20 >parameters[$ant][$parameter] = $value / 5;
21                 break;
22               }
23             }
24           }

```

#### Kode Program 5.4 Konversi nilai parameter BPNN

Keterangan Kode Program 5.4:

1. Baris 1 deklarasi fungsi beserta parameternya
2. Baris 2-23 perulangan sebanyak jumlah semut
3. Baris 3 deklarasi dan inisialisasi variabel lokal
4. Baris 4-22 perulangan sebanyak *node* kunjungan semut dengan *increment* bernilai 3
5. Baris 5-12 seleksi kondisi ketika nilai *counter* kurang dari 2, maka melakukan konversi parameter *learning rate* dan momentum
6. Baris 13-21 seleksi kondisi ketika kondisi sebelumnya tidak terpenuhi, maka melakukan konversi parameter jumlah *hidden node*

##### 5.1.1.5 Update Feromon Global

Algoritma ini berfungsi untuk memperbarui feromon global setelah semua semut menyelesaikan perjalanannya. Implementasi algoritma tersebut ke dalam sebuah fungsi ditunjukkan oleh Kode Program 5.5.

##### Algoritma 5: Fungsi Globalpheromoneupdate

```

1 private function globalPheromoneUpdate($iteration) {
2   $this->globalbest[$iteration][0] = min($this-
3   >antofRMSE);
4   $bestant = array_search($this-
5   >globalbest[$iteration][0], $this->antofRMSE);
6   $this->globalbest[$iteration][1] = $this-
7   >antofBPNNcomponent[$bestant];
8   $numberofparameters = count($this->parameters[0]);
9   for ($parameter = 0; $parameter < $numberofparameters;
$parameter++) {
10     $this->globalbest[$iteration][$parameter + 2] =
11     $this->parameters[$bestant][$parameter];
12   }
13   for ($route = 0; $route < $this->numberofnodes; $route
14  ++) {
15     $deltatau = 1 / $this->antofRMSE[$bestant];
16     $this->trails[$this->routess[$bestant][$route]][$this-
17 >routess[$bestant][$route + 1]] = (1 - $this->delta) *
18     $this->trails[$this->routess[$bestant][$route]][$this-
19 >routess[$bestant][$route + 1]];
20   }
21 }

```

```

12 >routes[$bestant][$route + 1]] + ($this->delta *
13 $deltatau);
14     }
15     for ($city = 0; $city < $this->numberofcities + 1; $city
16     ++
17     {
18         for ($citydestination = 0; $citydestination < $this-
19             >numberofcities; $citydestination++)
20             {
21                 $this->trails[$city][$citydestination] = (1 -
22 $this->delta) * $this->trails[$city][$citydestination] + 0;
23             }
24     }
25 }

```

### Kode Program 5.5 *Update feromon global*

Keterangan Kode Program 5.5:

1. Baris 1 deklarasi fungsi beserta parameternya
2. Baris 2 inisialisasi variabel global dengan nilai RMSE terkecil
3. Baris 3 deklatasi dan inisialisasi variabel lokal dengan indeks dari *global best*
4. Baris 4 inisialisasi variabel global dengan array berisi komponen BPNN *global best*
5. Baris 5 deklarasi dan inisialisasi variabel lokal dengan jumlah parameter BPNN *global best*
6. Baris 6-8 perulangan sebanyak jumlah parameter untuk menyimpan *global best*
7. Baris 9-12 perulangan sebanyak kumpulan *node* kunjungan semut untuk memperbarui feromon *global best*
8. Baris 13-17 *nested loop* sebanyak jumlah *node* untuk memperbarui feromon rute selain *global best*

#### 5.1.1.6 Menyimpan *Global Best ACO*

Algoritma ini berfungsi untuk menyimpan parameter dan komponen BPNN milik *global best ACO* terakhir. Implementasi algoritma tersebut ke dalam sebuah fungsi ditunjukkan oleh Kode Program 5.6.

Algoritma 6: Fungsi *Savelastglobalbest*

```

1 private function saveLastGlobalBest() {
2     $iterationlength = count($this->globalbest);
3     $arraylength = count($this->globalbest[0]);
4     for ($parameter = 0; $parameter < $arraylength;
5         $parameter++)
6         $this->ACObest[$parameter] = $this-
7         >globalbest[$iterationlength - 1][$parameter];
8     }
9     $bpnnclass = new BPNN(4, $this->ACObest[4], 1);
10    $bpnnclass->import($this->ACObest[1]);

```

9	\$bpnnclass->save('BPNN component (ACO-BPNN).txt');
10	}

#### Kode Program 5.6 Menyimpan *global best* ACO terakhir

Keterangan Kode Program 5.6:

1. Baris 1 deklarasi fungsi tanpa parameter
2. Baris 2-3 deklarasi dan inisialisasi variabel lokal untuk mengetahui jumlah iterasi ACO dan panjang suatu *array*
3. Baris 4-6 perulangan sebanyak jumlah parameter untuk menyimpan *global best* ACO terakhir
4. Baris 8 instansiasi objek beserta argumennya
5. Baris 9-10 menyimpan komponen BPNN dalam sebuah file berformat .txt

#### 5.1.2 Implementasi Algoritma *Backpropagation Neural Network*

##### 5.1.2.1 Min-Max Normalization

Algoritma ini berfungsi untuk menyelaraskan interval data dengan interval yang dimiliki jaringan. Implementasi algoritma tersebut ke dalam sebuah fungsi ditunjukkan oleh Kode Program 5.7.

##### Algoritma 7: Fungsi Normalization

1	private function normalization(\$value) {
2	return (\$value - \$this->mindatavalue) / (\$this-
3	>maxdatavalue - \$this->mindatavalue);

#### Kode Program 5.7 Min-max normalization

Keterangan Kode Program 5.7:

1. Baris 1 deklarasi fungsi beserta parameternya
2. Baris 2 mengembalikan nilai hasil perhitungan normalisasi menggunakan Persamaan 2.24

##### 5.1.2.2 Proses Feedforward

Algoritma ini berfungsi untuk menghasilkan keluaran pada *output layer*. Implementasi algoritma tersebut ke dalam sebuah fungsi ditunjukkan oleh Kode Program 5.8.

##### Algoritma 8: Fungsi Feedforward

1	private function feedforward(\$input) {
2	foreach (\$input as \$index => \$value) {
3	\$this->nodeValue[0][\$index] = \$value;
4	}
5	for (\$layer = 1; \$layer < \$this->layerCount; \$layer ++)
6	{
7	\$prev_layer = \$layer -1;

```

8          $node_value = 0.0;
9          if ($layer != 0 && $layer != $this->layerCount
10         - 1 && $node == 0) {
11              $this->nodeValue[$layer][$node] = 1;
12              continue;
13          }
14          for ($prev_node = 0; $prev_node < ($this-
15             >nodeCount[$prev_layer]); $prev_node++) {
16              $inputnode_value = $this-
17             >nodeValue[$prev_layer][$prev_node];
18              $edge_weight = $this-
19             >connectorWeight[$prev_layer][$prev_node][$node];
20              $node_value = $node_value +
21              ($inputnode_value * $edge_weight);
22          }
23          $node_value = $node_value - $this-
24          >nodeThreshold[$layer][$node];
25          $node_value = $this->activation($node_value);
26          $this->nodeValue[$layer][$node] = $node_value;
27      }
28  }
29  return $this->nodeValue[$this->layerCount - 1];
30 }
```

### Kode Program 5.8 Proses feedforward

Keterangan Kode Program 5.8:

1. Baris 1 deklarasi fungsi beserta parameternya
2. Baris 2-4 perulangan sebanyak panjang suatu array untuk mengisi nilai *node input*
3. Baris 5-22 proses propagasi maju untuk menghasilkan keluaran pada *hidden layer* dan *output layer*
4. Baris 23 mengembalikan nilai suatu array

#### 5.1.2.3 Proses Backpropagate

Algoritma ini berfungsi untuk mengoreksi bobot dan *threshold* pada jaringan. Implementasi algoritma tersebut ke dalam sebuah fungsi ditunjukkan oleh Kode Program 5.9.

Algoritma 9: Fungsi Backpropagate

```

1  private function backpropagate($output, $desired_output) {
2      $errorgradient = array ();
3      $outputlayer = $this->layerCount - 1;
4      $learning_rate = $this->getLearningRate();
5      $momentum = $this->getMomentum();
6      for ($layer = $this->layerCount - 1; $layer > 0; $layer
7      --) {
8          for ($node = 0; $node < $this->nodeCount[$layer];
9          $node++) {
10              if ($layer == $outputlayer) {
11                  $error = $desired_output[$node] -
12                  $output[$node];
13                  $errorgradient[$layer][$node] = $this-
14                  >derivativeActivation($output[$node]) * $error;
15              }
16          }
17      }
18  }
```

```

11         } else {
12             $next_layer = $layer +1;
13             $productsum = 0;
14             for ($next_index = 0; $next_index <
15                 ($this->nodeCount[$next_layer]); $next_index++) {
16                 $errorgradient =
17                     $errorgradient[$next_layer][$next_index];
18                     $edgeWeight = $this-
19 >connectorWeight[$layer][$node][$next_index];
20                     $productsum = $productsum +
21                     $errorgradient * $edgeWeight;
22                     }
23                     $nodeValue = $this-
24 >nodeValue[$layer][$node];
25                     $errorgradient[$layer][$node] = $this-
26 >derivativeActivation($nodeValue) * $productsum;
27                     }
28                     $prev_layer = $layer -1;
29                     for ($prev_index = 0; $prev_index < ($this-
30 >nodeCount[$prev_layer]); $prev_index++) {
31                         $nodeValue = $this-
32 >nodeValue[$prev_layer][$prev_index];
33                         $connectorWeight = $this-
34 >connectorWeight[$prev_layer][$prev_index][$node];
35                         $weight_correction = $learning_rate *
36 $nodeValue * $errorgradient[$layer][$node];
37                         $prev_weightcorrection = $this-
38 >previousWeightCorrection[$prev_layer][$prev_index][$node];
39                         $new_weight = $connectorWeight +
40 $weight_correction + $momentum * $prev_weightcorrection;
41                         $this-
42 >connectorWeight[$prev_layer][$prev_index][$node] =
43 $new_weight;
44                         $this-
45 >previousWeightCorrection[$prev_layer][$prev_index][$node]
46 = $weight_correction;
47                         }
48                         $threshold_correction = $learning_rate * -1 *
49 $errorgradient[$layer][$node];
50                         $new_threshold = $this-
51 >nodeThreshold[$layer][$node] + $threshold_correction;
52                         $this->nodeThreshold[$layer][$node] =
53 $new_threshold;
54                         }
55                     }
56                 }
57             }

```

### Kode Program 5.9 Proses *backpropagate*

Keterangan Kode Program 5.9:

1. Baris 1 deklarasi fungsi beserta parameternya
2. Baris 2 deklarasi dan inisialisasi variabel lokal
3. Baris 3-5 deklarasi dan inisialisasi variabel lokal dengan variabel global
4. Baris 6-36 perulangan sebanyak jumlah lapisan jaringan minus satu
5. Baris 7-35 perulangan sebanyak jumlah *node* pada suatu lapisan jaringan

6. Baris 8-11 seleksi kondisi untuk menghitung faktor kesalahan unit keluaran
7. Baris 11-21 seleksi kondisi untuk menghitung faktor kesalahan unit tersembunyi
8. Baris 22 deklarasi dan inisialisasi variabel lokal dengan variabel lokal
9. Baris 23-31 perulangan sejumlah *node* pada suatu lapisan jaringan untuk mengoreksi dan memperbarui bobot
10. Baris 32 menghitung koreksi *threshold*
11. Baris 33-34 memperbarui *threshold*

#### **5.1.2.4 Menghitung Root Mean Square Error**

Algoritma ini berfungsi untuk menghitung RMSE pada fase pelatihan jaringan. Implementasi algoritma tersebut ke dalam sebuah fungsi ditunjukkan oleh Kode Program 5.10.

Algoritma 10: Fungsi Rootmeansquarederror	
1	private function rootMeanSquaredError() {
2	\$sumofsquarederror = 0.0;
3	\$amountofdata = count(\$this->trainInputs);
4	for (\$i = 0; \$i < \$amountofdata; \$i++) {
5	\$sumofsquarederror += \$this->squaredError(\$this->trainInputs[\$i], \$this->trainOutput[\$i]);
6	}
7	\$rmse = sqrt(\$sumofsquarederror / count(\$this->trainInputs));
8	return \$rmse;
9	}

#### **Kode Program 5.10 Menghitung root mean square error**

Keterangan Kode Program 5.10:

1. Baris 1 deklarasi fungsi tanpa parameter
2. Baris 2 deklarasi dan inisialisasi variabel lokal
3. Baris 3 deklarasi dan inisialisasi variabel lokal dengan panjang suatu *array*
4. Baris 4-6 perulangan sebanyak jumlah data untuk menghitung *sum of squared error*
5. Baris 7 menghitung RMSE
6. Baris 8 mengembalikan nilai RMSE

#### **5.1.2.5 Min-Max Denormalization**

Algoritma ini berfungsi untuk mengembalikan interval data dan hasil prediksi kepada interval sebenarnya. Implementasi algoritma tersebut ke dalam sebuah fungsi ditunjukkan oleh Kode Program 5.11.

Algoritma 11: Fungsi Denormalization

```

1  private function denormalization($value) {
2      return ($value * ($this->maxdatavalue - $this-
3          >mindatavalue)) + $this->mindatavalue;
}

```

**Kode Program 5.11 Min-max denormalization**

Keterangan Kode Program 5.11:

1. Baris 1 deklarasi fungsi beserta parameternya
2. Baris 2 mengembalikan nilai hasil perhitungan denormalisasi menggunakan Persamaan 2.25

#### 5.1.2.6 Menghitung Mean Absolute Percentage Error

Algoritma ini berfungsi untuk menghitung MAPE pada fase pengujian jaringan. Implementasi algoritma tersebut ke dalam sebuah fungsi ditunjukkan oleh Kode Program 5.12.

Algoritma 12: Fungsi Meanabsolutepercentageerror

```

1  public function meanAbsolutePercentageError() {
2      $sumofabsoluteerror = 0.0;
3      $amountofdata = count($this->testInputs);
4      for ($i = 0; $i < $amountofdata; $i++) {
5          $sumofabsoluteerror += $this->absoluteError($this-
6              >testInputs[$i], $this->testOutput[$i]);
7      }
8      $mape = ($sumofabsoluteerror / $amountofdata) * 100;
9      return $mape;
}

```

**Kode Program 5.12 Menghitung mean absolute percentage error**

Keterangan Kode Program 5.12:

1. Baris 1 deklarasi fungsi tanpa parameter
2. Baris 2 deklarasi dan inisialisasi variabel lokal
3. Baris 3 deklarasi dan inisialisasi variabel lokal dengan panjang suatu array
4. Baris 4-6 perulangan sebanyak jumlah data untuk menghitung *sum of absolute error*
5. Baris 7 menghitung MAPE
6. Baris 8 mengembalikan nilai MAPE

## 5.2 Implementasi Antarmuka

Perancangan antarmuka pada bab sebelumnya menjadi acuan dalam implementasi antarmuka halaman data mentah, halaman pelatihan ACO-BPNN, dan halaman pengujian ACO-BPNN. Diberikan screenshot dari setiap implementasi halaman tersebut pada sistem yang telah dikembangkan.

### 5.2.1 Implementasi Antarmuka Halaman Data Mentah

Implementasi antarmuka halaman ini ditunjukkan pada Gambar 5.1.

No	Date	Open	High	Low	Close	Adj Close	Volume
1	1/1/2018	3640	3640	3640	3640	3532.066895	0
2	1/2/2018	3690	3720	3610	3630	3522.363525	91127100
3	1/3/2018	3630	3630	3560	3610	3502.956543	70148700
4	1/4/2018	3610	3610	3460	3550	3444.735596	156406300
5	1/5/2018	3550	3590	3520	3590	3483.549561	87095600
6	1/8/2018	3590	3590	3560	3590	3483.549561	54559100
7	1/9/2018	3600	3600	3530	3570	3464.142578	90606700
8	1/10/2018	3580	3590	3500	3530	3425.328613	116409100

Gambar 5.1 Implementasi antarmuka halaman data mentah

### 5.2.2 Implementasi Antarmuka Halaman Pelatihan ACO-BPNN

Tampilan antarmuka halaman ini dibagi menjadi 3 bagian, yaitu saat *tab menu* prapelatihan, proses pelatihan, atau hasil pelatihan sedang aktif.

#### 5.2.2.1 Tab Menu Prapelatihan Aktif

Implementasi antarmuka ketika *tab menu* tersebut aktif ditunjukkan pada Gambar 5.2.

Gambar 5.2 Implementasi antarmuka *tab menu* prapelatihan aktif

### 5.2.2.2 Tab Menu Proses Pelatihan Aktif

Implementasi antarmuka ketika *tab menu* tersebut aktif ditunjukkan pada Gambar 5.3.

The screenshot shows a software interface for training a BPNN. On the left, there's a sidebar with 'PARAMETER ACO' and 'DATA' sections. The main area has three tabs: 'Prapelatihan', 'Proses Pelatihan' (which is active), and 'Hasil Pelatihan'. The 'Proses Pelatihan' tab displays a table of training iterations (1 to 11) with columns for Iterasi, RMSE, Learning Rate, Momentum, and Jumlah Hidden Node. Below the table is a search bar and a note: 'Showing 1 to 21 of 21 entries'.

Iterasi	RMSE	Learning Rate	Momentum	Jumlah Hidden Node
1	0.012	0.345	0.678	9
2	0.012	0.345	0.678	9
3	0.012	0.345	0.678	9
4	0.012	0.345	0.678	9
5	0.012	0.345	0.678	9
6	0.012	0.345	0.678	9
7	0.012	0.345	0.678	9
8	0.012	0.345	0.678	9
9	0.012	0.345	0.678	9
10	0.012	0.345	0.678	9
11	0.012	0.345	0.678	9

Gambar 5.3 Implementasi antarmuka *tab menu* proses pelatihan aktif

### 5.2.2.3 Tab Menu Hasil Pelatihan Aktif

Implementasi antarmuka ketika *tab menu* tersebut aktif ditunjukkan pada Gambar 5.4.

The screenshot shows the 'Hasil Pelatihan' tab active. It displays various performance metrics and optimal parameters for the BPNN. On the left is a sidebar with 'PARAMETER ACO' and 'DATA' sections. The central area includes sections for 'JUMLAH DATA LATIH', 'RMSE', 'PARAMETER BPNN OPTIMAL' (with sliders for Learning Rate, Momentum, and Jumlah Hidden Node), and a button to 'Lihat komponen Lengkap BPNN'.

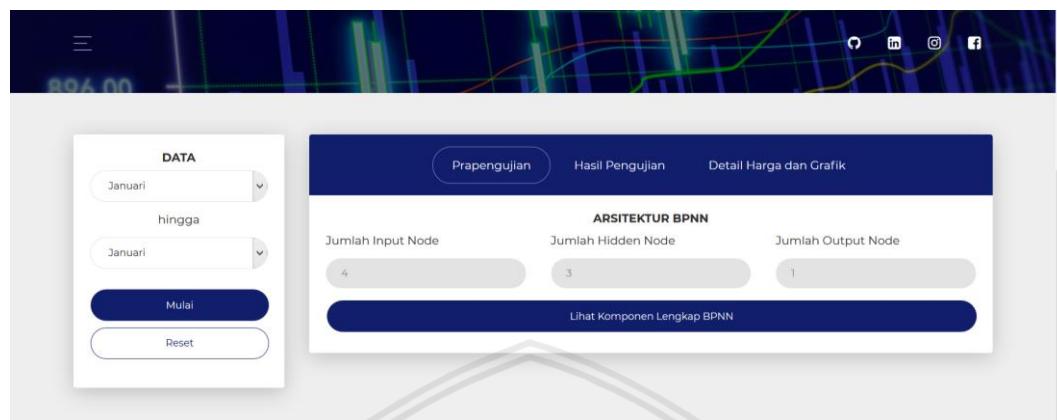
Gambar 5.4 Implementasi antarmuka *tab menu* hasil pelatihan aktif

### 5.2.3 Implementasi Antarmuka Halaman Pengujian ACO-BPNN

Tampilan antarmuka halaman ini dibagi menjadi 3 bagian, yaitu saat *tab menu* prapengujian, hasil pengujian, atau detail harga dan grafik sedang aktif.

### 5.2.3.1 Tab Menu Prapengujian Aktif

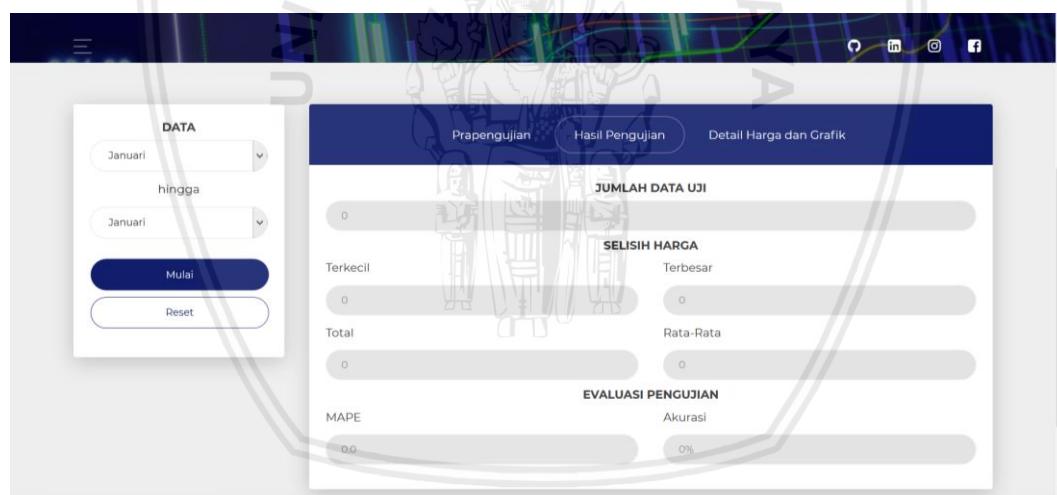
Implementasi antarmuka ketika *tab menu* tersebut aktif ditunjukkan pada Gambar 5.5.



Gambar 5.5 Implementasi antarmuka *tab menu* prapengujian aktif

### 5.2.3.2 Tab Menu Hasil Pengujian Aktif

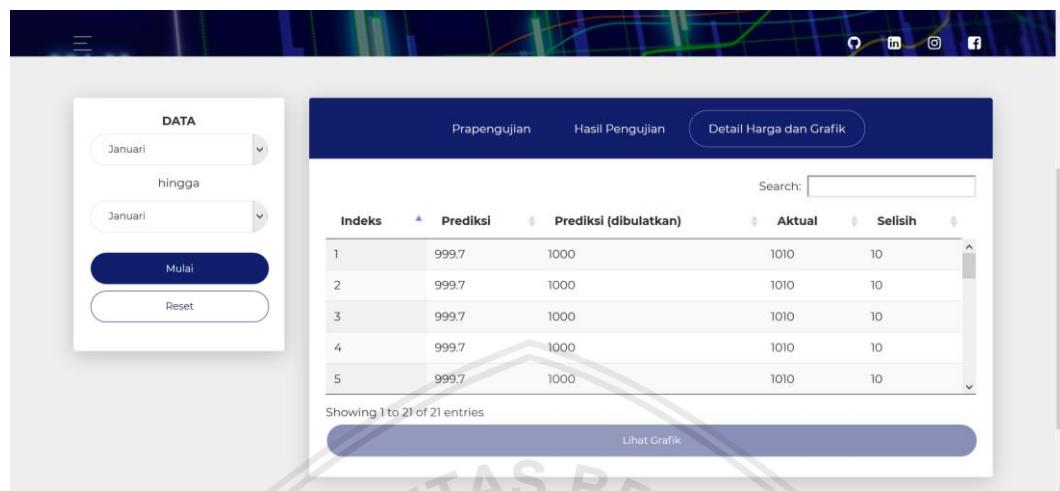
Implementasi antarmuka ketika *tab menu* tersebut aktif ditunjukkan pada Gambar 5.6.



Gambar 5.6 Implementasi antarmuka *tab menu* hasil pengujian aktif

### 5.2.3.3 Tab Menu Detail Harga dan Grafik Aktif

Implementasi antarmuka ketika *tab menu* tersebut aktif ditunjukkan pada Gambar 5.7.



Gambar 5.7 Implementasi antarmuka *tab menu* detail harga dan grafik aktif

## BAB 6 PENGUJIAN DAN ANALISIS

Bagian ini memuat penjelasan dan analisis hasil pengujian setiap parameter ACO. Nilai parameter yang diuji adalah tetapan siklus semut, tetapan pengendali intensitas feromon, tetapan pengendali visibilitas, tetapan penguapan feromon lokal, tetapan penguapan feromon global, jumlah semut, dan jumlah iterasi. Hasil dari pengujian tersebut adalah kombinasi parameter yang memperoleh MAPE prediksi harga saham terkecil. Kemudian MAPE dan waktu komputasi ACO-BPNN dibandingkan dengan BPNN dalam memprediksi harga saham. Analisis dari seluruh hasil pengujian digunakan untuk menjawab rumusan masalah dalam penelitian ini.

### 6.1 Hasil dan Analisis Pengujian Parameter ACO

Nilai setiap parameter menjadi variabel bebas yang diubah secara sekuensial untuk memperoleh kombinasi nilai parameter terbaik. Parameter tersebut menghasilkan *learning rate*, momentum, dan jumlah *hidden node* yang membuat BPNN mendapatkan MAPE sekecil mungkin. Pengujian ini memprediksi harga saham harian dalam suatu bulan dengan harga saham harian 4 bulan sebelumnya sebagai data latih. Diambil bulan Mei, Agustus, November, dan Desember sebagai sampel dari 12 bulan pada data penelitian ini. Pemilihan sampel tersebut didasarkan pada 3 macam pergerakan harga saham. Bulan Mei mewakili kondisi *bearish*, bulan November mewakili kondisi *bullish*, serta bulan Agustus dan Desember mewakili kondisi *sideways*. Grafik pergerakan harga saham dari data penelitian ini dapat dilihat pada LAMPIRAN A.2. Oleh karena itu, masing-masing nilai dalam interval pengujian setiap parameter akan diuji 4 kali.

#### 6.1.1 Pengujian Nilai Tetapan Siklus Semut

Nilai ( $q_0$ ) yang diuji adalah 0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, dan 0,9. Hasil pengujian ditunjukkan pada Tabel 6.1.

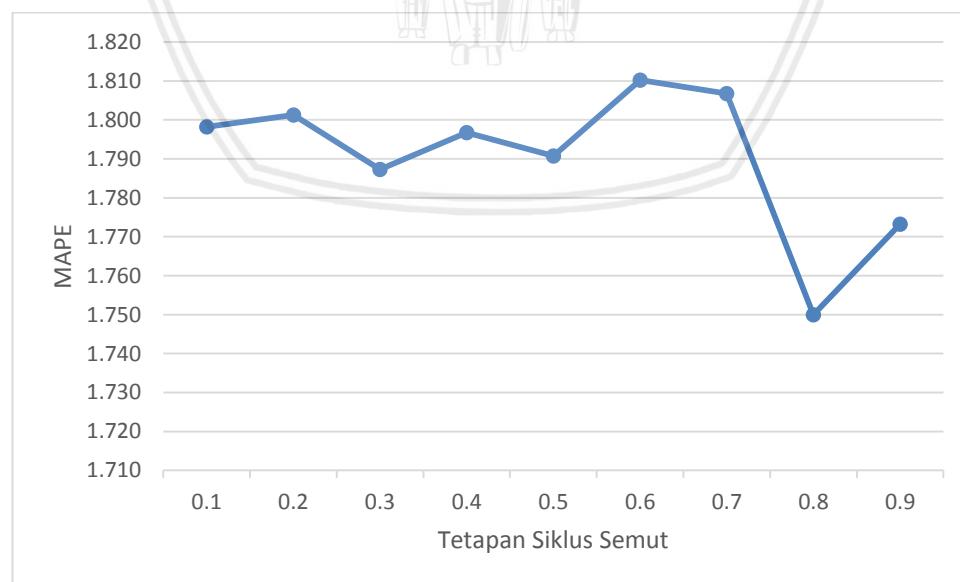
Nilai parameter lain dalam pengujian ini:

1. Nilai tetapan pengendali intensitas feromon : 0,5
2. Nilai tetapan pengendali visibilitas : 0,5
3. Nilai tetapan penguapan feromon lokal : 0,5
4. Nilai tetapan penguapan feromon global : 0,5
5. Jumlah semut : 7
6. Jumlah iterasi : 5
7. Nilai intensitas feromon awal : 0,5
8. Jumlah *epoch* BPNN : 2000

**Tabel 6.1 Hasil pengujian nilai tetapan siklus semut**

Nilai $q_0$	Nilai MAPE Percobaan ke-i				Rata-Rata MAPE
	1	2	3	4	
0,1	2,461	2,190	1,843	0,699	1,798
0,2	2,516	2,157	1,833	0,699	1,801
0,3	2,515	2,142	1,805	0,687	1,787
0,4	2,407	2,226	1,852	0,702	1,797
0,5	2,403	2,230	1,840	0,690	1,791
0,6	2,466	2,262	1,821	0,692	1,810
0,7	2,514	2,187	1,836	0,690	1,807
0,8	2,373	2,112	1,833	0,682	1,750
0,9	2,395	2,171	1,835	0,692	1,773

Hasil pengujian menyatakan bahwa variasi nilai ( $q_0$ ) berpengaruh terhadap MAPE. Ketika nilai ( $q_0$ ) semakin kecil, maka pencarian rute terbaik lebih mengarah kepada eksplorasi lintasan. Sebaliknya, maka pencarian rute terbaik lebih mengarah kepada eksplorasi lintasan. Dalam penelitian ini cara eksplorasi memiliki performa terbaik dengan membuat semut condong untuk menelusuri rute lama. Didapatkan 0,8 sebagai nilai ( $q_0$ ) terbaik dengan rata-rata MAPE 1,750. Selain itu, rata-rata MAPE untuk nilai 0,6-0,9 lebih baik daripada nilai 0,1-0,4. Grafik pergerakan MAPE akibat variasi nilai ( $q_0$ ) ditampilkan pada Gambar 6.1.

**Gambar 6.1 Grafik hasil pengujian nilai tetapan siklus semut**

### 6.1.2 Pengujian Nilai Tetapan Pengendali Intensitas Feromon

Nilai ( $\alpha$ ) yang diuji adalah 0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, dan 0,9. Hasil pengujian ditunjukkan pada Tabel 6.2.

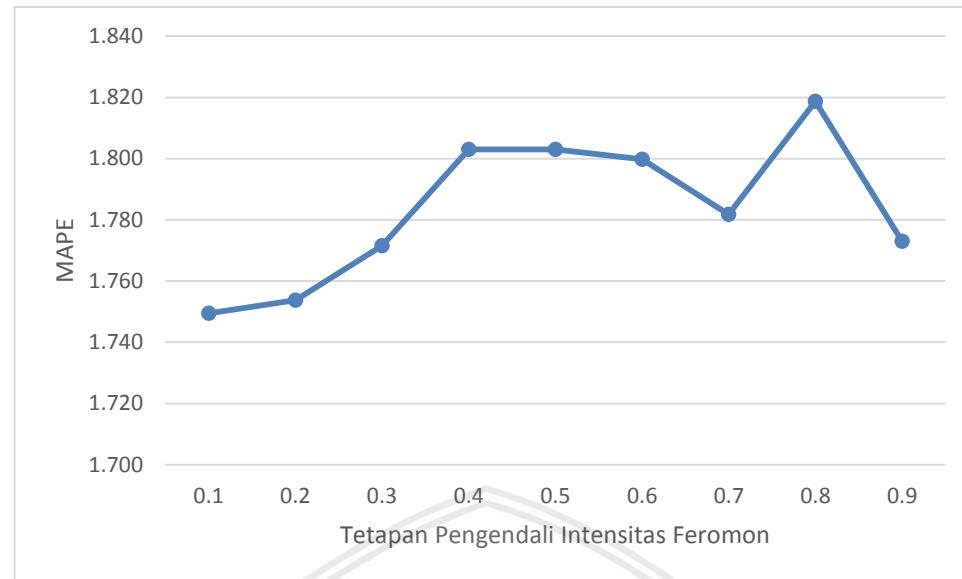
Nilai parameter lain dalam pengujian ini:

1. Nilai tetapan siklus semut : 0,8
2. Nilai tetapan pengendali visibilitas : 0,5
3. Nilai tetapan penguapan feromon lokal : 0,5
4. Nilai tetapan penguapan feromon global : 0,5
5. Jumlah semut : 7
6. Jumlah iterasi : 5
7. Nilai intensitas feromon awal : 0,5
8. Jumlah *epoch* BPNN : 2000

**Tabel 6.2 Hasil pengujian nilai tetapan pengendali intensitas feromon**

Nilai $\alpha$	Nilai MAPE Percobaan ke-i				Rata-Rata MAPE
	1	2	3	4	
0,1	2,382	2,094	1,837	0,685	1,750
0,2	2,378	2,120	1,827	0,690	1,754
0,3	2,417	2,152	1,832	0,685	1,772
0,4	2,493	2,191	1,834	0,694	1,803
0,5	2,461	2,270	1,787	0,694	1,803
0,6	2,508	2,146	1,848	0,697	1,800
0,7	2,478	2,179	1,783	0,687	1,782
0,8	2,508	2,256	1,819	0,692	1,819
0,9	2,409	2,193	1,800	0,690	1,773

Hasil pengujian menyatakan bahwa variasi nilai ( $\alpha$ ) berpengaruh terhadap MAPE. Semakin besar nilai ( $\alpha$ ), maka pencarian rute terbaik lebih bergantung kepada intensitas feromon di setiap jalur. Dalam penelitian ini feromon tidak berperan besar untuk memandu semut mencari rute terbaik. Sehingga nilai ( $\alpha$ ) terbaik dengan rata-rata MAPE sebesar 1,750 adalah 0,1. Selain itu, rata-rata MAPE untuk nilai 0,1-0,4 lebih kecil dari nilai 0,6-0,9. Grafik yang menggambarkan pergerakan MAPE akibat variasi nilai ( $\alpha$ ) ditampilkan pada Gambar 6.2.



**Gambar 6.2 Grafik hasil pengujian nilai tetapan pengendali intensitas feromon**

### 6.1.3 Pengujian Nilai Tetapan Pengendali Visibilitas

Nilai ( $\beta$ ) yang diuji adalah 0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, dan 0,9. Hasil pengujian ditunjukkan pada Tabel 6.3.

Nilai parameter lain dalam pengujian ini:

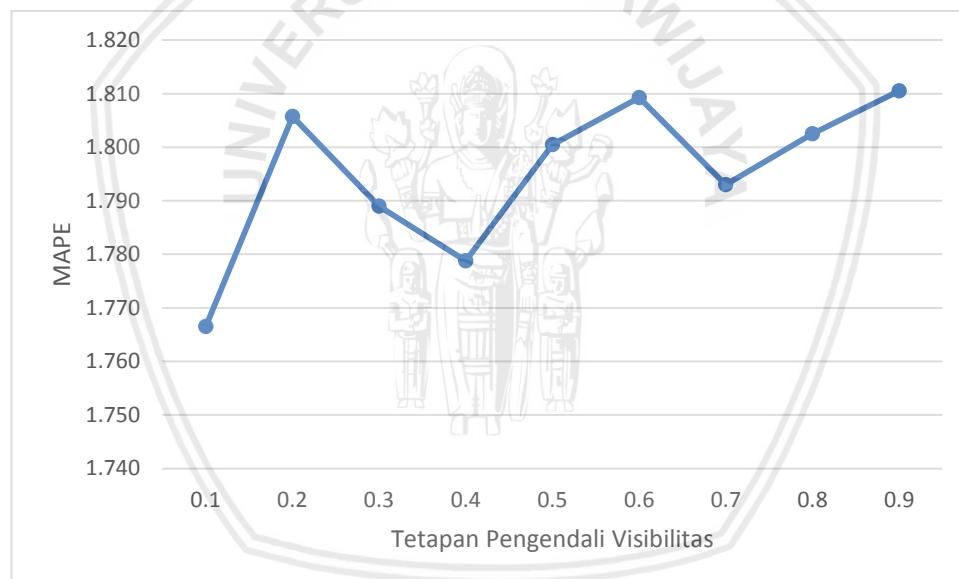
1. Nilai tetapan siklus semut : 0,8
2. Nilai tetapan pengendali intensitas feromon : 0,1
3. Nilai tetapan penguapan feromon lokal : 0,5
4. Nilai tetapan penguapan feromon global : 0,5
5. Jumlah semut : 7
6. Jumlah iterasi : 5
7. Nilai intensitas feromon awal : 0,5
8. Jumlah epoch BPNN : 2000

**Tabel 6.3 Hasil pengujian nilai tetapan pengendali visibilitas**

Nilai $\beta$	Nilai MAPE Percobaan ke-i				Rata-Rata MAPE
	1	2	3	4	
0,1	2,409	2,153	1,822	0,682	1,767
0,2	2,458	2,252	1,818	0,695	1,806
0,3	2,424	2,221	1,822	0,689	1,789
0,4	2,410	2,154	1,844	0,707	1,779
0,5	2,428	2,256	1,828	0,690	1,801

0,6	2,562	2,150	1,835	0,690	1,809
0,7	2,392	2,239	1,849	0,692	1,793
0,8	2,393	2,270	1,860	0,687	1.803
0,9	2,421	2,288	1,841	0,692	1.811

Dikarenakan RMSE pelatihan tidak diketahui saat proses pencarian parameter BPNN berlangsung, maka jarak antar *node* tidak diketahui. Oleh karena itu, tetapan pengendali visibilitas disini berfungsi sebagai tetapan pengendali intensitas feromon juga. Analisis hasil pengujinya pun juga sama seperti pengujian nilai ( $\alpha$ ). Selanjutnya nilai ( $\beta$ ) terbaik dalam pengujian ini adalah 0,1 dengan rata-rata MAPE sebesar 1,767. Selain itu, rata-rata MAPE untuk nilai 0,1-0,4 lebih kecil dari nilai 0,6-0,9. Hal tersebut dapat disebabkan karena penggunaan feromon yang berlebihan untuk memandu semut, sehingga alternatif solusi kurang tersebar. Grafik yang menggambarkan pergerakan MAPE akibat variasi nilai ( $\beta$ ) ditampilkan pada Gambar 6.3.



**Gambar 6.3 Grafik hasil pengujian nilai tetapan pengendali visibilitas**

#### 6.1.4 Pengujian Nilai Tetapan Penguapan Feromon Lokal

Nilai ( $\rho$ ) yang diuji adalah 0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, dan 0,9. Hasil pengujian ditunjukkan pada Tabel 6.4.

Nilai parameter lain dalam pengujian ini:

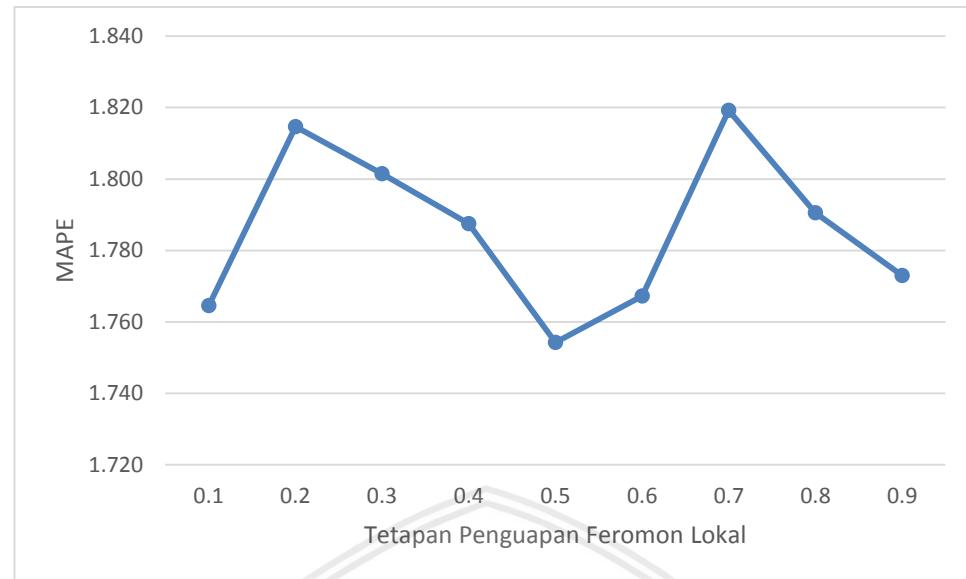
1. Nilai tetapan siklus semut : 0,8
2. Nilai tetapan pengendali intensitas feromon : 0,1
3. Nilai tetapan pengendali visibilitas : 0,1
4. Nilai tetapan penguapan feromon global : 0,5

5. Jumlah semut : 7
6. Jumlah iterasi : 5
7. Nilai intensitas feromon awal : 0,5
8. Jumlah *epoch* BPNN : 2000

**Tabel 6.4 Hasil pengujian nilai tetapan penguapan feromon lokal**

Nilai $\rho$	Nilai MAPE Percobaan ke-i				Rata-Rata MAPE
	1	2	3	4	
0,1	2,418	2,163	1,794	0,683	1,765
0,2	2,438	2,297	1,828	0,696	1,815
0,3	2,401	2,325	1,795	0,685	1,802
0,4	2,389	2,221	1,841	0,699	1,788
0,5	2,395	2,116	1,823	0,683	1,754
0,6	2,418	2,161	1,807	0,683	1,767
0,7	2,435	2,300	1,842	0,700	1,819
0,8	2,496	2,192	1,787	0,687	1,791
0,9	2,394	2,171	1,828	0,699	1,773

Hasil pengujian menyatakan bahwa variasi nilai ( $\rho$ ) berpengaruh terhadap MAPE. Ketika nilai ( $\rho$ ) semakin besar, maka pencarian rute terbaik semakin tersebar. Namun, pembentukan rute terbaik itu sendiri menjadi semakin sulit karena setiap semut condong untuk memilih rutennya masing-masing. Pada pengujian ini nilai ( $\rho$ ) terbaik dengan rata-rata MAPE sebesar 1,754 adalah 0,5. Sehingga dapat dikatakan penguapan feromon yang seimbang saat setiap semut melewati suatu jalur cocok untuk penelitian ini. Grafik yang menggambarkan pergerakan MAPE akibat variasi nilai ( $\rho$ ) ditampilkan pada Gambar 6.4.



Gambar 6.4 Grafik hasil pengujian nilai tetapan penguapan feromon lokal

#### 6.1.5 Pengujian Nilai Tetapan Penguapan Feromon Global

Nilai ( $\delta$ ) yang diuji adalah 0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, dan 0,9. Hasil pengujian ditunjukkan pada Tabel 6.5.

Nilai parameter lain dalam pengujian ini:

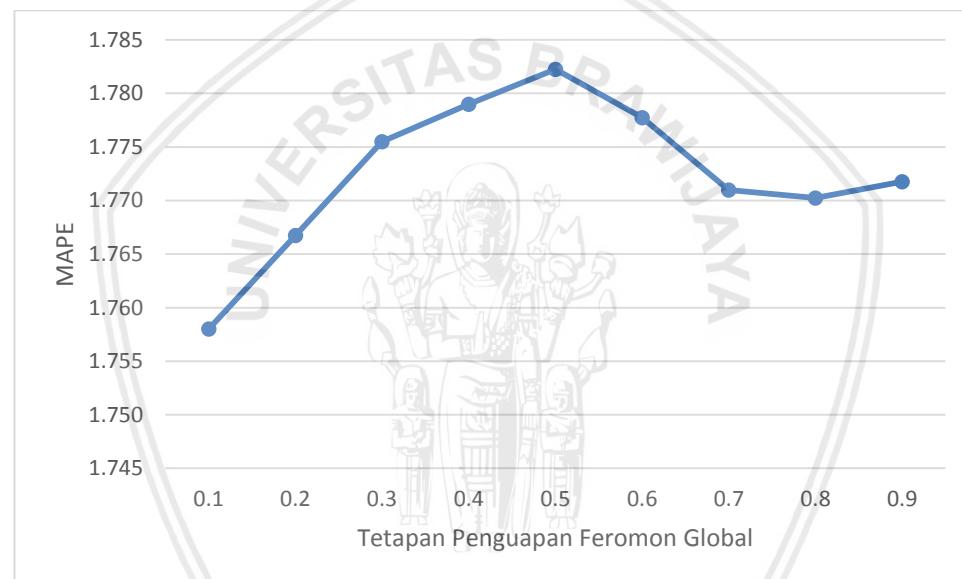
1. Nilai tetapan siklus semut : 0,8
2. Nilai tetapan pengendali intensitas feromon : 0,1
3. Nilai tetapan pengendali visibilitas : 0,1
4. Nilai tetapan penguapan feromon lokal : 0,5
5. Jumlah semut : 7
6. Jumlah iterasi : 5
7. Nilai intensitas feromon awal : 0,5
8. Jumlah epoch BPNN : 2000

Tabel 6.5 Hasil pengujian nilai tetapan penguapan feromon global

Nilai $\delta$	Nilai MAPE Percobaan ke-i				Rata-Rata MAPE
	1	2	3	4	
0,1	2,384	2,113	1,845	0,690	1,758
0,2	2,413	2,159	1,806	0,689	1,767
0,3	2,465	2,160	1,785	0,692	1,776
0,4	2,412	2,198	1,818	0,688	1,779
0,5	2,409	2,190	1,836	0,694	1,782

0,6	2,398	2,190	1,835	0,688	1,778
0,7	2,405	2,155	1,832	0,692	1,771
0,8	2,409	2,168	1,817	0,687	1,770
0,9	2,409	2,118	1,866	0,694	1,772

Hasil pengujian menyatakan bahwa variasi nilai ( $\delta$ ) berpengaruh terhadap MAPE. Semakin besar nilai ( $\delta$ ), maka pembentukan rute terbaik semakin mudah. Akan tetapi, persebaran alternatif solusi yang diperoleh semut semakin sempit. Dalam pengujian ini nilai ( $\delta$ ) yang kecil memberikan performa lebih baik. Sehingga nilai ( $\delta$ ) terbaik dengan rata-rata MAPE sebesar 1,758 adalah 0,1. Grafik yang menggambarkan pergerakan MAPE akibat variasi nilai ( $\delta$ ) ditampilkan pada Gambar 6.5.



Gambar 6.5 Grafik hasil pengujian nilai tetapan penguapan feromon global

### 6.1.6 Pengujian Jumlah Semut

Nilai ( $m$ ) yang diuji adalah 3, 5, 7, 9, dan 11. Hasil pengujian ditunjukkan pada Tabel 6.6.

Nilai parameter lain dalam pengujian ini:

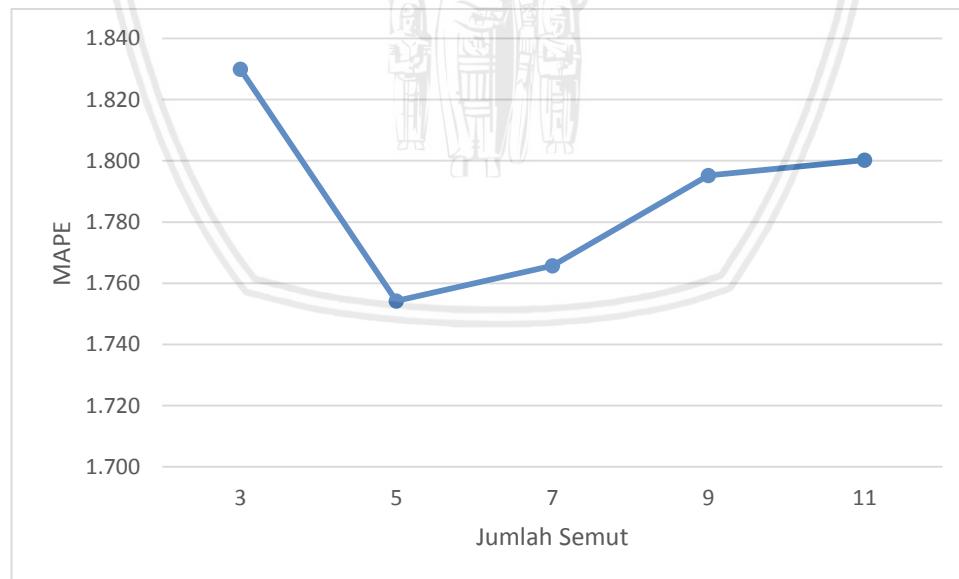
1. Nilai tetapan siklus semut : 0,8
2. Nilai tetapan pengendali intensitas feromon : 0,1
3. Nilai tetapan pengendali visibilitas : 0,1
4. Nilai tetapan penguapan feromon lokal : 0,5
5. Nilai tetapan penguapan feromon global : 0,1
6. Jumlah iterasi : 5

7. Nilai intensitas feromon awal : 0,5  
 8. Jumlah *epoch* BPNN : 2000

**Tabel 6.6 Hasil pengujian jumlah semut**

Nilai <i>m</i>	Nilai MAPE Percobaan ke-i				Rata-Rata MAPE
	1	2	3	4	
3	2,489	2,359	1,790	0,682	1,830
5	2,436	2,075	1,823	0,683	1,754
7	2,394	2,133	1,841	0,695	1,766
9	2,434	2,201	1,854	0,692	1,795
11	2,478	2,209	1,822	0,692	1,800

Hasil pengujian menyatakan bahwa variasi nilai (*m*) berpengaruh terhadap MAPE. Ketika jumlah semut semakin banyak, maka pembentukan rute terbaik membutuhkan waktu yang lebih lama. Akan tetapi, alternatif solusi yang diperoleh semut semakin luas. Pada pengujian ini kuantitas semut yang semakin tinggi akan menurunkan performa. Disamping itu, kuantitas semut yang terlalu kecil memperoleh alternatif solusi terlalu sempit. Sehingga nilai (*m*) terbaik dengan rata-rata MAPE sebesar 1,754 adalah 5. Grafik yang menggambarkan pergerakan MAPE akibat variasi nilai (*m*) ditampilkan pada Gambar 6.6.

**Gambar 6.6 Grafik hasil pengujian jumlah semut**

### 6.1.7 Pengujian Jumlah Iterasi

Nilai ( $NCmax$ ) yang diuji adalah 3, 4, 5, 6, dan 7. Hasil pengujian ditunjukkan pada Tabel 6.7.

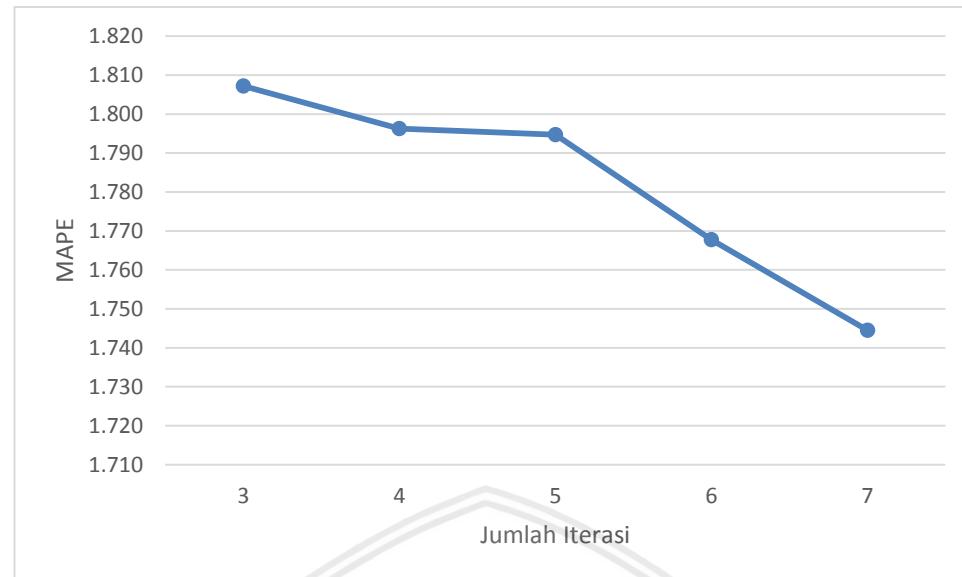
Nilai parameter lain dalam pengujian ini:

1. Nilai tetapan siklus semut : 0,8
2. Nilai tetapan pengendali intensitas feromon : 0,1
3. Nilai tetapan pengendali visibilitas : 0,1
4. Nilai tetapan penguapan feromon lokal : 0,5
5. Nilai tetapan penguapan feromon global : 0,1
6. Jumlah semut : 5
7. Nilai intensitas feromon awal : 0,5
8. Jumlah *epoch* BPNN : 2000

**Tabel 6.7 Hasil pengujian jumlah iterasi**

Nilai $NCmax$	Nilai MAPE Percobaan ke-i				Rata-Rata MAPE
	1	2	3	4	
3	2,430	2,277	1,818	0,704	1,807
4	2,511	2,182	1,805	0,687	1,796
5	2,547	2,120	1,808	0,704	1,795
6	2,402	2,151	1,823	0,695	1,768
7	2,483	2,019	1,789	0,687	1,745

Hasil pengujian menyatakan bahwa variasi nilai ( $NCmax$ ) berpengaruh terhadap MAPE. Ketika jumlah iterasi semakin banyak, maka proses pencarian rute terbaik diulang semakin banyak. Pengulangan tersebut dilakukan dengan harapan akan memperoleh rute yang lebih baik di iterasi selanjutnya. Namun, jika jumlah iterasi berlebihan, maka rute terbaik yang telah dibentuk terhapus oleh hasil eksplorasi rute baru. Pada pengujian ini nilai ( $NCmax$ ) terbaik dengan rata-rata MAPE sebesar 1,745 adalah 7. Grafik yang menggambarkan pergerakan MAPE akibat variasi nilai ( $NCmax$ ) ditampilkan pada Gambar 6.7.



**Gambar 6.7 Grafik hasil pengujian jumlah iterasi**

Dari pengujian parameter ACO, didapatkan kombinasi nilai terbaik, yaitu tetapan siklus semut sebesar 0,8, tetapan pengendali intensitas feromon sebesar 0,1, tetapan pengendali visibilitas sebesar 0,1, tetapan penguapan feromon lokal sebesar 0,5, tetapan penguapan feromon global sebesar 0,1, jumlah semut 5, dan jumlah iterasi 7. Kombinasi nilai tersebut mampu menghasilkan rata-rata MAPE 1,745.

## 6.2 Hasil dan Analisis Pengujian ACO-BPNN dan BPNN

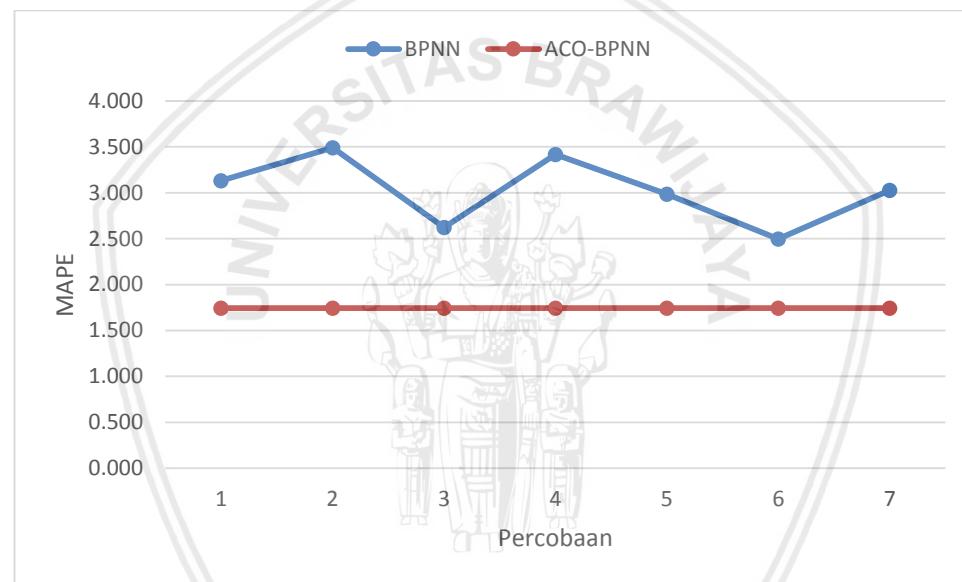
Setelah mendapatkan kombinasi nilai parameter ACO terbaik beserta MAPE-nya, selanjutnya dilakukan pengujian dengan membandingkan MAPE ACO-BPNN dan BPNN. MAPE BPNN didapatkan dengan memprediksi harga saham harian dengan cara yang sama seperti pengujian parameter ACO. Sebelumnya pengujian diulang berdasarkan banyak nilai dalam interval parameter ACO, sekarang pengujian diulang 7 kali berdasarkan jumlah iterasi ACO terbaik. Selain itu, *learning rate*, momentum, dan jumlah *hidden node* BPNN diberikan nilai acak dalam interval yang sama dengan optimasi ACO pada ACO-BPNN. Hasil pengujian ditunjukkan pada Tabel 6.8.

**Tabel 6.8 Hasil pengujian MAPE ACO-BPNN dan BPNN**

Percobaan ke-i	Nilai MAPE Percobaan ke-j				Rata-Rata MAPE BPNN	MAPE ACO-BPNN
	1	2	3	4		
1	7,218	2,136	1.859	1.314	3,132	1,745
2	8,353	2,095	2,485	1,021	3,489	1,745
3	4,876	2,722	1,842	1,047	2,622	1,745
4	8,845	2,098	1,969	0,754	3,417	1,745

5	7,203	2,159	1,808	0,775	2,986	1,745
6	4,298	2,151	1,873	1,663	2,496	1,745
7	7,010	2,192	2,198	0,713	3,028	1,745
<b>Rata-Rata Seluruh Percobaan</b>					3,024	1,745

Hasil pengujian menyatakan bahwa ACO-BPNN memiliki MAPE yang lebih rendah dibandingkan BPNN dengan rata-rata MAPE masing-masing 1,745 dan 3,024. Selain itu, BPNN mencapai rata-rata MAPE terbaik pada percobaan ke-6 sebesar 2,496 dan terburuk pada percobaan ke-2 sebesar 3,489. Sehingga dapat dikatakan ACO berhasil menurunkan MAPE BPNN sebesar 0,751-1,744. Grafik yang menggambarkan pergerakan MAPE BPNN terhadap MAPE ACO-BPNN ditampilkan pada Gambar 6.8.



**Gambar 6.8 Grafik hasil pengujian MAPE ACO-BPNN dan BPNN**

Perbandingan waktu komputasi ACO-BPNN dan BPNN juga dilakukan pada pengujian ini. Kedua metode mendapatkan nilai parameter dengan cara yang sama seperti pengujian perbandingan MAPE sebelumnya. Perbedaannya terletak pada kedua metode yang diuji 4 kali berdasarkan jumlah bulan data sampel, lalu dihitung rata-ratanya. Waktu komputasi adalah waktu yang dibutuhkan untuk melakukan fase pelatihan dan fase pengujian dalam satuan detik. ACO-BPNN memperoleh rata-rata waktu komputasi 232,0019622 detik, sedangkan BPNN 11,5876255 detik. Hasil pengujian ditunjukkan pada Tabel 6.9.

**Tabel 6.9 Hasil pengujian waktu komputasi ACO-BPNN dan BPNN**

Percobaan ke-i	Waktu Komputasi (detik)	
	ACO-BPNN	BPNN
1	196,515518903	11,398292064
2	278,325927019	11,467285871
3	179,540083169	10,687114000
4	273,626319885	12,797810077
<b>Rata-Rata Waktu</b>	<b>232,001962200</b>	<b>11,587625500</b>

Hasil pelatihan dan pengujian ACO-BPNN dengan kombinasi nilai parameter terbaik menggunakan seluruh data ditampilkan pada LAMPIRAN A.1 dan A.2. Kemudian grafik yang menggambarkan perbandingan hasil prediksi tersebut dengan harga aktual ditunjukkan pada LAMPIRAN A.3.

## BAB 7 PENUTUP

Pada bagian akhir dilakukan penarikan kesimpulan dan pemberian saran terhadap hasil penelitian “Prediksi Harga Saham menggunakan Metode *Backpropagation* dengan Optimasi *Ant Colony Optimization*”. Kesimpulan dan saran dipaparkan untuk menjelaskan garis besar penelitian, menjawab rumusan masalah penelitian, serta menyampaikan pertimbangan perbaikan bagi penelitian selanjutnya.

### 7.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan hingga tahap pengujian dan analisis dapat ditarik kesimpulan sebagai berikut:

1. Metode ACO-BPNN dapat diterapkan untuk melakukan prediksi harga saham Bank Rakyat Indonesia (Persero) Tbk. tahun 2018. ACO berfungsi untuk mengoptimalkan kombinasi nilai *learning rate*, momentum, dan jumlah *hidden node* bagi fase pelatihan BPNN. Lalu bobot, *threshold*, dan jumlah *hidden node* terbaik dari fase pelatihan digunakan untuk memprediksi harga saham harian pada suatu periode.
2. Perbedaan pemberian kombinasi nilai parameter ACO berpengaruh pada MAPE yang dihasilkan metode ACO-BPNN. Kombinasi nilai parameter terbaik yang diperoleh yaitu tetapan siklus semut sebesar 0,8, tetapan pengendali intensitas feromon sebesar 0,1, tetapan pengendali visibilitas sebesar 0,1, tetapan penguapan feromon lokal sebesar 0,5, tetapan penguapan feromon global sebesar 0,1, jumlah semut 5, dan jumlah iterasi 7. Kombinasi tersebut menghasilkan rata-rata MAPE 1,745. Oleh karena itu, dapat dikatakan bahwa metode ACO-BPNN sangat baik dalam memprediksi harga saham karena memiliki MAPE < 10%.
3. ACO mampu menurunkan MAPE BPNN dalam memprediksi harga saham sebesar 0,751-1,744 dengan rata-rata MAPE BPNN sebesar 3,024. Namun, waktu komputasi bertambah selama 166,743-267,638 detik dengan selisih rata-rata waktu komputasi kedua metode 220,414 detik.

### 7.2 Saran

Dalam penelitian ini tentunya terdapat beberapa kekurangan yang perlu diperbaiki di penelitian selanjutnya. Bagi peneliti yang melanjutkan penelitian ini dapat dipertimbangkan untuk melakukan pengujian pada jumlah fitur BPNN. Selain itu, pengoptimalan BPNN dengan metode metaheuristik lain layak dicoba. Tidak hanya itu, pemakaian variabel analisis fundamental suatu saham sebagai masukan sistem juga menarik untuk diteliti.

## DAFTAR REFERENSI

- Adrian, A. M., Utamima, A. & Wang, K., 2014. A Comparative Study of GA, PSO and ACO for Solving Construction Site Layout Optimization. *Korean Society of Civil Engineers Journal of Civil Engineering*, [online] Tersedia di: <[https://www.researchgate.net/publication/271661138\\_A\\_comparative\\_study\\_of\\_GA\\_PSO\\_and\\_ACO\\_for\\_solving\\_construction\\_site\\_layout\\_optimization](https://www.researchgate.net/publication/271661138_A_comparative_study_of_GA_PSO_and_ACO_for_solving_construction_site_layout_optimization)> [Diakses 2 April 2018]
- Cao, C., Wang, Y., Jia, L. & Liu, Y., 2008. Research on Optimization of Speed Identification Based on ACO-BP Neural Network and application. In: WCICA (World Congress on Intelligent Control and Automation), 2008. *7<sup>th</sup> World Congress on Intelligent Control and Automation*. Chongqing, China, 25-27 June 2008. Los Alamitos, CA: IEEE.
- Dorigo, M., Birattari, M. & Stutzle, T., 2006. Ant colony optimization. *Computational Intelligence Magazine*, [e-journal] 1(4). Tersedia melalui: IEEE <<https://ieeexplore.ieee.org/document/4129846>> [Diakses 4 April 2019]
- Dreo, J., 2006. *Shortest Path Find by an Ant Colony*. [online] Tersedia di: <[https://commons.wikimedia.org/wiki/File:Aco\\_branches.svg](https://commons.wikimedia.org/wiki/File:Aco_branches.svg)> [Diakses 15 Januari 2019]
- Haroun, S. A., Jamal, B. & Hicham, E. H., 2015. A Performance Comparison of GA and ACO Applied to TSP. *International Journal of Computer Applications*, [online] Tersedia di: <<http://www.ijcaonline.org/archives/volume117/number20/20674-3466>> [Diakses 2 April 2018]
- Jumarwanto, A., 2009. Aplikasi Jaringan Saraf Tiruan Backpropagation untuk Memprediksi Penyakit THT di Rumah Sakit Mardi Rahayu Kudus. *Jurnal Teknik Elektro*, [online] Tersedia di: <<https://journal.unnes.ac.id/nju/index.php/jte/article/view/1601>> [Diakses 20 Januari 2019]
- Khan, A. U., Bandopadhyaya, T. K. & Sharma, S., 2008. Comparisons of Stock Rates Prediction Accuracy using Different Technical Indicators with Backpropagation Neural Network and Genetic Algorithm Based Backpropagation Neural Network. In: ICETET (International Conference on Emerging Trends in Engineering and Technology), 2008. *1<sup>st</sup> International Conference on Emerging Trends in Engineering and Technology*. Nagpur, Maharashtra, India, 16-18 July 2008. Los Alamitos, CA: IEEE.
- Lixing, D., Jinhu, L., Xuemei, L. & Lanlan, L., 2010. Support Vector Regression and Ant Colony Optimization for HVAC Cooling Load Prediction. In: 3CA (International Symposium on Computer, Communication, Control and Automation), 2010. *International Symposium on Computer, Communication, Control and Automation*. Tainan, Taiwan, 5-7 May 2010. Los Alamitos, CA: IEEE.

- Negnevitsky, M., 2005. *Artificial Intelligence A Guide to Intelligent Systems*. [e-book] Harlow, England: Pearson Education. Tersedia di: Academia Denmark <[http://www.academia.dk/BiologiskAntropologi/Epidemiologi/DataMining/Artificial\\_Intelligence-A\\_Guide\\_to\\_Intelligent\\_Systems.pdf](http://www.academia.dk/BiologiskAntropologi/Epidemiologi/DataMining/Artificial_Intelligence-A_Guide_to_Intelligent_Systems.pdf)> [Diakses 25 Januari 2019]
- Nurkholiq, N., Sukmadi, T. & Nugroho, A., 2014. Analisis Perbandingan Metode Logika Fuzzy dengan Jaringan Syaraf Tiruan Backpropagation pada Peramalan Kebutuhan Energi Listrik Jangka Panjang di Indonesia sampai Tahun 2022. *Jurnal Ilmiah Teknik Elektro*, [online] Tersedia di: <<https://ejournal3.undip.ac.id/index.php/transient/article/view/5527>> [Diakses 1 April 2018]
- Rofi'ah, A., Cholissodin, I. & Dewi, C., 2018. *Peramalan Jumlah Pemakaian Air di PT Pembangkit Jawa Bali Unit Gresik dengan Extreme Learning Machine dan Ant Colony Optimization*. S1. Universitas Brawijaya.
- Rosenberg, L., Pescetelli, N. & Willcox, G., 2018. Artificial Swarm Intelligence Amplifies Accuracy when Predicting Financial Markets. In: UEMCON (Annual Ubiquitous Computing, Electronics and Mobile Communication Conference), 2017. *Konferensi Nasional Sistem & Informatika*. Sanur, Bali, Indonesia, 15 November 2008. Los Alamitos, CA: IEEE.
- Setiawan, W., 2008. Prediksi Harga Saham menggunakan Jaringan Syaraf Tiruan Multilayer Feedforward Network dengan Algoritma Backpropagation. Dalam: KNS&I (Konferensi Nasional Sistem & Informatika), 2008. *8<sup>th</sup> Annual Ubiquitous Computing, Electronics and Mobile Communication Conference*. New York, USA, 19-21 October 2017. Bali, Indonesia: STMIK STIKOM Bali.
- Wang, L., Wang, D., Zhu, J. & Zhao, X., 2010. A new model based on improved ACA and BP to predict Silicon content in hot metal. In: ICCMS (International Conference on Computer Modeling and Simulation), 2010. *2<sup>th</sup> International Conference on Computer Modeling and Simulation*. Sanya, China, 22-24 January 2010. Los Alamitos, CA: IEEE.
- WBBA, A. & Pratomo, W. A., 2013. Analisis Fundamental dan Resiko Sistematik terhadap Harga Saham Perbankan yang Terdaftar pada Indeks LQ 45. *Jurnal Ekonomi dan Keuangan*, [online] Tersedia di: <<https://jurnal.usu.ac.id/index.php/edk/article/view/3170>> [Diakses 10 September 2018]
- Yahoo Finance, 2018. *BBRI.JK : Bank Rakyat Indonesia (Persero) - Yahoo Finance*. [online] Tersedia di: <<https://finance.yahoo.com/quote/BBRI.JK/chart?p=BBRI.JK>> [Diakses 1 Januari 2019]

## LAMPIRAN A DATA MENTAH

### A.1 Data

Data harga saham Bank Rakyat Indonesia (Persero) Tbk. periode 1 Januari 2018 sampai 31 Desember 2018.

Date	Open	High	Low	Close	Adj Close	Volume
1/1/2018	3640	3640	3640	3640	3532.066895	0
1/2/2018	3690	3720	3610	3630	3522.363525	91127100
1/3/2018	3630	3630	3560	3610	3502.956543	70148700
1/4/2018	3610	3610	3460	3550	3444.735596	156406300
1/5/2018	3550	3590	3520	3590	3483.549561	87095600
1/8/2018	3590	3590	3560	3590	3483.549561	54559100
1/9/2018	3600	3600	3530	3570	3464.142578	90606700
1/10/2018	3580	3590	3500	3530	3425.328613	116409100
1/11/2018	3540	3540	3480	3540	3435.032227	114378900
1/12/2018	3550	3580	3500	3540	3435.032227	133445300
1/15/2018	3540	3570	3540	3550	3444.735596	86615800
1/16/2018	3530	3610	3530	3610	3502.956543	92106400
1/17/2018	3620	3630	3580	3630	3522.363525	81465500
1/18/2018	3630	3650	3590	3620	3512.659912	79140800
1/19/2018	3620	3630	3580	3620	3512.659912	79576200
1/22/2018	3620	3670	3610	3650	3541.770508	116838000
1/23/2018	3680	3920	3670	3920	3803.764404	169304700
1/24/2018	3900	3900	3820	3830	3716.433105	150067600
1/25/2018	3860	3860	3720	3770	3658.212158	141273900
1/26/2018	3790	3860	3780	3850	3735.840088	135628300
1/29/2018	3850	3850	3760	3800	3687.32251	122622900
1/30/2018	3800	3800	3710	3720	3609.694824	179332000
1/31/2018	3700	3720	3630	3700	3590.287842	263207500
2/1/2018	3730	3760	3700	3700	3590.287842	145586000
2/2/2018	3740	3750	3710	3740	3629.101807	114113200
2/5/2018	3650	3790	3650	3780	3667.915771	112677000

2/6/2018	3690	3720	3660	3680	3570.880859	260283000
2/7/2018	3710	3750	3700	3700	3590.287842	137926500
2/8/2018	3680	3730	3680	3720	3609.694824	111109200
2/9/2018	3690	3730	3680	3710	3599.991211	93140200
2/12/2018	3700	3800	3700	3800	3687.32251	81517000
2/13/2018	3810	3890	3810	3880	3764.950439	108236400
2/14/2018	3900	3910	3830	3860	3745.543457	67665200
2/15/2018	3860	3870	3820	3840	3726.136475	52893500
2/16/2018	3840	3840	3840	3840	3726.136475	0
2/19/2018	3850	3860	3820	3860	3745.543457	63409000
2/20/2018	3840	3860	3780	3800	3687.32251	91506500
2/21/2018	3800	3800	3750	3800	3687.32251	109361300
2/22/2018	3780	3790	3760	3770	3658.212158	103834700
2/23/2018	3760	3810	3760	3790	3677.619141	107749300
2/26/2018	3750	3790	3730	3740	3629.101807	97544400
2/27/2018	3740	3790	3730	3770	3658.212158	76509500
2/28/2018	3780	3790	3740	3780	3667.915771	126297800
3/1/2018	3800	3820	3780	3790	3677.619141	107991400
3/2/2018	3790	3790	3740	3790	3677.619141	61833600
3/5/2018	3800	3800	3770	3770	3658.212158	55258400
3/6/2018	3780	3810	3700	3730	3619.398193	91074000
3/7/2018	3720	3730	3650	3660	3551.473877	141612300
3/8/2018	3730	3770	3690	3770	3658.212158	130641100
3/9/2018	3720	3750	3680	3690	3580.584229	99179200
3/12/2018	3700	3760	3700	3760	3648.508789	144225600
3/13/2018	3760	3770	3720	3760	3648.508789	127738800
3/14/2018	3750	3750	3700	3750	3638.805176	58574900
3/15/2018	3700	3720	3640	3640	3532.066895	150015200
3/16/2018	3640	3680	3550	3680	3570.880859	226448600
3/19/2018	3690	3720	3670	3690	3580.584229	90107600
3/20/2018	3720	3740	3660	3740	3629.101807	105284400

3/21/2018	3750	3820	3740	3810	3697.026123	134110400
3/22/2018	3800	3800	3660	3670	3561.17749	144557000
3/23/2018	3560	3600	3450	3600	3493.25293	196695900
3/26/2018	3580	3610	3540	3600	3493.25293	89913100
3/27/2018	3660	3670	3570	3600	3493.25293	102444600
3/28/2018	3640	3650	3500	3550	3444.735596	98368300
3/29/2018	3590	3600	3520	3600	3493.25293	130860500
3/30/2018	3600	3600	3600	3600	3493.25293	0
4/2/2018	3500	3600	3490	3600	3600	69286700
4/3/2018	3530	3600	3530	3600	3600	45524700
4/4/2018	3600	3600	3460	3490	3490	88879000
4/5/2018	3510	3540	3470	3470	3470	80015200
4/6/2018	3470	3490	3450	3480	3480	67557000
4/9/2018	3530	3530	3480	3530	3530	48595300
4/10/2018	3540	3630	3530	3580	3580	85668600
4/11/2018	3650	3670	3630	3650	3650	87702400
4/12/2018	3670	3670	3570	3590	3590	61989600
4/13/2018	3640	3650	3550	3550	3550	58708000
4/16/2018	3550	3550	3550	3550	3550	0
4/17/2018	3550	3550	3550	3550	3550	0
4/18/2018	3640	3660	3620	3660	3660	67903000
4/19/2018	3660	3660	3660	3660	3660	0
4/20/2018	3660	3660	3660	3660	3660	0
4/23/2018	3630	3640	3580	3580	3580	87721200
4/24/2018	3570	3580	3480	3490	3490	101460000
4/25/2018	3490	3490	3490	3490	3490	0
4/26/2018	3490	3490	3490	3490	3490	0
4/27/2018	3490	3490	3490	3490	3490	0
4/30/2018	3190	3280	3160	3220	3220	130203400
5/1/2018	3220	3220	3220	3220	3220	0
5/2/2018	3230	3240	3190	3230	3230	118974600

5/3/2018	3200	3200	3140	3150	3150	144844300
5/4/2018	3150	3170	3010	3030	3030	182507000
5/7/2018	3080	3100	2990	3050	3050	184094400
5/8/2018	3090	3090	3020	3070	3070	110461000
5/9/2018	3120	3230	3070	3210	3210	185427300
5/10/2018	3210	3210	3210	3210	3210	0
5/11/2018	3260	3300	3160	3160	3160	161449300
5/14/2018	3150	3190	3100	3190	3190	96224700
5/15/2018	3220	3220	3060	3070	3070	108029400
5/16/2018	3010	3070	2950	3070	3070	298925000
5/17/2018	3070	3100	2990	2990	2990	172832900
5/18/2018	3000	3040	2920	2940	2940	215062300
5/21/2018	2900	2920	2750	2760	2760	517921400
5/22/2018	2770	2840	2720	2720	2720	332514600
5/23/2018	2780	2920	2770	2880	2880	409918600
5/24/2018	2930	3070	2920	3060	3060	251856200
5/25/2018	3090	3130	3020	3120	3120	192249700
5/28/2018	3130	3290	3120	3250	3250	186045100
5/29/2018	3250	3250	3250	3250	3250	0
5/30/2018	3250	3280	3140	3140	3140	306227900
5/31/2018	3170	3190	3070	3080	3080	312316900
6/1/2018	3080	3080	3080	3080	3080	0
6/4/2018	3150	3170	3100	3130	3130	211936400
6/5/2018	3150	3160	3090	3110	3110	135979600
6/6/2018	3120	3160	3090	3150	3150	99509500
6/7/2018	3180	3270	3160	3270	3270	183746200
6/8/2018	3210	3250	3110	3140	3140	262972300
6/11/2018	3140	3140	3140	3140	3140	0
6/12/2018	3140	3140	3140	3140	3140	0
6/13/2018	3140	3140	3140	3140	3140	0
6/14/2018	3140	3140	3140	3140	3140	0

6/15/2018	3140	3140	3140	3140	3140	0
6/18/2018	3140	3140	3140	3140	3140	0
6/19/2018	3140	3140	3140	3140	3140	0
6/20/2018	3020	3030	2950	2960	2960	355486900
6/21/2018	2950	2990	2910	2910	2910	135602900
6/22/2018	2900	2980	2870	2980	2980	211595500
6/25/2018	2980	3000	2870	2890	2890	143039300
6/26/2018	2810	2910	2810	2820	2820	185812800
6/27/2018	2880	2880	2830	2840	2840	133617400
6/28/2018	2820	2850	2740	2750	2750	197464500
6/29/2018	2800	2900	2780	2840	2840	216878100
7/2/2018	2890	2910	2820	2850	2850	151334600
7/3/2018	2850	2860	2770	2830	2830	88835200
7/4/2018	2830	2950	2790	2930	2930	128906900
7/5/2018	2890	2920	2860	2910	2910	63494300
7/6/2018	2910	2910	2840	2840	2840	76368500
7/9/2018	2860	3030	2860	3010	3010	127889400
7/10/2018	3040	3080	2980	3050	3050	131641500
7/11/2018	3000	3000	2950	2990	2990	133041300
7/12/2018	2960	3000	2950	2970	2970	91682700
7/13/2018	2960	3030	2960	2970	2970	101916100
7/16/2018	3000	3010	2890	2970	2970	86614100
7/17/2018	2930	2940	2860	2870	2870	137087200
7/18/2018	2870	2900	2850	2850	2850	114063400
7/19/2018	2900	2960	2870	2890	2890	142315400
7/20/2018	2890	2980	2870	2980	2980	72551000
7/23/2018	3000	3050	2990	3040	3040	88975300
7/24/2018	3060	3070	3010	3010	3010	72961600
7/25/2018	2980	3020	2950	2990	2990	71955500
7/26/2018	3050	3070	3030	3040	3040	84696100
7/27/2018	3060	3090	3020	3090	3090	54705200

7/30/2018	3100	3110	3070	3080	3080	66856300
7/31/2018	3080	3080	3000	3070	3070	107458400
8/1/2018	3070	3200	3050	3190	3190	147098300
8/2/2018	3200	3260	3190	3250	3250	158406800
8/3/2018	3270	3350	3230	3330	3330	100589700
8/6/2018	3350	3420	3340	3410	3410	143310900
8/7/2018	3350	3400	3330	3350	3350	89404000
8/8/2018	3400	3400	3310	3330	3330	90710200
8/9/2018	3330	3350	3300	3330	3330	69654300
8/10/2018	3350	3470	3350	3390	3390	115961500
8/13/2018	3300	3320	3130	3140	3140	179328600
8/14/2018	3090	3150	3040	3130	3130	154831700
8/15/2018	3150	3220	3020	3210	3210	138367300
8/16/2018	3180	3180	3050	3050	3050	163665300
8/17/2018	3050	3050	3050	3050	3050	0
8/20/2018	3080	3190	3080	3180	3180	101377000
8/21/2018	3130	3180	3120	3150	3150	59513000
8/22/2018	3150	3150	3150	3150	3150	0
8/23/2018	3180	3270	3130	3270	3270	155835200
8/24/2018	3270	3290	3230	3270	3270	96811700
8/27/2018	3320	3380	3290	3380	3380	112367400
8/28/2018	3330	3380	3300	3330	3330	99638200
8/29/2018	3260	3300	3250	3290	3290	79096600
8/30/2018	3290	3310	3190	3190	3190	83543000
8/31/2018	3180	3180	3110	3180	3180	142665100
9/3/2018	3180	3220	3160	3170	3170	64396700
9/4/2018	3200	3200	3090	3110	3110	76142400
9/5/2018	3070	3070	2900	2930	2930	212574400
9/6/2018	2910	2990	2910	2960	2960	205998900
9/7/2018	2980	3030	2960	3030	3030	141556200
9/10/2018	3030	3030	2960	2970	2970	86547600

9/11/2018	2970	2970	2970	2970	2970	0
9/12/2018	2980	3010	2910	2910	2910	118077900
9/13/2018	2960	3010	2960	3000	3000	125523000
9/14/2018	3030	3070	3000	3070	3070	105124900
9/17/2018	3060	3060	2950	2970	2970	91365700
9/18/2018	2970	2980	2920	2940	2940	138381800
9/19/2018	2970	3020	2970	3000	3000	105405100
9/20/2018	3050	3120	3020	3090	3090	154589900
9/21/2018	3150	3190	3120	3120	3120	203213300
9/24/2018	3120	3120	3000	3010	3010	95256300
9/25/2018	3010	3030	2980	3030	3030	110115400
9/26/2018	3030	3060	2990	2990	2990	66799500
9/27/2018	3000	3080	3000	3070	3070	123373000
9/28/2018	3090	3170	3080	3150	3150	154582600
10/1/2018	3150	3190	3150	3180	3180	91436300
10/2/2018	3150	3180	3080	3110	3110	132361000
10/3/2018	3150	3150	3080	3100	3100	70057000
10/4/2018	3060	3060	2980	3000	3000	147172500
10/5/2018	3000	3040	2960	2980	2980	145459400
10/8/2018	2970	3010	2960	2970	2970	64842800
10/9/2018	2970	3030	2970	3020	3020	51204000
10/10/2018	3030	3040	3010	3010	3010	61258600
10/11/2018	2930	2950	2870	2910	2910	175505100
10/12/2018	2930	2990	2920	2950	2950	118327100
10/15/2018	2960	3030	2960	3000	3000	85743500
10/16/2018	3000	3020	2960	3000	3000	65060100
10/17/2018	3060	3080	3030	3070	3070	80082100
10/18/2018	3070	3100	3050	3090	3090	65133400
10/19/2018	3070	3090	3020	3020	3020	80989700
10/22/2018	3010	3080	3010	3050	3050	48806100
10/23/2018	3040	3070	3010	3020	3020	58554200

10/24/2018	3000	3040	2950	2950	2950	84562000
10/25/2018	2910	3000	2910	3000	3000	131210200
10/26/2018	3000	3020	2980	2990	2990	66228600
10/29/2018	3000	3020	2950	2950	2950	51288900
10/30/2018	2960	3030	2950	3030	3030	72385100
10/31/2018	3050	3150	3030	3150	3150	218534400
11/1/2018	3190	3250	3160	3200	3200	197180000
11/2/2018	3240	3290	3190	3280	3280	167275500
11/5/2018	3260	3280	3220	3270	3270	79594900
11/6/2018	3280	3330	3260	3330	3330	139465400
11/7/2018	3340	3370	3310	3370	3370	162720800
11/8/2018	3400	3490	3390	3450	3450	200247500
11/9/2018	3430	3430	3330	3340	3340	115086300
11/12/2018	3320	3360	3280	3280	3280	77097900
11/13/2018	3280	3360	3230	3350	3350	101884200
11/14/2018	3350	3420	3350	3350	3350	113793200
11/15/2018	3370	3420	3330	3410	3410	106887100
11/16/2018	3430	3550	3420	3490	3490	173768700
11/19/2018	3540	3540	3450	3520	3520	78672100
11/20/2018	3520	3520	3520	3520	3520	0
11/21/2018	3420	3470	3350	3430	3430	138069100
11/22/2018	3430	3550	3430	3550	3550	118307300
11/23/2018	3530	3580	3470	3480	3480	77910800
11/26/2018	3450	3550	3450	3540	3540	82014400
11/27/2018	3480	3620	3480	3600	3600	172878800
11/28/2018	3650	3660	3580	3650	3650	96069700
11/29/2018	3700	3760	3680	3710	3710	138182700
11/30/2018	3660	3690	3610	3620	3620	211886100
12/3/2018	3720	3780	3690	3690	3690	136737400
12/4/2018	3690	3750	3680	3710	3710	134411900
12/5/2018	3640	3680	3610	3650	3650	140136800

12/6/2018	3640	3680	3610	3650	3650	108827300
12/7/2018	3640	3660	3610	3620	3620	111239100
12/10/2018	3610	3630	3570	3610	3610	93075400
12/11/2018	3610	3630	3580	3620	3620	112667200
12/12/2018	3620	3640	3590	3620	3620	88520300
12/13/2018	3690	3690	3650	3680	3680	106283800
12/14/2018	3670	3690	3640	3680	3680	122909500
12/17/2018	3650	3680	3600	3600	3600	124669800
12/18/2018	3550	3640	3550	3630	3630	118738900
12/19/2018	3600	3670	3600	3670	3670	133097500
12/20/2018	3610	3660	3610	3650	3650	67329000
12/21/2018	3610	3630	3600	3620	3620	174467000
12/24/2018	3620	3620	3620	3620	3620	0
12/25/2018	3620	3620	3620	3620	3620	0
12/26/2018	3550	3630	3520	3600	3600	83406800
12/27/2018	3650	3670	3620	3660	3660	71717800
12/28/2018	3680	3680	3650	3660	3660	122945700
12/31/2018	3660	3660	3660	3660	3660	0
1/1/2018	3640	3640	3640	3640	3532.066895	0
1/2/2018	3690	3720	3610	3630	3522.363525	91127100
1/3/2018	3630	3630	3560	3610	3502.956543	70148700
1/4/2018	3610	3610	3460	3550	3444.735596	156406300
1/5/2018	3550	3590	3520	3590	3483.549561	87095600
1/8/2018	3590	3590	3560	3590	3483.549561	54559100
1/9/2018	3600	3600	3530	3570	3464.142578	90606700
1/10/2018	3580	3590	3500	3530	3425.328613	116409100
1/11/2018	3540	3540	3480	3540	3435.032227	114378900

## A.2 Grafik

Grafik pergerakan harga saham Bank Rakyat Indonesia (Persero) Tbk. periode 1 Januari 2018 sampai 31 Desember 2018.



Sumber: Yahoo Finance (2018)

## LAMPIRAN B HASIL PELATIHAN DAN PENGUJIAN

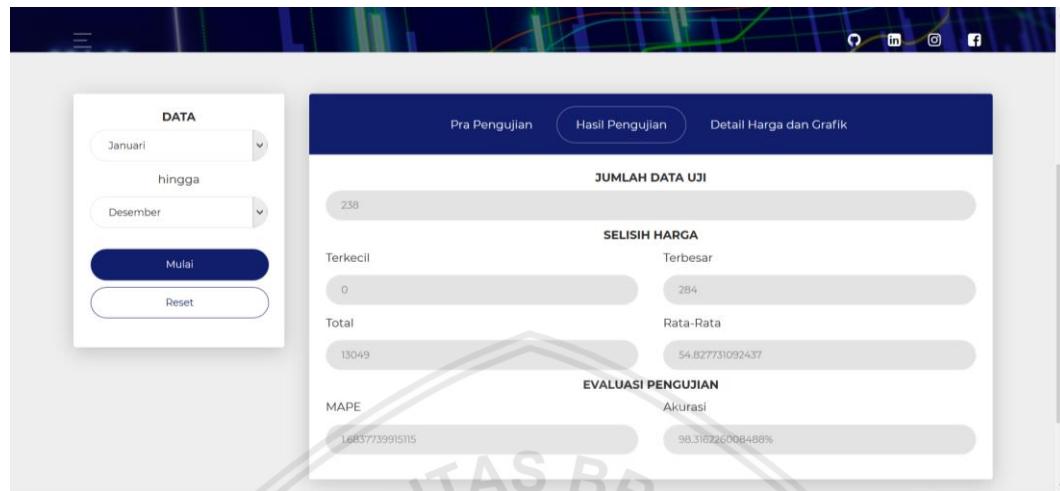
### B.1 Hasil Pelatihan

Pelatihan ACO-BPNN menggunakan kombinasi nilai parameter terbaik dari hasil pengujian. Pelatihan dengan 238 data menghasilkan RMSE sebesar 0,06000510166722, *learning rate* sebesar 0,04, momentum sebesar 0,37, dan jumlah *hidden node* 8.



## B.2 Hasil Pengujian

Pengujian ACO-BPNN dengan 238 data menghasilkan MAPE sebesar 1,6837739915115.



## B.3 Grafik Hasil Pengujian

Grafik yang menunjukkan perbandingan harga hasil prediksi dan harga aktual.

