

Prediksi Harga Saham Harian Menggunakan *Cascade Forward Neural Network (CFNN)* dengan *Particle Swarm Optimization (PSO)*

YUNITA PIPIET SUGANDHI¹, BUDI WARSITO², ARIEF RACHMAN HAKIM³

^{1,2,3}Departemen Statistika, Fakultas Sains dan Matematika, Universitas Diponegoro
email: ¹yunitapi Piet@gmail.com; ²budiwrst2@gmail.com; ³arief.rachman@live.undip.ac.id

ABSTRAK

Jaringan Syaraf Tiruan (JST) merupakan suatu sistem pemrosesan informasi yang karakteristik dan cara kerjanya mirip dengan otak manusia. Jaringan Syaraf Tiruan dapat digunakan sebagai salah satu metode untuk peramalan tanpa harus memenuhi berbagai asumsi seperti pada metode ARIMA. *Feed Forward Neural Network (FFNN)* merupakan salah satu model jaringan syaraf tiruan yang mempunyai arsitektur jaringan sederhana, dimana hubungan antara input dan output terjadi secara tidak langsung yaitu melalui hidden layer. *Cascade Forward Neural Network (CFNN)* merupakan jaringan syaraf tiruan yang memiliki arsitektur mirip dengan *Feed Forward Neural Network (FFNN)*, tetapi terdapat hubungan langsung antara input dan output. CFNN dapat dikombinasikan dengan berbagai metode optimasi seperti *Particle Swarm Optimization (PSO)*. *Particle Swarm Optimization (PSO)* terinspirasi dari kawanan burung yang terbang berkelompok. Perpaduan antara *Cascade Forward Neural Network (CFNN)* dan *Particle Swarm Optimization (PSO)* bertujuan untuk mendapatkan bobot optimum yang meminimumkan error. Pada penelitian ini, data yang digunakan adalah data harga saham harian PT. XL Axiata Tbk. Berdasarkan hasil perhitungan menggunakan CFNN dengan optimasi PSO diperoleh MAPE training sebesar 1,8248% dan MAPE testing sebesar 2,3136% dimana nilai MAPE tersebut masih kurang dari 10% sehingga keakuratan model dikatakan sangat baik.

Kata kunci: *cascade forward neural network*, peramalan, saham, *particle swarm optimization*, *time series*

ABSTRACT

Artificial Neural Network (ANN) is an information-processing system which characteristics and how it worked are similar to human brain. ANN can be used as one of the forecasting method without having to fulfill various assumption such as ARIMA method. Feed Forward Neural Network (FFNN) is one of the ANN model that have a simple network architecture, where the connection between input and output occurs indirectly through the hidden layer. Cascade Forward Neural Network (CFNN) is ANN that its architecture similar to Feed Forward Neural Network (FFNN), but there is also a direct connection between input layer and output layer. CFNN can be combined with various optimization method such as Particle Swarm Optimization (PSO). Particle Swarm Optimization (PSO) inspired by flocks of birds flying in groups. Combination between Cascade Forward Neural Network (CFNN) and Particle Swarm Optimization (PSO) aimed to obtain optimum weight that minimize error. In this study, the data used was daily stock price of PT. XL Axiata Tbk. Based on the calculation results using CFNN with PSO obtained MAPE of training data equal to 1,8248% and MAPE of testing data equal to 2,3136%, which MAPE value is still less than 10% so the accuracy of the model is considered very good.

Keywords: *cascade forward neural network*, forecasting, stock price, *particle swarm optimization*, *time series*

1. PENDAHULUAN

Jaringan Syaraf Tiruan (JST) merupakan sistem pemroses informasi yang memiliki karakteristik mirip jaringan syaraf pada makhluk hidup. Warsito (2009) menjelaskan bahwa prinsip dari pemodel *neural network* dikembangkan dari karakteristik dan cara kerja otak

manusia, dimana dalam memproses informasi otak manusia terdiri dari sejumlah neuron yang melakukan tugas sederhana. Adanya keterhubungan antar neuron menyebabkan otak manusia dapat melakukan fungsi pemrosesan yang sangat kompleks. Terdapat beberapa model *neural network* salah satunya adalah *Cascade Forward Neural Network* (CFNN). Model CFNN adalah model NN yang memiliki arsitektur jaringan mirip dengan FFNN. Jika pada FFNN tidak terdapat hubungan langsung antara *input layer* dengan *output layer*, maka pada CFNN terdapat hubungan langsung antara *input layer* dengan *output layer*.

Pada umumnya CFNN dilatih menggunakan algoritma *backpropagation* untuk memperoleh bobot-bobotnya. Algoritma *backpropagation* memiliki kemampuan untuk memperoleh solusi optimum lokal tetapi kemampuan untuk memperoleh solusi optimum global sangat lemah. Solusi untuk masalah tersebut adalah melatih CFNN menggunakan optimasi *Particle Swarm Optimization* (PSO).

Particle Swarm Optimization (PSO) adalah teknik optimasi berbasis populasi yang dikembangkan oleh James Kennedy dan Russ Eberhart pada tahun 1995. PSO didasarkan pada tingkah laku sosial pada kawanan burung yang terbang berkelompok. Suyanto (2014) menjelaskan bahwa PSO dimulai dengan sekumpulan partikel (solusi) yang dibangkitkan secara acak. Setiap partikel kemudian dievaluasi kualitasnya menggunakan fungsi *fitness*. Selanjutnya, partikel-partikel akan terbang mengikuti partikel yang optimum.

Peramalan data *time series* pada umumnya menggunakan metode *Autoregressive Integrated Moving Average* (ARIMA). Metode tersebut memiliki banyak asumsi yang harus dipenuhi seperti stasioneritas data, autokorelasi residual dan normalitas residual. Akan tetapi dalam praktiknya banyak terdapat data terutama data finansial yang tidak dapat memenuhi asumsi tersebut. Hal tersebut menunjukkan adanya kelemahan model ARIMA untuk digunakan sebagai alat prediksi. Kondisi inilah yang mendorong untuk mencoba menggunakan model CFNN dengan optimasi PSO untuk prediksi data harga saham.

Berdasarkan uraian tersebut, maka dalam penelitian ini dilakukan Prediksi Harga Saham Harian Menggunakan *Cascade Forward Neural Network* (CFNN) dengan *Particle Swarm Optimization* (PSO) pada harga saham harian PT. XL Axiata Tbk.

2. TINJAUAN PUSTAKA

Saham

Menurut Darmadji dan Fakhruddin (2001), saham merupakan tanda penyertaan atau kepemilikan seseorang atau badan dalam suatu perusahaan atau perseroan terbatas. Wujud saham adalah selembar kertas yang menerangkan bahwa pemilik kertas tersebut adalah pemilik perusahaan yang menerbitkan surat berharga tersebut. Porsi kepemilikan ditentukan oleh seberapa besar penyertaan yang ditanamkan di perusahaan tersebut.

Time Series

Menurut Makridarkis dan Wheelwright (1999), runtun waktu (*time series*) adalah suatu rangkaian variabel yang diamati pada interval waktu yang sama ditunjukkan sebagai sebuah deret berkala. Analisis runtun waktu berguna untuk menentukan pola data masa lampau yang telah dikumpulkan secara teratur. Apabila telah dapat menentukan pola data tersebut, maka dapat digunakan sebagai peramalan dalam rangka pengambilan keputusan di masa datang.

Autocorrelation Function (ACF)

Menurut Wei (2006), pada suatu runtun waktu yang stasioner terdapat nilai mean $E(Z_t) = \mu$ dan varian $Var(Z_t) = E(Z_t - \mu)^2 = \sigma^2$ yang konstan. Kovarian antara Z_t dan Z_{t-k} adalah sebagai berikut:

$$\begin{aligned} \gamma_k &= cov(Z_t, Z_{t-k}) = E(Z_t - \mu)(Z_{t-k} - \mu) \\ &= \frac{1}{n} \sum_{t=1}^n (Z_t - \bar{Z})(Z_{t-k} - \bar{Z}) \end{aligned}$$

$$= \frac{1}{n-k} \sum_{t=1}^{n-k} (Z_t - \bar{Z})(Z_{t-k} - \bar{Z})$$

dengan $Var(Z_t) = Var(Z_{t-k}) = \gamma_0$.

Koefisien korelasi untuk lag-k dari data runtun waktu dinyatakan sebagai berikut:

$$\rho_k = \frac{cov(Z_t, Z_{t-k})}{\sqrt{var(Z_t)}\sqrt{var(Z_{t-k})}} = \frac{E(Z_t - \mu)(Z_{t-k} - \mu)}{\sqrt{E(Z_t - \mu)^2}\sqrt{E(Z_{t-k} - \mu)^2}} = \frac{\gamma_k}{\gamma_0}$$

dengan:

- μ = rata-rata
- γ_k = autokovariansi pada lag-k
- γ_0 = autokovariansi pada lag-k
- t = waktu pengamatan, $t = 1, 2, 3, \dots$

Koefisien korelasi ρ_k dapat diduga dengan koefisien korelasi sampel sebagai berikut:

$$\hat{\rho}_k = \frac{\gamma_k}{\gamma_0} = \frac{\sum_{t=1}^{n-k} (Z_t - \bar{Z})(Z_{t-k} - \bar{Z})}{\sum_{t=1}^{n-k} (Z_t - \bar{Z})^2}$$

dengan

- $\hat{\rho}_k$ = koefisien autokorelasi pada lag- k
- k = selisih waktu
- n = jumlah observasi
- \bar{Z} = rata-rata dari pengamatan
- Z_t = pengamatan pada waktu ke- t
- Z_{t-k} = pengamatan pada waktu ke- $t - k$

Plot *Autocorrelation Function* (ACF) digunakan untuk menentukan proses *Moving Average* (MA) yaitu dengan melihat lag-lag yang signifikan pada batas plot ACF.

Partial Autocorrelation Fuction (PACF)

Menurut Soejoeti (1987), autokovariansi merupakan variansi bersama dari variabel yang sama, dalam hal ini adalah data runtun waktu itu sendiri. Autokovariansi dapat dicari dengan menggunakan fungsi autokorelasi parsial (PACF). PACF didefinisikan sebagai $\{\phi_{kk}; k = 1, 2, \dots\}$, yaitu himpunan fungsi autokorelasi parsial untuk berbagai lag k dengan rumus sebagai berikut:

$$\phi_{kk} = \frac{|P_{\sim k}^*|}{|P_{\sim k}|}, \text{ untuk } k = 1, 2, \dots \tag{1}$$

dimana $P_{\sim k} = \begin{bmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{k-1} \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_{k-2} \\ \rho_2 & \rho_1 & 1 & \cdots & \rho_{k-3} \\ \vdots & \vdots & \vdots & & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \cdots & 1 \end{bmatrix}$

$P_{\sim k}$: matriks autokorelasi $k \times k$

$P_{\sim k}^*$: $P_{\sim k}$ dengan kolom terakhir diganti dengan $\begin{bmatrix} \rho_1 \\ \rho_2 \\ \dots \\ \rho_k \end{bmatrix}$

Plot *Partial Autocorrelation Function* (PACF) digunakan untuk menentukan proses *Autoregressive* (AR) yaitu dengan melihat lag-lag yang signifikan pada batas plot PACF.

Neural Network

Haykin (1994) menyatakan bahwa Jaringan Syaraf Tiruan (JST) atau *Neural Network* (NN) terinspirasi dari cara kerja otak manusia yang sangat berbeda dengan komputer digital pada umumnya. Otak manusia sangat kompleks, tidak linier dan bersifat paralel dalam hal memproses informasi. Otak manusia juga mempunyai kemampuan untuk mengatur unsur struktural yang dikenal dengan *neuron*, sehingga sejauh ini dalam memproses informasi otak manusia jauh lebih cepat daripada komputer digital yang ada pada saat ini. JST memiliki dua tahap pemrosesan informasi, yaitu tahap penelitian dan tahap pengujian. Warsito (2009) menjelaskan bahwa tahap pelatihan dimulai dengan memasukkan pola-pola belajar (data latih) ke dalam jaringan. Dengan menggunakan pola-pola ini, jaringan akan mengubah-ubah bobot yang menjadi penghubung antar *node*, sedangkan tahap pengujian dilakukan terhadap suatu pola masukan yang belum pernah dilatihkan sebelumnya (data uji) dengan menggunakan bobot hasil tahap pelatihan. Diharapkan bobot-bobot hasil pelatihan yang sudah menghasilkan *error* minimal juga akan memberikan *error* yang kecil pada tahap pengujian.

Menurut Kusumadewi (2003), NN memiliki komponen-komponen utama yang terdiri dari *input*, fungsi aktivasi, neuron, bias dan threshold, *hidden layer*, *output*. Dalam NN terdapat susunan dari neuron-neuron dalam lapisan yang mempunyai pola keterhubungannya dalam dan antar lapisan yang disebut dengan arsitektur jaringan. NN biasanya terdiri dari lapisan *input* (*input layer*), lapisan tersembunyi (*hidden layer*) dan lapisan *output* (*output layer*). *Feed Forward Neural Network* (FFNN) adalah salah satu model NN yang mempunyai arsitektur jaringan cukup sederhana dengan satu lapisan tersembunyi (*hidden layer*).

Cascade Forward Neural Network (CFNN)

Warsito *et al* (2018) menjelaskan jika hubungan yang terbentuk dari perceptron dan *multi layer network* dikombinasikan, maka akan terbentuk jaringan dengan hubungan langsung antara *input layer* dan *output layer*, selain hubungan tidak langsung yang terbentuk antara *input layer* dan *output layer* yaitu melalui *hidden layer*. Jaringan yang terbentuk dari pola hubungan tersebut disebut dengan *Cascade Forward Neural Network*.

Arsitektur jaringan *Cascade Forward Neural Network* (CFNN) hampir serupa dengan *Feed Forward Neural Network* (FFNN). Pada FFNN hubungan *input* dan *output* bukan merupakan hubungan langsung, sedangkan pada CFNN terdapat hubungan langsung antara *input* dan *output*.

Persamaan yang terbentuk dari model CFNN dengan satu unit *output layer* dapat ditulis sebagai berikut:

$$y = \sum_{i=1}^I f^i u_{ik} x_i + f^0 (w_{ok} + \sum_{j=1}^J w_{jk} f^j (v_{oj} + \sum_{i=1}^I v_{ij} x_i)) \quad (2)$$

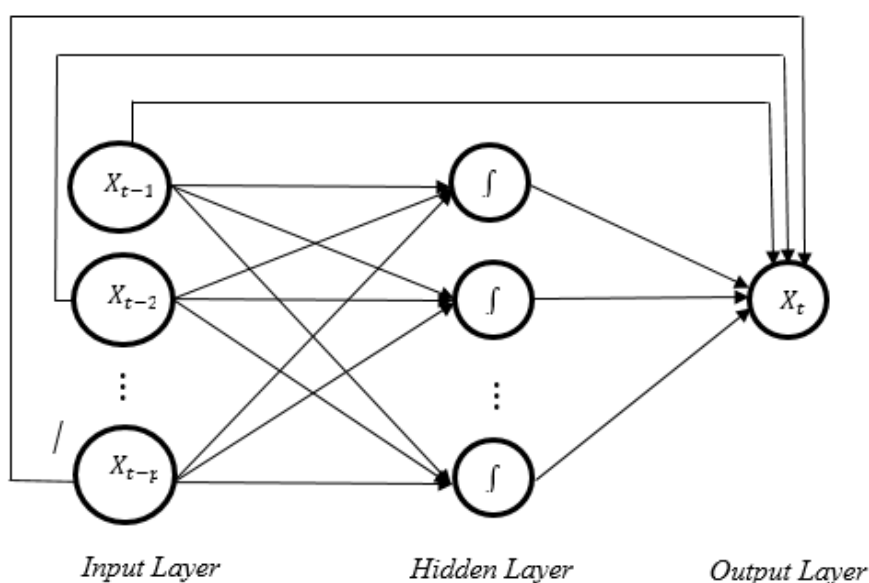
dengan:

v_{ij} : bobot untuk hubungan antara unit-unit pada *input layer* ke-i dan unit-unit pada *hidden layer* ke-j

w_{jk} : bobot untuk hubungan antara unit-unit pada *hidden layer* ke-j dan unit pada *output layer* ke-k

u_{ik} : bobot untuk hubungan antara unit-unit pada *input layer* ke-i dan unit pada *output layer* ke-k

- v_{oj} : bobot bias pada *input layer* ke *hidden layer* ke- j
- w_{ok} : bobot bias pada *hidden layer* ke *output layer* ke- k
- f^j : fungsi aktivasi yang digunakan pada *hidden layer*
- f^0 : fungsi aktivasi yang digunakan pada *output layer*
- f^i : fungsi aktivasi yang digunakan pada *input layer* ke *output layer*



Gambar 1. Arsitektur Cascade Forward neural Network

Particle Swarm Optimization (PSO)

Menurut Suyanto (2014), *Swarm Intelligent* (SI) merupakan suatu disiplin ilmu yang berhubungan dengan alam dan sistem-sistem buatan yang tersusun atas banyak individu yang berkoordinasi menggunakan kontrol terdesentralisasi dan dapat mengorganisasi diri sendiri (*self-organization*). Secara khusus, disiplin ilmu ini fokus pada tingkah laku kolektif yang dihasilkan dari interaksi lokal antara satu individu dengan individu lain dan antara individu dengan lingkungannya. Logika *Swarm Intelligent* (SI) kemudian menginspirasi sebagian besar peneliti untuk mengembangkan algoritma berdasarkan pada kecerdasan binatang, terutama untuk mengatasi permasalahan optimasi. Berbagai algoritma mengenai perilaku kolektif diusulkan untuk mengatasi permasalahan optimasi. Permasalahan optimasi mengarah pada studi untuk menentukan nilai minimal ataupun maksimal dari sebuah fungsi. Nilai maksimal atau minimal tersebut disebut sebagai nilai optimal. Menurut Santosa (2017), ada dua jenis nilai optimal, yaitu *global optimum* dan *local optimum*. Sebagian besar algoritma optimasi dibuat dengan tujuan mencari nilai *global optimum*.

Salah satu algoritma *Swarm Intelligent* (SI) adalah *Particle Swarm Optimization* (PSO). *Particle Swarm Optimization* (PSO) adalah teknik optimasi populasi yang dikembangkan oleh James Kennedy dan Russ Eberhart pada tahun 1995. Teknik ini terinspirasi oleh tingkah laku sosial pada kawanan burung yang terbang berduyun-duyun atau pada segerombolan ikan yang berenang berkelompok membentuk formasi tertentu tanpa bertabrakan satu sama lain meskipun jarak antar individu begitu dekat (Suyanto, 2014). Setiap burung dianggap sebagai partikel yang merepresentasikan solusi dari suatu permasalahan yang memiliki posisi (x) dan kecepatan (v). *Particle Swarm Optimization* (PSO) memiliki dua fungsi utama, yaitu update kecepatan dan update posisi yang digunakan untuk memperbarui kecepatan dan posisi

partikel sehingga akan terus dipercepat sampai mendekati posisi partikel terbaik dari partikel tersebut (*local best*) dan partikel terbaik dari seluruh kawanan (*global best*) sampai kondisi minimum *error* tercapai. PSO dimulai dengan sekumpulan partikel (solusi) yang dibangkitkan secara acak. Berikut ini merupakan rumus perhitungan PSO:

$$v_j^{t+1} = w \cdot v_j^t + c_1 \cdot r_1 (pBest_j^t - x_j^t) + c_2 \cdot r_2 (gBest_j^t - x_j^t)$$

$$x_j^{t+1} = x_j^t + v_j^{t+1}$$

dimana:

v_j^t : kecepatan partikel j pada iterasi ke- t

x_j^t : posisi partikel j pada iterasi ke- t

w : bobot inersia

c_1 & c_2 : *learning rates*

r_1 & r_2 : bilangan acak yang bernilai antara 0 sampai 1

$pBest_j^t$: posisi terbaik dari partikel j pada iterasi ke- t

$gBest_j^t$: global optimal dari partikel j pada iterasi ke- t

3. METODOLOGI PENELITIAN

Jenis dan Sumber Data

Data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dari *website* Yahoo finance. Data tersebut merupakan data harga saham penutupan (harian) PT. XL Axiata Tbk periode 1 Januari 2016 sampai dengan 31 Desember 2018 dengan menggunakan hari aktif.

Variabel Penelitian

Variabel yang digunakan dalam penelitian ini adalah data harga saham penutupan (harian) PT. XL Axiata Tbk periode 1 Januari 2016 sampai dengan 31 Desember 2018 yang terdiri dari 754 data.

Langkah-Langkah Penelitian

1. Pengumpulan data saham harian PT. XL Axiata Tbk terhitung mulai dari periode 1 Januari 2016 sampai dengan 31 Desember 2018 dengan menggunakan hari aktif.
2. Menentukan nilai *input* berdasarkan plot PACF (*Partial Auto Correlation Function*).
3. Melakukan pembagian data menjadi 2, yaitu data *training* dan data *testing*.
4. Menentukan jumlah *neuron hidden layer*.
5. Menentukan banyaknya partikel (N), nilai c_1 dan c_2 serta bobot inersia pada PSO.
6. Menginisialisasi bobot sebagai partikel pada PSO.
7. Mengevaluasi bobot dengan PSO
 - a. Mengevaluasi nilai *fitness* dari setiap partikel.
 - b. Memperbarui nilai *pBest* (posisi terbaik) dan *gBest* (global optimal).
 - c. Memperbarui kecepatan dan posisi dari setiap partikel dengan cara iterasi.
 - d. Jika nilai *fitness* antar partikel tersebut memenuhi konvergen maka iterasi akan berhenti. Jika tidak lanjutkan ke iterasi selanjutnya dengan cara yang sama.
8. Menentukan bobot akhir, dimana algoritma PSO ini akan membantu mencari bobot yang dapat menghasilkan MSE terkecil.
9. Melakukan proses *testing*.
10. Melakukan peramalan untuk beberapa periode ke depan.

Tabel 1. Nilai MAPE Pembagian Data dan Jumlah Neuron *Hidden Layer*

Pembagian Data	Jumlah Neuron	MAPE <i>Training</i> (%)	MAPE <i>Testing</i> (%)
60% data <i>training</i> 40% data <i>testing</i>	1	1,8422	2,2034
	2	1,8425	2,2094
	3	1,8322	2,1997
	4	1,8357	2,1936
	5	1,8349	2,2014
	6	1,8380	2,2105
	7	1,8371	2,1927
	8	1,8284	2,1841
	9	1,8391	2,2272
	10	1,8335	2,1976
70% data <i>training</i> 30% data <i>testing</i>	1	1,8330	2,3407
	2	1,8334	2,3411
	3	1,8270	2,3324
	4	1,8293	2,3244
	5	1,8248	2,3136
	6	1,8476	2,3370
	7	1,8504	2,3749
	8	1,8265	2,3198
	9	1,8337	2,3381
	10	1,8379	2,3489
80% data <i>training</i> 20% data <i>testing</i>	1	1,9262	2,1747
	2	1,9385	2,1825
	3	1,9223	2,1767
	4	1,9254	2,1779
	5	1,9240	2,1735
	6	1,9335	2,1821
	7	1,9311	2,1653
	8	1,9216	2,1743
	9	1,9194	2,1933
	10	1,9302	2,1799
90% data <i>training</i> 10% data <i>testing</i>	1	1,9821	2,0684
	2	1,9817	2,0666
	3	1,9677	2,0488
	4	1,9696	2,0518
	5	1,9712	2,0525
	6	1,9714	2,0505
	7	1,9758	2,0561
	8	1,9671	2,0456
	9	1,9823	2,0511
	10	1,9886	2,0720

4. HASIL DAN PEMBAHASAN

Penentuan *Input*

Penentuan *input* jaringan dilakukan dengan menggunakan plot PACF. Hal ini dikarenakan plot PACF digunakan untuk menentukan proses *Autoregressive* (AR), dimana proses AR menggambarkan bahwa data masa kini dipengaruhi oleh data periode sebelumnya. *Input* jaringan berdasarkan lag-lag yang signifikan pada plot PACF yang dapat dilihat dari garis yang melewati batas plot PACF. Berdasarkan plot PACF diperoleh 1 variabel lag yang akan dijadikan *input*, yaitu lag 1 atau dapat dikatakan bahwa x_t dipengaruhi oleh x_{t-1} sehingga data yang digunakan sebagai *input* ada sebanyak 753 data yang akan dilatih dengan jaringan CFNN.

Pembagian Data dan Penentuan Jumlah Neuron pada *Hidden Layer*

Dalam penelitian ini penentuan pembagian data *training* dan data *testing* serta jumlah *neuron* pada *hidden layer* dilakukan dengan *trial and error* menggunakan CFNN dengan algoritma PSO. Selanjutnya, pemilihan perbandingan pembagian data dan jumlah neuron pada *hidden layer* dilihat dari MAPE *training* terkecil. Hasilnya disajikan dalam Tabel 1. Berdasarkan Tabel 1 dapat dilihat bahwa nilai MAPE *training* terkecil adalah 1,8248% yaitu pada pembagian data 70% data *training* dan 30% data *testing* dengan jumlah *hidden layer* sebanyak 5 *neuron*.

Arsitektur Jaringan CFNN

Berdasarkan pada percobaan pembagian data dan penentuan jumlah neuron pada *hidden layer*, maka arsitektur jaringan yang terbentuk terdiri dari 1 neuron pada *input layer*, 1 unit *hidden layer* yang terdiri dari 5 neuron, 1 neuron pada *output layer* dan melibatkan bias. Sedangkan fungsi aktivasi yang digunakan pada *hidden layer* adalah sigmoid bipolar dan fungsi aktivasi yang digunakan pada *output layer* adalah fungsi pureline.

Berdasarkan arsitektur jaringan CFNN yang telah terbentuk maka jumlah bobot atau parameter yang akan diestimasi dengan menggunakan PSO sebanyak 17 bobot terdiri dari 5 bobot memberikan sinyal *input* ke *hidden layer*, 5 bobot bias untuk *hidden layer*, 5 bobot neuron dari *hidden layer* ke *output layer*, 1 bobot neuron dari *input layer* ke *output layer*, 1 bobot bias untuk *output layer*.

Proses *Training* CFNN dengan Algoritma PSO

Ada beberapa hal yang harus ditetapkan terlebih dahulu sebelum memulai algoritma PSO diantaranya:

1. Jumlah Partikel (N)

Untuk mengurangi evaluasi fungsi yang diperlukan untuk menemukan solusi, sebaiknya ukuran N tidak terlalu besar tetapi juga tidak terlalu kecil. Pada penelitian ini, jumlah partikel (N) yang digunakan adalah 20 partikel.

2. Nilai c_1 dan c_2

Pada penelitian ini, nilai c_1 dan c_2 yang digunakan adalah 2 ($c_1 = c_2 = 2$).

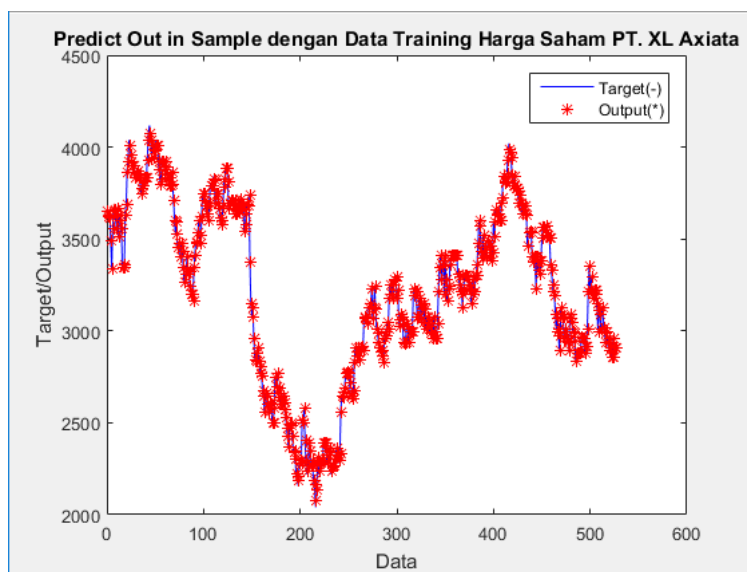
3. Bobot Inertia (w)

Bobot inertia digunakan untuk mengurangi kecepatan pada formula update kecepatan. Pada penelitian ini, cara penentuan nilai w yang digunakan adalah nilai w yang random dan konstan dengan rumus sebagai berikut:

$$w = 0,1 + (\text{rand} \times 0,4)$$

4. Penentuan Nilai *Fitness* pada Posisi Awal Partikel

Penentuan nilai *fitness* pada PSO digunakan untuk menentukan posisi individu ($pBest$) dan posisi global ($gBest$). Nilai *fitness* yang digunakan dalam penelitian ini adalah fungsi meminimumkan nilai MSE.



Gambar 2. Prediksi Target dan Output Data Training

Tabel 2. Bobot Akhir Hasil Pelatihan

Bobot	Keterangan
0,12150	Bobot input x_{t-1} ke hidden layer 1
-0,76140	Bobot input x_{t-1} ke hidden layer 2
-0,21711	Bobot input x_{t-1} ke hidden layer 3
-0,89683	Bobot input x_{t-1} ke hidden layer 4
0,058223	Bobot input x_{t-1} ke hidden layer 5
0,55992	Bobot bias ke hidden layer 1
0,65129	Bobot bias ke hidden layer 2
-0,23194	Bobot bias ke hidden layer 3
1,0000	Bobot bias ke hidden layer 4
-0,47488	Bobot bias ke hidden layer 5
0,70084	Bobot input x_{t-1} ke output
-0,24972	Bobot bias ke output
-0,73043	Bobot hidden layer 1 ke output
-0,99793	Bobot hidden layer 2 ke output
-0,98129	Bobot hidden layer 3 ke output
0,77331	Bobot hidden layer 4 ke output
-0,85351	Bobot hidden layer 5 ke output

5. Inisialisasi Bobot sebagai Partikel PSO

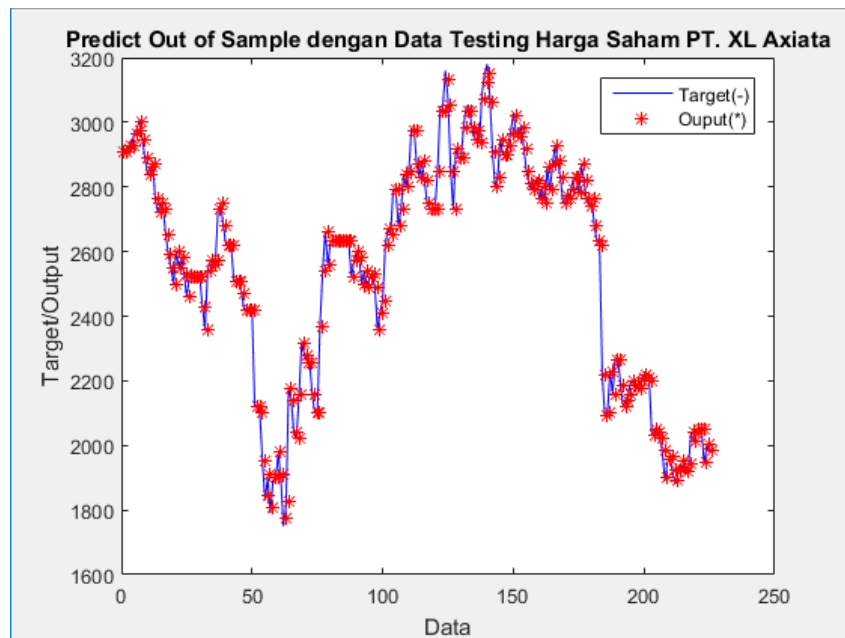
Pada penelitian ini, parameter CFNN yang dioptimasi menggunakan PSO adalah bobot, sehingga dapat dikatakan bahwa bobot merupakan partikel pada PSO. Inisialisasi bobot awal dilakukan sebelum dilakukan proses iterasi pertama kali, jadi inisialisasi bobot awal dilakukan pada iterasi ke-0. Bobot awal diperoleh dengan membangkitkan bilangan random, dimana banyaknya bilangan random sesuai dengan jumlah bobot yang ada dalam jaringan dengan interval pembangkitan bilangan random antara -1 dan 1. Algoritma akan mencari bobot

optimal dalam ruang pencarian yang bernilai antara -1 dan 1. Ketika terdapat nilai bobot yang melebihi nilai -1 dan 1 maka bobot tersebut akan diinisialisasi ulang. Jika melebihi -1 maka akan bernilai sama dengan -1 dan jika melebihi 1 akan bernilai sama dengan 1. Karena jumlah partikel (N) yang digunakan adalah 20 maka matriks bobot awal yang terbentuk sebanyak 20.

Setelah membangun arsitektur jaringan CFNN terbaik, maka proses *training* dapat dilakukan dengan memasukkan data input (x_{t-1}) dan target (x_t) sebanyak 527 data, sehingga diperoleh hasil seperti pada Gambar 2. Berdasarkan Gambar 2 dapat dilihat bahwa bahwa pelatihan jaringan telah memberikan hasil prediksi cukup akurat yang ditunjukkan oleh kedekatan antara target (data asli = x_t) yang disimbolkan dengan garis dan prediksinya ($output = y_t$) yang disimbolkan dengan bintang (*). Nilai MAPE yang dihasilkan dari proses *training* sebesar 1,8248%. Nilai bobot atau parameter yang optimum ditampilkan dalam tabel berikut:

Proses Testing

Proses *testing* dilakukan dengan memasukan data testing dan target yaitu sebanyak 226 data kemudian diuji dengan menggunakan bobot akhir yang telah dihasilkan dari proses *training*. Hasil proses *testing* adalah sebagai berikut:



Gambar 3. Prediksi Target dan *Output* Data *Testing*

Berdasarkan Gambar 3 dapat dilihat bahwa proses *testing* telah berhasil memberikan prediksi yang cukup akurat yang ditunjukkan oleh kedekatan antara target (data asli = x_t) yang disimbolkan dengan garis dan prediksinya ($output = y_t$) yang disimbolkan dengan bintang (*). Nilai MAPE yang dihasilkan dari proses *testing* sebesar 2,3136%.

Penentuan Model

Berdasarkan bobot atau parameter yang optimum dari hasil pelatihan maka dapat diperoleh model CFNN untuk harga saham harian PT. XL Axiata Tbk dapat dituliskan sebagai berikut:

$$y = u_{11}x_1 + w_{o1} + \sum_{j=1}^5 w_{j1} \left(\frac{1 - e^{-(v_{oj} + \sum_{j=1}^5 v_{1j}x_1)}}{1 + e^{-(v_{oj} + \sum_{j=1}^5 v_{1j}x_1)}} \right) \tag{3}$$

dengan:

$$\begin{array}{lll}
 u_{11} = 0,70084 & w_{51} = -0,85351 & v_{11} = 0,1215 \\
 w_{01} = -0,24972 & v_{01} = 0,55992 & v_{12} = -0,7614 \\
 w_{11} = -0,73043 & v_{02} = 0,65129 & v_{13} = -0,21711 \\
 w_{21} = -0,99793 & v_{03} = -0,23194 & v_{14} = -0,89683 \\
 w_{31} = -0,98129 & v_{04} = 1 & v_{15} = 0,058223 \\
 w_{41} = 0,77331 & v_{05} = -0,47488 &
 \end{array}$$

Peramalan

Berikut ini adalah hasil peramalan harga saham harian PT. XL Axiata Tbk untuk 10 langkah kedepan berdasarkan pada model CFNN terbaik yang telah diperoleh sebelumnya adalah sebagai berikut:

Tabel 3. Hasil Peramalan

t Langkah ke depan	Hasil Peramalan	Aktual	Error
1	1.985,335	1980	5,3351
2	1.990,369	1980	10,3687
3	1.995,131	2020	24,8694
4	1.999,635	2040	40,3649
5	2.003,911	2140	136,0892
6	2.007,958	2190	182,0423
7	2.011,804	2250	238,1956
8	2.015,451	2250	234,5491
9	2.018,926	2200	181,0742
10	2.022,229	2110	87,7709
MAPE			5,2075%

Berdasarkan Tabel 3, diperoleh nilai MAPE sebesar 5,2075%. Dapat dilihat bahwa nilai MAPE yang didapatkan kurang dari 10%, maka dapat dikatakan keakuratan hasil peramalan masih sangat baik.

KESIMPULAN

Berdasarkan hasil penelitian, dapat diambil kesimpulan sebagai berikut:

1. Pembagian data *training* sebesar 70% dan data *testing* sebesar 30% dari total 753 data merupakan pembagian data terbaik karena menghasilkan nilai MAPE *training* yang relatif lebih kecil daripada pembagian data 60:40, 80:20, 90:10.
2. Arsitektur model terbaik jaringan syaraf tiruan *Cascade Forward Neuron Network* dibangun dari 1 *neuron* lapisan *input* yaitu x_{t-1} , 5 *neuron* lapisan tersembunyi dan 1 *neuron* lapisan *output*.
3. Model persamaan CFNN yang dihasilkan adalah sebagai berikut:

$$y = u_{11}x_1 + w_{o1} + \sum_{j=1}^5 w_{j1} \left(\frac{1 - e^{-(v_{oj} + \sum_{j=1}^5 v_{1j}x_1)}}{1 + e^{-(v_{oj} + \sum_{j=1}^5 v_{1j}x_1)}} \right)$$

Model tersebut menghasilkan MAPE *training* sebesar 1,8248% dan MAPE *testing* sebesar 2,3136%, dimana nilai MAPE tersebut masih di bawah 10% sehingga dapat dikatakan bahwa model sangat baik untuk dijadikan model prediksi

4. Peramalan harga saham harian PT. XL Axiata Tbk untuk 10 langkah kedepan dengan model terbaik memberikan nilai MAPE sebesar 5,2075% dimana nilai tersebut masih kurang dari 10% sehingga dapat dikatakan keakuratan hasil peramalan sangat baik.

DAFTAR PUSTAKA

- Fausett, L. 1994. *Fundamental of Neural Network*. Prentice Hall. New Jersey.
- Haykin, S. 2009. *Neural Network and Learning Machines*. Third Edition. Pearson Education. New Jersey.
- Kusumadewi, S. 2003. *Artificial Intellegent (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- Makridarkis, S., & Wheelwright, S. C. 1999. *Metode dan Aplikasi Peramalan*. Jakarta: Erlangga.
- Santosa, B., & Jin Ai, T. 2017. *Pengantar Metaheuristik Implementasi dengan Matlab2017*. Surabaya: ITS Tekno Sains.
- Soejoeti, Z. 1987. *Analisis Runtun Waktu*. Jakarta: Karunika.
- Suyanto. 2014. *Algoritma Optimasi Deterministik atau Probabilistik*. Yogyakarta: Graha Ilmu.
- Warsito, B. 2009. *Kapita Selektta Statistika Neural Network*. Semarang: BP Universitas Diponegoro.
- Warsito, B., Santoso, R., Suparti., & Yasin, H. 2018. *Cascade Forward Neural Network for Time Series Prediction*. Journal of Physics Vol. 1025.
- Wei, W. 2006. *Time Analysis Univariate and Multivariate Methods*. 2nd Edition. USA: Addison Wesley Publishing Company, Inc.
- www.finance.yahoo.com (diakses pada tanggal 9 Februari 2019).