

**MODEL FEED FORWARD NEURAL NETWORK (FFNN) DENGAN ALGORITMA
PARTICLE SWARM SEBAGAI OPTIMASI BOBOT
(Studi Kasus : Harga Daging Sapi dari Bank Dunia Periode Januari 2007 – Desember 2018)**

Faisal Fikri Utama¹, Budi Warsito², Sugito³

^{1,2,3} Departemen Statistika FSM Universitas Diponegoro
e-mail : faisalfikri92@gmail.com

ABSTRACT

Beef is one of the important food commodities to fulfill the nutritional adequacy of humans. The World Bank notes the beef prices that are exported worldwide every month. For this reason, those data becomes a predictable series for the next period. Feed Forward Neural Network is a non-parametric method that can be used to make predictions from time series data without having to be bound by classical assumptions. The initiated weight will be evaluated by an algorithm that can minimize errors. Particle Swarm Optimization (PSO) is an optimization algorithm based on particle speed to reach the destination. The FFNN model will be combined with PSO to get predictive results that are close to the target. The best architecture on FFNN is obtained with 2 units of input, 1 unit of bias, 3 hidden units, and 1 unit of output by producing MAPE training 11.7735% and MAPE testing 8.14%. According to Lewis (1982) in Moreno et. al (2013), the MAPE value below 10% is highly accurate forecasting.

Keywords: Feed Forward Neural Network (FFNN), Particle Swarm Optimization (PSO), neurons, weights, predictions.

1. PENDAHULUAN

Pangan merupakan kebutuhan dasar utama bagi manusia yang harus dipenuhi setiap saat. Hak untuk memperoleh pangan merupakan salah satu hak asasi manusia, sebagaimana tersebut dalam pasal 27 UUD 1945 maupun dalam Deklarasi Roma (1996). Pertimbangan tersebut mendasari terbitnya UU No. 7/1996 tentang Pangan. Sebagai kebutuhan dasar dan salah satu hak asasi manusia, pangan mempunyai arti dan peran yang sangat penting bagi kehidupan suatu bangsa. Ketersediaan pangan yang lebih kecil dibandingkan kebutuhannya dapat menciptakan ketidakstabilan ekonomi (Bulog, 2014).

Salah satu sumber data yang kredibel dalam memuat harga komoditas pangan terutama daging sapi adalah Bank Dunia. Harga daging sapi yang tercantum adalah harga produk daging Australia dan Selandia Baru yang diimpor ke banyak negara, sehingga harga produk ini menjadi tolak ukur harga daging sapi dunia. Data runtun waktu ini dirilis setiap bulan sehingga tidak lepas dari prediksi harga pada periode berikutnya untuk memberikan sikap kontrol terhadap perubahan harga daging.

Menurut Wei (2006), runtun waktu adalah deret observasi yang tersusun. Meskipun penyusunan umumnya berdasarkan waktu, terutama dalam selang waktu yang sama. Hasil terpenting yang didapat dari data runtun waktu adalah prediksi pada periode berikutnya berdasarkan syarat-syarat dan asumsi-asumsi yang harus dipenuhi untuk menyelesaikan analisis runtun waktu. Namun terdapat metode yang lebih sederhana dalam membuat satu prediksi terlepas dari asumsi-asumsi klasik yang harus dipenuhi. Salah satu metode yang populer adalah Jaringan Syaraf Tiruan.

Jaringan Syaraf Tiruan (JST) adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan syaraf makhluk hidup yang pertama kali diperkenalkan oleh McCulloch dan Pitts pada 1943 berupa model sederhana dari suatu syaraf nyata dalam otak manusia. Operasi JST pada penelitian berikut akan diterapkan pada kasus runtun waktu, yaitu peramalan rata-rata harga daging sapi dengan satuan Dollar Amerika Serikat per

kilogram dengan data input bulanan selama mulai Januari 2007 hingga Desember 2018. Dari output analisis akan diketahui perubahan harga daging sapi pada selang waktu berikutnya sesuai pembagian sektor pelatihan dan pengujian. Pelatihan adalah simulasi berulang yang dilakukan oleh jaringan untuk mendapatkan bobot optimum sehingga jaringan akan mendapatkan error yang paling minimum. Pengulangan dilakukan dengan menentukan jumlah maksimum iterasi atau minimum error yang harus dicapai. Sementara pengujian adalah proses jaringan menerapkan bobot optimum yang telah didapat sehingga bisa didapatkan hasil prediksi terbaik.

Pada penelitian ini, algoritma yang digunakan adalah algoritma Particle Swarm Optimization (PSO). PSO secara bahasa bermakna optimasi segerombol partikel. Partikel yang dimaksud adalah suatu koloni hewan. Algoritma PSO memiliki model tiruan perilaku hewan yang berhasil dalam berpindah-pindah tempat mencari makanan.

Model JST dan algoritma optimasi PSO berpotensi untuk menghasilkan error yang minim karena iterasi akan berjalan hingga target penghentian tercapai. Bobot jaringan yang telah dibangkitkan secara acak akan dievaluasi dengan PSO menjadi bobot baru agar memperhalus hasil prediksi.

2. TINJAUAN PUSTAKA

2.1 Jaringan Syaraf Tiruan *Model Feed Forward Neural Network (FFNN)*

Menurut Fausset (1994), Jaringan Syaraf Tiruan (JST) adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan syaraf makhluk hidup. JST dikembangkan sebagai model matematika yang menyerupai pola pikir manusia atau jaringan syaraf makhluk hidup, dengan asumsi sebagai berikut (Warsito, 2009).

1. Proses informasi terjadi pada banyak elemen sederhana yang disebut *neuron*.
2. Sinyal yang melewati antar *neuron* menggunakan hubungan tertentu.
3. Setiap penghubung antar *neuron* mempunyai bobot yang bersesuaian, dengan mengalikan sinyal yang dikirim.
4. Setiap *neuron* menggunakan fungsi aktivasi terhadap input (jumlah sinyal terkirim yang dibobot) untuk menentukan sinyal output. dan fluktuasi data yang konstan di sekitar rata-rata.

JST dengan lapisan tunggal memiliki keterbatasan dalam pengenalan pola. Kelemahan ini bisa ditanggulangi dengan menambahkan satu atau beberapa lapisan tersembunyi di antara lapisan input dan output. Meskipun penggunaan *hidden multilayer* atau banyak lapis tersembunyi memiliki banyak manfaat, namun proses pelatihannya membutuhkan waktu yang lebih lama (Siang, 2005).

FFNN menggunakan metode penurunan gradien untuk meminimalkan kuadrat *error* output. Terdapat tiga fase dalam pelatihan jaringan, yaitu fase maju, fase mundur, serta tahap perubahan bobot dan bias. Arsitektur FFNN terdiri dari *input layer*, *hidden layer*, dan *output layer* (Sutojo, *et. al.*, 2011).

2.2 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) secara bahasa bermakna optimasi segerombol partikel. Cara kerja PSO menyerupai perilaku satu koloni atau kawanannya seperti semut, rayap, lebah, dan tawon, kawanannya burung, maupun kawanannya ikan. Partikel yang dimaksud memiliki arti seekor lebah dalam koloninya, atau seekor burung dalam kawanannya terbangnya. Algoritma PSO memiliki model tiruan perilaku hewan-hewan tersebut yang berhasil dalam berpindah-pindah tempat mencari makanan. Setiap individu atau partikel dalam suatu kawanannya atau koloni berperilaku secara tersebar dengan menggunakan kecerdasannya sendiri dan kecerdasan kolektif atau kelompok dari kawanannya atau koloninya. Dengan demikian, jika satu partikel menemukan jalan yang tepat menuju lokasi makanan,

sisa kawanannya juga akan mengikuti jalan yang tepat dengan menemukan cara sendiri meskipun lokasinya jauh (Rao, 2009).

Menurut Cholissodin dan Riyandani (2016), terdapat beberapa komponen dalam algoritma PSO, diantaranya:

1. *Swarm*, merupakan jumlah partikel dalam populasi pada suatu algoritma. Ukuran *swarm* bergantung pada seberapa kompleks masalah yang dihadapi. Secara umum, ukuran *swarm* cenderung lebih kecil jika dibandingkan dengan algoritma *evolusioner* yang lain dalam mencari solusi terbaik.
2. *Partikel*, merupakan individu dalam suatu *swarm* yang merepresentasikan solusi penyelesaian masalah. Setiap partikel memiliki posisi dan kecepatan yang ditentukan oleh representasi solusi pada saat itu.
3. *Personal Best (pBest)*, merupakan posisi terbaik yang pernah dicapai partikel dengan membandingkan *fitness* pada posisi partikel sekarang dan sebelumnya. *Personal Best* dipersiapkan untuk mendapatkan solusi terbaik.
4. *Global Best (gBest)*, merupakan posisi terbaik partikel yang yang diperoleh dengan membandingkan nilai *fitness* terbaik dari keseluruhan partikel dan *swarm*.
5. *Kecepatan (velocity)*, v merupakan vektor yang menentukan arah perpindahan posisi partikel. Perubahan kecepatan dilakukan setiap iterasi dengan tujuan memperbaiki posisi partikel semula.
6. *Bobot Inersia (inertia weight)*, r digunakan untuk mengendalikan dampak dari perubahan kecepatan yang diberikan oleh partikel.
7. *Koefisien Akselerasi*, merupakan faktor pengontrol sejauh mana partikel berpindah dalam suatu iterasi. Secara umum nilai koefisien akselerasi c_1 dan c_2 adalah sama yaitu dalam rentang 0 sampai 4. Namun demikian, nilai tersebut dapat ditentukan sendiri untuk setiap penelitian berbeda.

2.3 Pelatihan FFNN dengan Algoritma PSO

Pelatihan FFNN meliputi 2 fase. Fase pertama adalah fase maju. Pola input dihitung maju mulai dari lapisan input hingga lapisan output menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi.

Algoritma pelatihan dengan PSO untuk jaringan dengan satu lapisan tersembunyi (dengan fungsi aktivasi sigmoid bipolar) adalah sebagai berikut:

Fase I : Propagasi maju

Langkah 1 : Inisialisasi semua bobot w_{ji} dan w_b dengan bilangan acak kecil pada interval 0 – 1. Tetapkan kondisi penghentian dengan nilai target MSE tertentu, misal $\varepsilon = 0,01$

Langkah 2 : Tiap unit input menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya

Langkah 3 : Hitung semua output di unit tersembunyi z_j ($j = 1,2,\dots,p$)

$$z_net_j = w_{j0} + \sum_{i=1}^n x_i w_{ji} \quad (1)$$

$$z_j = f(z_net_j) = \frac{1 - e^{-z_net_j}}{1 + e^{-z_net_j}} \quad (2)$$

Langkah 4 : Hitung output jaringan

$$y_net = v_{j0} + \sum_{j=1}^p z_j v_j \quad (3)$$

dengan

$z_{_netj}$: nilai yang masuk dari unit input ke unit hidden

z_j : nilai unit hidden ke-j

w_{j0} : bobot bias ke hidden

x_i : nilai unit input ke-i

w_{ji} : bobot unit input ke-i menuju unit hidden ke-j

$y_{_net}$: nilai yang masuk dari unit hidden ke unit output

v_{j0} : bobot bias ke output

v_j : bobot unit hidden ke-j menuju unit output

Fase II : Evaluasi Bobot

Pada fase ini, bobot yang telah didapat dan digunakan untuk memprediksi nilai output dievaluasi menggunakan algoritma *Particle Swarm Optimization* (PSO) sebagaimana dimulai dari langkah 5:

Langkah 5 : Gunakan pasangan bobot yang telah dibangkitkan dengan PSO untuk kemudian diambil pasangan bobot-bobot yang memiliki *error* terkecil.

Langkah 6 : cari nilai y untuk masing-masing partikel

Langkah 7 : kuadratkan selisih target dengan $y_{_net}$ untuk mendapatkan rata-rata kuadrat error (MSE).

Langkah 8 : tentukan v_0 , $pbest$, dan $gbest$. Untuk iterasi pertama, nilai $pbest$ akan sama dengan nilai partikelnya. Nilai $gbest$ ditentukan pasangan partikel atau bobot jaringan dengan MSE terkecil.

Langkah 9 : Update kecepatan partikel

$$\delta_j(i + 1) = \theta v_j(i) + c_1 r_1 (Pbest_j(i) - \alpha_j(i)) + c_2 r_2 (Gbest(i) - \alpha_j(i)) \quad (4)$$

Karena terdapat 9 parameter yang diestimasi dengan PSO, maka partikel dibentuk dalam vektor berukuran 1×9 : [v_{j0} w_{11} ... v_1 v_2]

Langkah 10 : Update posisi partikel baru

$$\alpha_j(i + 1) = \delta_j(i + 1) + \alpha_j(i) \quad (5)$$

Langkah 11 : Hitung MSE yang bersesuaian dengan bobot baru $\alpha_j(i+1)$. Jika MSE sudah mencapai *error* yang senilai dengan ϵ , iterasi akan dihentikan. Jika belum mencapai target, ulangi dari langkah 2.

2.5 Ukuran Kesalahan Peramalan

Menurut Chiulli (1999), pengukuran kesalahan peramalan perlu dilakukan secara formal. Pengukuran semacam itu berfokus pada besaran dan tanda (positif atau negatif) antara data aktual dan data peramalan di setiap periodenya. Beberapa ukuran kesalahan yang digunakan dalam menentukan ketepatan data peramalan adalah:

1. Mean Absolute Deviation

$$MAD = \frac{\sum_{i=1}^n |A_i - F_i|}{N} \quad (6)$$

2. Mean Absolute Percentage Error

$$MAPE = \frac{1}{N} \sum_{i=1}^n \left| \frac{A_i - F_i}{A_i} \right| \times 100\% \quad (7)$$

3. Mean Square Error

$$MSE = \frac{1}{N} \sum_{i=1}^n (A_i - F_i)^2 \quad (8)$$

dimana,

A_i : data aktual / target ke- i

F_i : data peramalan ke- i

N : jumlah data

3. METODE PENELITIAN

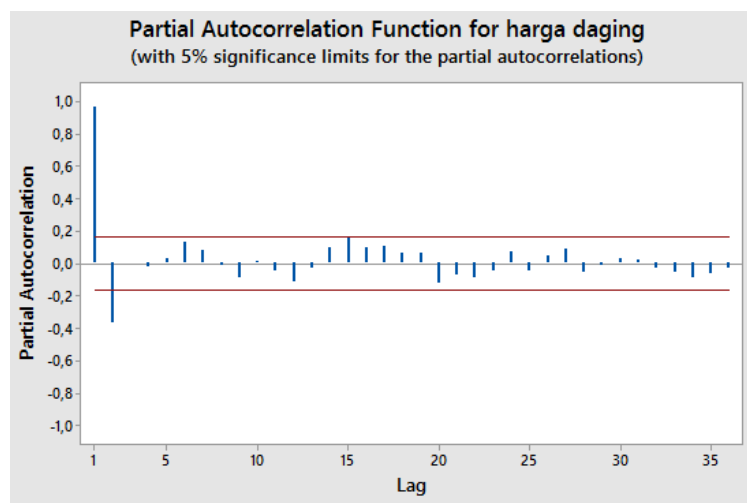
3.1. Data Penelitian

Data yang digunakan pada penelitian ini adalah data rata-rata bulanan harga komoditas daging sapi dengan satuan dollar Amerika Serikat per kilogram selama 12 tahun. Data tersebut merupakan data sekunder yang diperoleh dari laporan rutin situs resmi bank dunia sebanyak 144 data. Variabel yang digunakan dalam penelitian ini adalah rata-rata harga daging sapi dunia per bulan terhitung sejak Januari 2007 hingga Desember 2018.

3.2. Tahapan Analisis Data

Software yang digunakan dalam penelitian berikut adalah MATLAB r2012b dan Minitab 17. Tahapan dalam analisis data yang diperlukan adalah sebagai berikut:

1. Menentukan data input, target, dan tingkat MSE yang ingin dicapai. Data target dalam peramalan adalah data pada periode waktu berikutnya berdasarkan data periode sebelumnya.
2. Membentuk model jaringan feed forward dengan neuron input sebanyak jumlah lag PACF, neuron hidden layer ditentukan dengan simulasi, dan 1 neuron pada lapisan output.
3. Analisis pada jaringan syaraf tiruan memerlukan partisi data untuk training dan testing. Dari 144 data, 80% akan dialokasikan untuk dilatih atau disimulasikan dengan algoritma optimasi dan 20% dialokasikan untuk diuji.
4. Inisialisasi bobot secara acak untuk diaplikasikan pada perhitungan model feed forward dari lapisan input ke lapisan tersembunyi hingga dari lapisan tersembunyi ke lapisan output.
5. Evaluasi bobot yang telah digunakan dengan membangkitkan lagi bilangan-bilangan acak dengan algoritma PSO. Bobot-bobot acak yang baru selanjutnya disebut partikel. Algoritma PSO akan membantu mencari bobot yang dapat menghasilkan nilai Mean Square Error (MSE) terkecil bagi nilai output y .
6. Input kembali sisa data yang ada sebanyak 20% untuk diuji. Jaringan akan memprediksi nilai-nilai output berdasarkan simulasi yang telah dijalankan sebelumnya dengan parameter hasil pelatihan dengan bobot akhir yang sudah dievaluasi algoritma PSO.



Gambar 1. Plot Autokorelasi Parsial

4. HASIL DAN PEMBAHASAN

4.1 Penentuan Variabel Input

Pada pemeriksaan plot PACF, terdapat 2 lag yang keluar yaitu x_{t-1} dan x_{t-2} karena memiliki nilai autokorelasi terbesar. Kedua variabel ini akan menjadi input FFNN – PSO. Kemudian data akan dibagi menjadi 2 yaitu data training dan data testing. Dari 144 data tersebut, 80% akan dilibatkan proses training dan 20% akan diuji.

4.2 Arsitektur Feed Forward Neural Network (FFNN)

Pada penelitian ini, arsitektur jaringan terdiri dari 3 lapisan dengan neuron pada setiap, yaitu lapisan input, lapisan tersembunyi, dan lapisan output. Lapisan input terdiri dari 2 neuron yang menandakan 2 variabel input yaitu x_{t-1} dan x_{t-2} , 1 unit neuron bias, 1 unit neuron output, dan 1 hidden neuron yang ditentukan dengan uji coba dari 1 unit sampai dengan n unit sehingga hasil paling optimum didapat, atau peneliti menghentikan ujicoba di angka tertentu. (Panchal dan Panchal, 2014).

Fungsi aktivasi yang digunakan dalam arsitektur jaringan juga menentukan hasil prediksi. Jaringan pada penelitian ini menggunakan fungsi aktivasi untuk menjalankan transfer sinyal dari lapisan input ke lapisan tersembunyi adalah logistik sigmoid bipolar (tansig), dan sinyal diteruskan dari lapisan tersembunyi ke lapisan output menggunakan fungsi identitas (purelin).

Penentuan jumlah *node* pada lapisan tersembunyi (*hidden layer*) dilakukan dengan simulasi berulang dari 1 unit hingga 15 unit. Berdasarkan masing-masing hasil uji coba, hasil simulasi dicatat berupa MSE *training*, MSE *testing*, dan koefisien determinasi (R^2). MSE training dan testing memperlihatkan MSE yang diperoleh pada proses pembelajaran dan pelatihan, sementara koefisien determinasi menunjukkan besarnya pengaruh variabel x_{t-1} dan x_{t-2} terhadap nilai prediksi.

Jumlah unit hidden neuron akan ditentukan berdasarkan kriteria MSE testing yang paling optimum pada masing-masing kriteria. Nilai MSE testing terkecil terdapat pada hidden neuron 3 unit. Selanjutnya akan dibandingkan hasil prediksi pada arsitektur jaringan.

4.3 Analisis Hasil Pelatihan FFNN – PSO

Pada penelitian ini neuron input yang digunakan jaringan sebanyak 2 unit, berdasarkan lag *Partial Autocorrelation Function (PACF)* yang diperiksa terhadap data. Hidden neuron ditentukan dengan metode *trial and error*, yaitu membuat simulasi jaringan dengan 1 hidden neuron kemudian jumlah *node* ditambah menjadi 2, 3, dan seterusnya dengan pencatatan kriteria penghentian percobaan yaitu MSE dan MAPE.

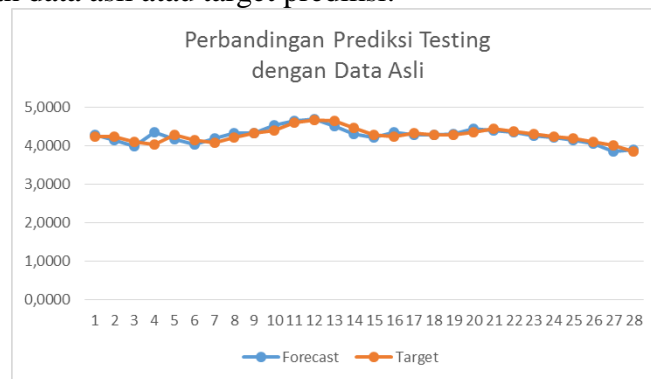
Tabel 1. Hasil Simulasi Jaringan 1 – 15 Hidden Neuron

Hidden Neuron	MSE Train	MSE Test	MAPE Train	MAPE Test
1	0,0764	0,038	19,5259	16,7453
2	0,0488	0,022	15,8558	12,9219
3	0,0273	0,0108	11,6977	8,3631
4	0,0321	0,0297	12,7944	8,8104
5	0,0319	0,0119	14,0639	9,1273
6	0,0386	0,0249	16,6713	13,58
7	0,0389	0,0282	15,5733	13,1744
8	0,0461	0,0279	16,7981	13,2235
9	0,0439	0,0127	17,2791	9,4747
10	0,0284	0,023	12,3492	12,3853
11	0,0333	0,0256	14,7207	12,7576

Hidden Neuron	MSE Train	MSE Test	MAPE Train	MAPE Test
12	0,0398	0,0151	15,4708	9,1014
13	0,0812	0,0374	21,3321	16,5136
14	0,0359	0,0192	15,7211	12,1173
15	0,0297	0,0209	12,1991	12,2305

4.5 Penetapan Output

Hasil yang diharapkan pada fase ini adalah terbentuknya suatu nilai prediksi harga daging sapi dunia berdasarkan harga pada 2 periode sebelumnya. Setelah jaringan dilatih dan mendapatkan bobot optimum, *testing set* diadaptasikan ke dalam jaringan dengan bobot optimum hasil pembelajaran. Kemudian akan muncul nilai prediksi harga daging sapi y pada periode berikutnya (\hat{y}_{t+1}). Untuk memprediksi data pada periode \hat{y}_{t+2} , gunakan data pada periode sebelumnya untuk diuji pada jaringan. Berikut ini merupakan perbandingan data prediksi *testing* dengan data asli atau target prediksi:



Gambar 2. Perbandingan Data *Testing* dengan Data Target

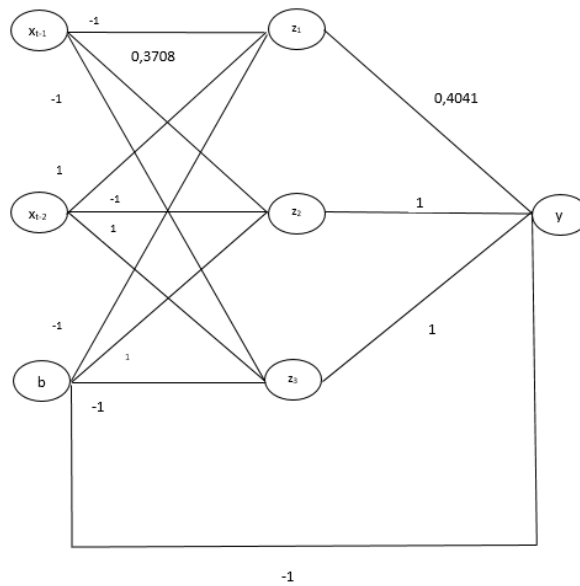
MSE yang dihasilkan pada proses *training* dan *testing* adalah 0,0273 dan 0,0108. Sementara MAPE training dan testing yang dihasilkan adalah 11,6977% dan 8,3631%. Lewis (1982) dalam Moreno *et. al* (2013) menggambarkan karakteristik nilai MAPE dalam peramalan seperti pada Tabel 2

Tabel 2. Karakteristik Nilai MAPE

MAPE (%)	Interpretasi Akurasi Peramalan
< 10	Sangat Baik
10 – 20	Baik
20 - 50	Cukup Baik
>50	Kurang

4.6 Prediksi Harga Daging Sapi

Berdasarkan pelatihan jaringan pada sub-bab sebelumnya, diperoleh arsitektur jaringan dengan konfigurasi 1 neuron pada lapisan input, 1 neuron pada lapisan tersembunyi, dan 1 neuron pada lapisan output, dengan 1 neuron bias yang mengirim sinyal ke lapisan tersembunyi dan lapisan output. Secara visual, arsitektur jaringan yang terbentuk seperti pada Gambar 3:



Gambar 3. Arsitektur *PSO-FFNN* Prediksi Harga Daging Sapi

Model jaringan tersebut mengirim sinyal antar lapisan dengan fungsi aktivasi:

$$\begin{aligned}
 y &= \Psi_0 \left(v_{k.0} + \sum_{j=1}^3 v_j \Psi_n \left(w_{j.0} + \sum_{i=1}^2 w_{ji} x_i \right) \right) \\
 &= \Psi_0 \left(v_{k.0} + \sum_{j=1}^3 v_j \Psi_n (w_{1.0} + w_{11}x_{t-1} + w_{12}x_{t-2}) + (w_{2.0} + w_{21}x_{t-1} + w_{22}x_{t-2}) \right. \\
 &\quad \left. + (w_{3.0} + w_{31}x_{t-1} + w_{32}x_{t-2}) \right) \\
 &= -1 + \frac{v_1(1 - e^{-(-1-1,6292x_{t-1}+x_{t-2})})}{1 + e^{-(-1-1,6292x_{t-1}+x_{t-2})}} + \frac{v_2(1 - e^{-(-1-1,6292x_{t-1}+x_{t-2})})}{1 + e^{-(-1-1,6292x_{t-1}+x_{t-2})}} \\
 &\quad + \frac{v_3(1 - e^{-(-1-1,6292x_{t-1}+x_{t-2})})}{1 + e^{-(-1-1,6292x_{t-1}+x_{t-2})}} \\
 &= -1 + \frac{0,4041(1 - e^{-(-1-1,6292x_{t-1}+x_{t-2})})}{1 + e^{-(-1-1,6292x_{t-1}+x_{t-2})}} + \frac{(1 - e^{-(-1-1,6292x_{t-1}+x_{t-2})})}{1 + e^{-(-1-1,6292x_{t-1}+x_{t-2})}} \\
 &\quad + \frac{(1 - e^{-(-1-1,6292x_{t-1}+x_{t-2})})}{1 + e^{-(-1-1,6292x_{t-1}+x_{t-2})}} \\
 &= -1 + \frac{0,4041(1 - e^{(1+1,6292x_{t-1}-x_{t-2})})}{1 + e^{(1+1,6292x_{t-1}-x_{t-2})}} + 2 \left(\frac{1 - e^{(1+1,6292x_{t-1}-x_{t-2})}}{1 + e^{(1+1,6292x_{t-1}-x_{t-2})}} \right) \\
 &= -1 + \frac{2,4041(1 - e^{(1+1,6292x_{t-1}-x_{t-2})})}{1 + e^{(1+1,6292x_{t-1}-x_{t-2})}}
 \end{aligned}$$

dengan:

Ψ_0 : fungsi identitas $\Psi_0(x) = x$

Ψ_n : fungsi aktivasi sigmoid bipolar $\Psi_n(x) = \frac{1-e^{-x}}{1+e^{-x}}$

$v_{k.0}$: bobot bias menuju lapisan output

v_j : bobot hidden layer menuju output`

$w_{j,0}$: bobot bias menuju hidden layer

w_{ji} : bobot input menuju hidden layer

Kemudian dapat diambil hasil prediksi selama 6 periode berikutnya yaitu Januari 2019 – Juni 2019 sebagaimana pada Tabel 3 berikut:

Tabel 3. Prediksi Harga Daging Sapi

Periode	Prediksi Harga (/kg)
Januari 2019	\$3,9098
Februari 2019	\$4,2865
Maret 2019	\$3,8395
April 2019	\$4,4380
Mei 2019	\$3,7258
Juni 2019	\$4,6873

5. KESIMPULAN

Berdasarkan plot autokorelasi parsial (PACF) data harga daging sapi dunia, ditentukan unit neuron input sebanyak 2 karena autokorelasi terbesar terjadi hingga lag-2, artinya harga pada bulan ini dipengaruhi harga pada 2 bulan sebelumnya. Model FFNN-PSO dengan 2 unit neuron input dan 3 unit hidden adalah $y = -1 + \frac{2,4041(1-e^{(1+1,6292x_{t-1}-x_{t-2})})}{1+e^{(1+1,6292x_{t-1}-x_{t-2})}}$. FFNN – PSO mampu membuat prediksi \$3,9/kg untuk Januari 2019 dan data aktual \$4,2/kg. MSE yang dihasilkan pada proses *training* dan *testing* adalah 0,0273 dan 0,0108. Sementara MAPE training dan testing yang dihasilkan adalah 11,6977% dan 8,3631% yang berarti hasil peramalan memiliki akurasi yang tinggi terhadap target.

DAFTAR PUSTAKA

- Brockwell, P. J., R.A. Davis. 1990. *Time Series: Theory and Methods, Second Edition*. New York: Springer-Verlag.
- Bulog. 2014. Ketahanan Pangan di <http://www.bulog.co.id/ketahananpangan.php> (diakses 26 Januari 2018)
- Chiulli, R. M. 1999. *Quantitive Analysis: An Introduction*. Amsterdam: Gordon and Breach Science Publishers.
- Cholisoddin, I., E. Riyandani. 2016. *Swarm Intelligence (Teori & Case Study)*. Malang: Fakultas Ilmu Komputer Universitas Brawijaya.
- Fausett, L. 1994. *Fundamentals of Neural Networks, Architecture, Algorithms, and Applications*. Prentice Hall.
- Moreno, J.J.M., Pol, A.P., Abad, A.S., & Blasco, B.C. (2013). Using The R-MAPE Index as a Resistant Measure of Forecast Accuracy. *Psicothema*. 25(4), 500-506. doi:10.7334/psicothema2013.23
- Panchal, F.S., & Panchal, M. (2014). Review Methods of Selecting Numbers of Hidden Nodes in Artificial Neural Network. *International Journal of Computer Science and Mobile Computing*, 3, 455-464.
- Rao, S.S. 2009. *Engineering Optimization: Theory and Practice, Fourth Editon*. John Wiley & Sons, Inc.

- Siang, J.J. 2005. *Jaringan Syaraf Tiruan dan Pemrogramannya menggunakan MATLAB*. Yogyakarta: Penerbit ANDI.
- Sutojo, T., E. Mulyanto., V. Suhartono. 2011. *Kecerdasan Buatan*. Yogyakarta: Penerbit ANDI.
- Warsito, B. 2009. *Kapita Selekta Statistika Neural Network*. Semarang: BP Undip Semarang.
- Wei, W.W.S. 2006. *Time Series Analysis Univariate and Multivariate Methods, Second Edition*. Pearson Education, Inc.