

**MODEL ARTIFICIAL NEURAL NETWORK
BERBASIS PARTICLE SWARM OPTIMIZATION UNTUK
PREDIKSI LAJU INFLASI**

Joko S. Dwi Raharjo

Program Studi Teknik Informatika

Sekolah Tinggi Manajemen Informatika dan Komputer Eresha Jakarta

E-Mail: joko.sdrh@gmail.com

Abstrak

Prediksi laju inflasi sangat diperlukan oleh pembuat kebijakan, investor maupun perusahaan untuk merencanakan strategi ekonomi dalam mengantisipasi perkembangan laju inflasi dan perencanaan keuangan ke depan. Prediksi laju inflasi yang akurat akan memberikan kontribusi pengambilan keputusan yang tepat. Banyak pemodelan digunakan oleh peneliti untuk mendapatkan akurasi prediksi terbaik dan yang paling umum adalah model ekonometri (misalnya AR, MA, ARIMA dll) namun dalam perkembangan, model *artificial neural network* (ANN) mulai banyak dimanfaatkan karena terbukti bahwa model ANN memiliki akurasi yang lebih baik dibanding model ekonometri khususnya dalam prediksi laju inflasi. Keandalan ANN lebih lanjut dikembangkan oleh beberapa peneliti melalui integrasi dengan model lain, salah satunya intergrasi antara ANN yang dioptimasi dengan *particle swarm optimization* (PSO). Pengintegrasian ini digunakan untuk saling mengatasi kelemahan dan meningkatkan kelebihan pada masing-masing model sehingga diperoleh hasil pengukuran yang lebih baik. Pengujian kemampuan pada penelitian ini, menguji kemampuan model *artificial neural network* (ANN) yang diintegrasikan dengan *particle swarm optimization* (PSO) berdasarkan *attribute weight* atau pembobotan atribut yang selanjutnya disebut awPSO-ANN. Hasil pengujian prediksi laju inflasi menunjukkan bahwa secara umum awPSO-ANN memberikan nilai RMSE yang lebih baik (0.157) dibandingkan ANN sebelum di optimasi (0.181)

Kata kunci: Prediksi, Inflasi, *Artificial neural network*, *Particle swarm optimization*, *attribute weight*..

Abstract

Prediction of inflation is needed by policy makers, investors and companies to plan economic strategies in anticipation of the inflation rate and financial planning in the future. Accurate prediction of the inflation rate will contribute to making the right decision. Many modeling used by researchers to obtain the best prediction accuracy is the most common and econometric models (eg AR, MA, ARIMA, etc.) but in the development, models of artificial neural network (ANN) from widely used because it proved that ANN models have better accuracy econometric models, especially compared to the predictions of inflation. ANN reliability further developed by several researchers through integration with other models, one of which integration between the ANN is optimized by particle swarm optimization (PSO). This integration is used to overcome weaknesses and improve the mutual advantages to each model in order to obtain a better measurement results. Testing capabilities in research, testing the ability of artificial neural network models (ANN), integrated with particle swarm optimization (PSO) based on the weight attribute or attribute weighting, hereinafter referred to awPSO-ANN. Test results show that the prediction rate of general inflation awPSO-ANN gives better RMSE value (0.157) compared ANN before the optimization (0.181)

Key words: Prediction, Inflation, *Artificial neural network*, *Particle swarm optimization*, *attribute weight*.

PENDAHULUAN

Inflasi tidak dapat dihindarkan oleh setiap negara dan menjadi indikator pertumbuhan ekonomi. Kondisi tinggi rendahnya tingkat inflasi akan mengindikasikan baik-buruknya pertumbuhan ekonomi suatu negara. Inflasi dapat menyebabkan ketidakpastian tentang masa depan harga, suku bunga, nilai tukar dan kemungkinan berakibat pada meningkatnya potensi risiko antara pedagang dan mitra suatu negara [1].

Inflasi tergolong dalam timeseries dan biasanya akan terulang kembali berdasarkan runtut waktu, lima tahun kemudian, atau sepuluh tahun kemudian. Oleh karena itu, permasalahan inflasi dapat dikaji dan dianalisa dengan menggunakan pendekatan-pendekatan yang ada dalam time series yang secara umum memanfaatkan metode *statistic econometric* antara lain *Moving Average* (MA), *Autoregressive* (AR), *Exponential Smoothing* (ES), *Autoregressive Integrated Moving Average* (ARIMA) atau *Self Exiting Autoregressive* (SETAR). Perkembangan terakhir pemodelan prediksi dengan menggunakan *Artificial Neural Network* (ANN) merupakan salah satu pemodelan yang banyak digunakan dalam prediksi ekonomi, keuangan, bisnis dan industri.

Kesimpulan beberapa penelitian tentang prediksi laju inflasi menunjukkan bahwa model ANN memberikan akurasi yang lebih baik dibanding model ekonometri [2] [3] [4] [5] [6] [7] [8] [9].

Kemampuan ANN dalam pembelajaran terhadap data training dilakukan melalui back propagation, namun ada beberapa kelemahan antara lain laju konvergensi yang lambat [10] dan terjebak dalam minimum lokal [11]. Untuk mengatasi permasalahan ada beberapa pendekatan optimasi seperti *genetik algoritma* (GA), *ant colony optimization* (ACO), dan *particle swarm optimization* (PSO) [12].

Penelitian terdahulu membuktikan bahwa secara individu PSO memiliki kinerja terbaik dibanding ACO dan GA [13] [14] [15]. Kemudian dikembangkan model hybrid PSO-ANN untuk menguji beberapa data klasifikasi [11] [16] [17] [18] yang hasilnya rata-rata cukup efektif dalam konvergensi dan ruang global minimum.

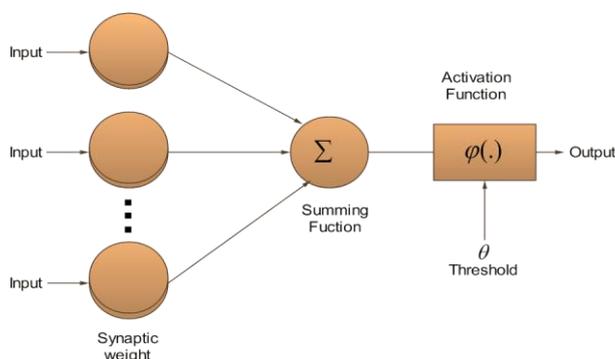
Pada pembahasan ini, menguji tingkat akurasi penerapan model PSO-ANN melalui *attribute weight* (aw) atau awPSO-ANN dengan obyek data inflasi bulanan Indonesia dari Januari 1979- Mei 2011.

ARTIFICIAL NEURAL NETWORK

Artificial Neural Network (ANN) merupakan model penalaran yang didasarkan pada otak manusia. ANN terdiri dari sejumlah prosesor sangat sederhana dan saling berhubungan yang disebut *neuron*. Neuron yang terhubung dengan pembobotan (*weight*) melewati sinyal dari neuron satu ke neuron yang lain.

ANN mampu mengenali sinyal input yang agak berbeda dari yang pernah diterima sebelumnya dan mampu bekerja meskipun beberapa neuronnya tidak mampu bekerja dengan baik. Jika sebuah neuron rusak maka neuron lain dapat dilatih untuk menggantikan fungsi neuron yang rusak tersebut.

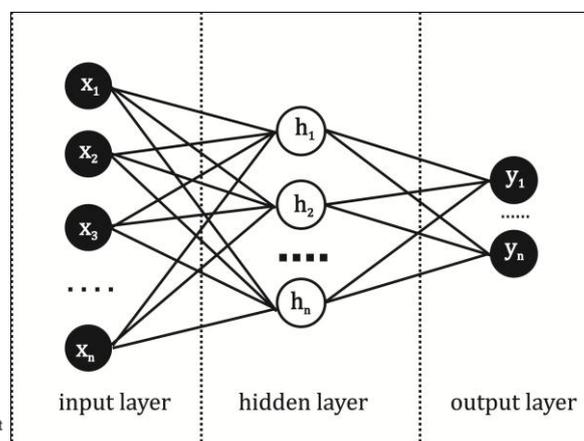
Model neuron ANN terdiri; fungsi penjumlahan (*summing function*), fungsi aktivasi (*activation function*), dan keluaran (*output*).



Gambar 1. Model Neuron

Cara kerja ANN adalah informasi yang masuk (*input*) akan dikirim ke neuron dengan bobot tertentu dan selanjutnya diproses oleh suatu fungsi yang akan menjumlahkan nilai-nilai bobot yang ada. Hasil penjumlahan akan dibandingkan dengan nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Jika input melewati suatu nilai ambang tertentu, maka neuron akan diaktifkan, jika tidak, maka neuron tidak akan diaktifkan. Neuron yang diaktifkan akan mengirimkan output melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya.

Struktur ANN terdiri dari layer *input layer*, *hidden layer* dan *output layer* seperti gambar 2.



Gambar 2. Struktur ANN

Dimana , *input layer* ($x_1, x_2, x_3, \dots, x_n$), terdiri dari unit-unit *input*. Unit-unit input menerima pola inputan data dari luar yang menggambarkan suatu

permasalahan. *Hidden Layer* (h_1, h_2, \dots, h_n), terdiri dari unit-unit tersembunyi. Dimana outputnya tidak dapat secara langsung diamati. Sedangkan *Output Layer* (y_1, \dots, y_n) terdiri dari unit-unit *output* yang merupakan solusi ANN terhadap suatu permasalahan.

BACK PROPAGATION

Model ANN umumnya menggunakan *Algoritma backpropagation* (BP) yang merupakan pengembangan dari algoritma *least mean square* untuk melatih jaringan dengan beberapa layer. BP menggunakan pendekatan algoritma *steepest descent* dengan performance indexnya adalah *mean square error* (MSE)

Dalam melatih jaringan diperlukan seperangkat pasangan data seperti berikut :

$$\{p_1, t_1\}, \{p_2, t_2\} \rightarrow \{p_n, t_n\} \quad (1)$$

dimana p_n adalah nilai input ke-n jaringan dan t_n adalah target, yaitu nilai output yang seharusnya dihasilkan. Algoritma yang akan mengatur atau menyesuaikan parameter-parameter jaringan dalam meminimalkan *mean square error*, yaitu :

$$O_k = f(\text{net}_k) = f(\sum_k W_{jk} O_j + \theta_k) \quad (2)$$

mana O_k adalah output dari neuron k, O_j adalah output dari neuron j pada lapisan yang lebih rendah, W_{jk} adalah bobot antara neuron k dan j, net_k merupakan net input

yang masuk ke neuron k dari lapisan yang lebih rendah, θ_k adalah bias untuk satuan k dan $f(\dots)$ adalah fungsi aktivasi neuron.

Kemudian fungsi kesalahan minimal yang digunakan BP pada lapisan keluaran adalah jumlah kuadrat rata-rata kesalahan E, didefinisikan:

$$\text{Error, } E = \frac{1}{2} \sum_k (t_k - o_k)^2 \quad (3)$$

dimana t_k adalah target output dan O_k adalah hasil yang aktual dari neuron.

Berdasar penurunan gradien, maka setiap perubahan bobot harus sebanding dengan gradien negatif dari penyimpangan terhadap bobot tertentu.

$$\Delta W = -\eta \frac{\partial E}{\partial W} \quad (4)$$

dimana η adalah konstanta yang mewakili *learning rate*. Semakin besar η , semakin besar perubahan bobot, sehingga semakin cepat bobot yang diinginkan ditemukan. Tetapi jika η terlalu besar, akan dapat menyebabkan osilasi [19]

Deferensiasi dari fungsi E merupakan penyimpangan terhadap bobot W dalam persamaan di atas sebanding dengan turunan pertama dari fungsi aktivasi, yaitu:

$$\frac{\partial E}{\partial W} \propto f'(\text{net}) \quad (5)$$

Perubahan bobot paling mudah dipahami dengan membedakan antara komponen kesalahan, $-\partial E/\partial \text{net}$, dan $\partial \text{net}/\partial W$.

Dengan demikian kesalahan untuk output neuron adalah :

$$\begin{aligned}\delta_k &= -\frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial net_k} \\ &= (t_k - o_k) f'(o_k)\end{aligned}\quad (6)$$

dan untuk hidden neuron adalah :

$$\begin{aligned}\delta_k &= \left(-\frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial o_j} \right) \frac{\partial o_j}{\partial net_j} \\ &= \sum_k \delta_k W_{jk} f'(o_j)\end{aligned}\quad (7)$$

Untuk polinomial orde pertama, $\partial net / \partial W$ sama dengan aktivasi input. Perubahan beban untuk beban output secara sederhana adalah :

$$\Delta W_{jk} = \eta \delta_k o_j \quad (8)$$

Sedangkan untuk beban input, adalah :

$$\Delta W_{ij} = \eta \delta_j o_i \quad (9)$$

Rumelhart [20] mengusulkan penambahan masa waktu yang disebut momentum yang membantu untuk meningkatkan learning rate tanpa menyebabkan osilasi sehingga dengan penambahan masa momentum dan subskrip waktu, perubahan beban untuk beban output menjadi:

$$\Delta W_{ij}(n+1) = \eta \delta_k o_i + \alpha \Delta W_{jk}(n) \quad (10)$$

Sedangkan perubahan beban input menjadi :

$$\Delta W_{ij}(n+1) = \eta \delta_j o_i + \alpha \Delta W_{ij}(n) \quad (11)$$

Jelas, bahwa α (momentum) adalah konstanta yang menentukan efek dari

perubahan beban terakhir pada arah gerakan.

PARTICEL SWARM OPTIMIZATION

Particle swarm optimization (PSO) diperkenalkan oleh Eberhart dan Kennedy pada tahun 1995 sebagai salah satu metode optimasi. *Particle Swarm Optimization* terinspirasi pada perilaku sosial koloni hewan seperti rayap, lebah, ikan atau burung. Perilaku sosial ini terdiri dari tindakan individu dan pengaruh dari individu-individu lain dalam suatu kelompok. Setiap individu berperilaku secara terdistribusi dengan menggunakan kecerdasannya (*intelligence*) sendiri dan juga dipengaruhi kelompok kolektifnya. Misalnya, jika seekor burung menemukan jalan yang tepat (*optimal*) menuju sumber makanan, maka sisa anggota kelompok yang lainnya juga akan mengikuti jalan tersebut meskipun lokasi mereka didalam kelompok tersebut tidak saling berdekatan.

PSO dikembangkan berdasarkan pada model tersebut, dimana ketika partikel mendekati target minimum atau maksimum dari fungsi tujuan, partikel tersebut akan secara cepat mengirim informasi kepada partikel yang lainnya dalam kawanan tertentu sehingga partikel yang lain akan mengikuti arah menuju ke target nilai tersebut tetapi tidak secara langsung.

Ada satu komponen yang bergantung pada memori dari setiap partikel yaitu memori tentang apa yang sudah dilalui sebelumnya. Jika jalur tersebut pernah dilalui dan ternyata tidak menghasilkan nilai sesuai fungsi tujuan maka jalur tersebut tidak akan diikuti.

Model ini disimulasikan dalam ruang berdimensi tertentu dengan sejumlah iterasi yang menyebabkan posisi akan semakin mengarah ke target nilai fungsi tujuan. Dalam setiap iterasi, partikel akan menyimpan solusi terbaiknya yang disebut sebagai *pbest* (*personal best*). Sedangkan solusi terbaik untuk semua partikel disebut sebagai *gbest* (*global best*) juga akan selalu digunakan. Proses pergantian posisi partikel didasarkan pada kecepatan (*velocity*) masing-masing partikel menuju *pbest* dan *gbest* [21].

Dalam PSO, pencarian solusi optimal akan dilakukan sampai semua partikel memiliki skema solusi yang sama atau ketika iterasi maksimum sudah tercapai. Formulasi untuk *update* kecepatan dan posisi pada iterasi selanjutnya adalah sebagai berikut:

$$V_{k+1} = V_k + c_1 \cdot R_1 \cdot (Pbest_k - X_k) + c_2 \cdot R_2 \cdot (Gbest_k - X_k) \quad (12)$$

$$X_{k+1} = X_k + V_{k+1} \quad (13)$$

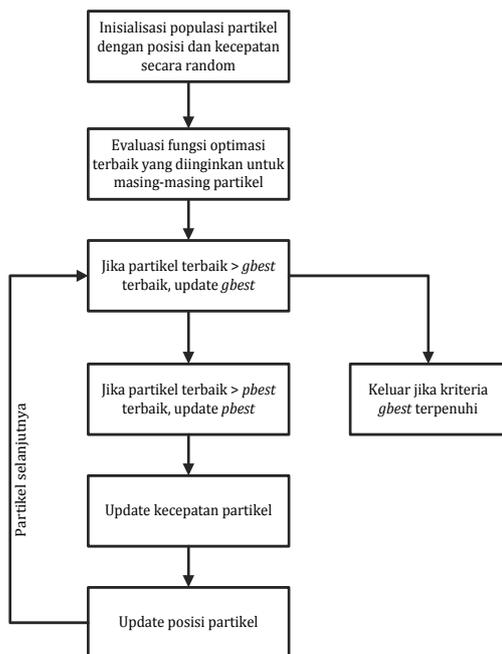
dengan, $k : 1, 2, 3, \dots, N$; V_{k+1} : kecepatan untuk posisi yang dituju; V_k : kecepatan

untuk posisi sekarang (awal); $Pbest_k$: *Personal best*; $Gbest_k$: *Global best*; c_1, c_2 : *learning rates*. Untuk kemampuan individu dan pengaruh sosial group yang menunjukkan bobot dari memori sebuah partikel terhadap memori dari kelompoknya, nilai dari c_1, c_2 biasanya merupakan konstanta dengan nilai $c_1 = 2$ dan $c_2 = 2$; R_1, R_2 : angka acak antara 0 dan 1; X_{k+1} : Posisi yang dituju; X_k : Posisi sekarang.

Selanjutnya, konsep bobot inersia (*inertia weight*) dikembangkan oleh Shi dan Eberhart pada tahun 1998 [22] untuk pengendalian yang lebih baik pada eksplorasi dan eksploitasi. Motivasinya adalah untuk dapat menghilangkan kebutuhan V_{max} . Sehingga *update* persamaan kecepatan menjadi:

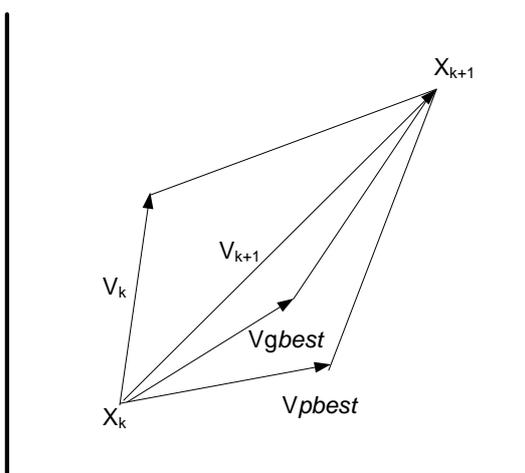
$$V_{k+1} = \omega \cdot V_k + c_1 \cdot R_1 \cdot (Pbest_k - X_k) + c_2 \cdot R_2 \cdot (Gbest_k - X_k) \quad (14)$$

Berikut dasar prosedur PSO :



Gambar 3. Dasar Prosedur PSO

Gambar 4, menunjukkan konsep modifikasi penelusuran tujuan dijelaskan oleh persamaan (12) dan (13), dimana V_{pbest} dan V_{gbest} masing-masing direpresentasikan oleh komponen persamaan (12) yaitu $c_1 \cdot R_1 \cdot (Pbest_k - X_k)$ dan $c_2 \cdot R_2 \cdot (Gbest_k - X_k)$.



Gambar 4. Konsep Modifikasi Penelusuran Tujuan

awPSO-ANN ALGORITMA

Langkah-langkah proses pembelajaran awPSO-ANN:

Langkah 1: Proses pembelajaran awPSO-ANN diinisialisasi dengan sekelompok partikel acak yang ditugaskan pada posisi acak PSO (bobot dan bias).

Langkah 2: awPSO-ANN ini dilatih menggunakan posisi partikel awal

Langkah 3: Kemudian, feedforward ANN dalam awPSO ANN akan menghasilkan kesalahan pembelajaran (partikel *fitness*) berdasarkan bobot awal dan bias.

Langkah 4: Kesalahan pembelajaran pada saat iterasi berlangsung akan dikurangi oleh perubahan posisi partikel, yang akan memperbarui bobot dan bias network. Nilai "pbest" dan nilai "gbest" yang diterapkan pada persamaan melakukan update kecepatan (persamaan 14) untuk menghasilkan nilai pada penyesuaian posisi untuk solusi terbaik atau target kesalahan pembelajaran

Langkah 5: Penetapan posisi baru (bobot dan bias NN) yang dihasilkan dengan menambahkan nilai kecepatan terhitung untuk posisi berjalan digunakan persamaan perpindahan (persamaan 12). Kemudian, penetapan posisi baru yang digunakan untuk menghasilkan kesalahan pembelajaran baru pada *feedforward* NN.

Langkah 6: Proses ini diulang sampai kondisi berhenti, baik kesalahan belajar

minimum atau jumlah iterasi maksimum terpenuhi. Optimasi output, yang merupakan solusi untuk permasalahan optimasi, didasarkan pada nilai posisi gbest.

Adapun pseudocode prosedur awPSO-ANN dalam pelatihan ini adalah :

```

For each particle
Initialize particle for NN problem
End
Do
For each particle
Calculate fitness value (feedforward error
or MSE in NN)
If the fitness value is better than the best
fitness value (Pbest) in history
Then set current value as the new
Pbest
End
Choose the particle with the best fitness
value of all the particles as the Gbest
For each particle
Calculate particle velocity
Update particle position (NN weight)
End

```

PARAMETER awPSO-ANN

Pengujian model awPSO-ANN parameter yang digunakan antara lain :

Parameter ANN

Parameter neural network pada data training penelitian ini adalah :

Table 1. Parameter ANN

Parameter	Nilai
Learning rates	0.1 .. 1
Momentum	0.1 .. 1
Training cycle	500 (default)
Hidden layer	1

Parameter awPSO

Parameter *weight pbest* (c1), *weight gbest* (c2) dan population size yang digunakan dalam penelitian ini mengacu pada penelitian Abdul Hamed [11] sedangkan parameter pembobotan (*inertia weight*, *max weight* dan *min weight*) menggunakan default data pada alat pengujian rapidminer ditunjukkan dalam Tabel 2.

Table 2. Parameter PSO

Parameter	Nilai
c1 (weight pbest)	2.0
c2 (weight gbest)	2.0
Number of particles/ population size	20
Maximun number	30 (default)
Inertia weight (ω)	1.0 (default)
Max weight (ω max)	1.0 (default)
Min weight (ω min)	0.0 (default)

HASIL DAN PEMBAHASAN

Eksperimen ini menggunakan dua model pengujian yaitu model ANN dan model awPSO-ANN. Pengujian model dilakukan terhadap empat kelompok data yang didasarkan pada perubahan jumlah neuron yaitu satu *hidden layer artificial neural*

network dengan satu neuron, lima neuron sepuluh neuron, limabelas neuron

Dalam penelitian ini, indikator untuk mengetahui akurasi terbaik ditunjukkan oleh besarnya nilai RMSE dan MAE untuk masing-masing pengujian pada data testing.

Tingkat akurasi terbaik pada masing-masing variasi jumlah neuron terhadap pengujian model ANN pada laju inflasi ditunjukkan oleh tabel dibawah ini.

Tabel 3. RMSE Terbaik pada Kasus Yang Diteliti Dengan Pengujian ANN

Parameter & Hasil ANN	Jumlah Neuron			
	1 N	5 N	10 N	15 N
Learning Rate	0.1	0.4; 0.3	0.4	0.2
Momentum	0.1	0.5; 0.6	0.5	0.7
RMSE	0.141	0.099	0.089	0.126
MAE	0.131	0.082	0.067	0.094

Berdasarkan nilai RMSE pengujian model ANN pada tabel 3 menunjukkan bahwa RMSE terkecil atau akurasi terbaik pada 1 neuron, 5 neuron, 10 neuron dan 15 neuron secara berurutan adalah 0,141; 0,099; 0.089 dan 0.126 . Dengan demikian akurasi terbaik pada pengujian model ANN dengan satu hidden layer terjadi pada 10 neuron dengan parameter learning rate sebesar 0.4 dan momentum sebesar 0.5.

Sedangkan untuk pengujian menggunakan model awPSO-ANN pada

laju inflasi, hasil terbaik pada masing-masing parameter jumlah neuron ditunjukkan pada tabel 4.

Tabel 4. RMSE Terbaik pada Kasus Yang Diteliti Dengan Pengujian awPSO-ANN

Parameter & Hasil awPSO-ANN	Jumlah Neuron			
	1 N	5 N	10 N	15 N
Learning Rate	0.1	0.2	0.3	0.1
Momentum	0.1	0.5	0.5	0.1
RMSE	0.131	0.104	0.104	0.131
MAE	0.107	0.084	0.086	0.107

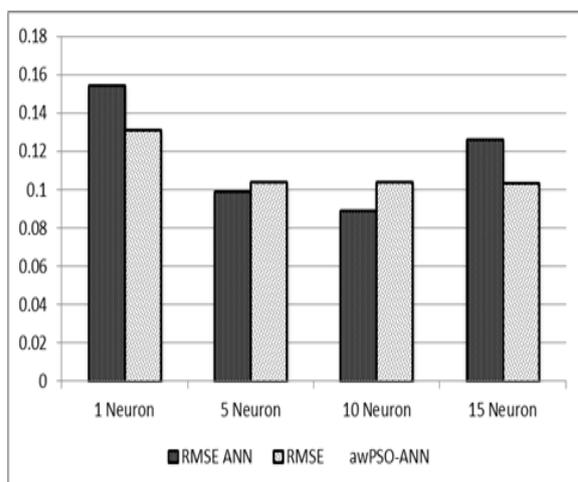
Berdasarkan nilai RMSE pengujian model awPSO-ANN pada tabel 4 menunjukkan bahwa RMSE terkecil atau akurasi terbaik pada 1 neuron, 5 neuron, 10 neuron dan 15 neuron secara berurutan adalah 0,131; 0,104; 0.104 dan 0.131 . Dengan demikian akurasi terbaik pada pengujian model awPSO-ANN dengan satu hidden layer terjadi pada 5 neuron dengan parameter learning rate sebesar 0.2 dan momentum sebesar 0.5 dan 10 neuron dengan parameter learning rate sebesar 0.3 dan momentum sebesar 0.5.

Penelitian ini menemukan bahwa maka akurasi terbaik dari keseluruhan hasil eksperimen terjadi pada nilai RMSE 0.089 yaitu pada penerapan model ANN dengan parameter satu hidden layer, 10 neuron, learning rate 0,4 dan momentum 0,5 .

Kemudian data nilai RMSE yang diperoleh pada setiap kelompok eksperimen dilakukan penghitungan nilai rata-rata RMSE, yang hasilnya ditunjukkan oleh tabel 5.

Tabel 5. : Rata-rata nilai RMSE pada Kasus Yang Diteliti

Size Neuron	Rata-Rata RMSE ANN	Rata-Rata RMSE awPSO-ANN
1 Neuron	0.279	0.196
5 Neuron	0.139	0.154
10 Neuron	0.147	0.148
15 Neuron	0.179	0.137
Overall	0.181	0.157



Gambar 5 : Grafik Perbandingan Rata-rata Nilai RMSE Pada Data Testing

Tabel 5, menunjukkan bahwa berdasarkan nilai rata-rata RMSE pengujian penerapan model ANN dibanding model awPSO-ANN pada

hidden layer dengan 1 neuron dan 15 neuron terjadi peningkatan akurasi prediksi laju inflasi. Sedangkan pada hidden layer dengan 5 neuron terjadi penurunan akurasi prediksi laju inflasi dari model ANN dibanding model awPSO-ANN. Kemudian pada pada hidden layer 10 neuron tidak terjadi perubahan akurasi prediksi laju inflasi pada model ANN terhadap model awPSO-ANN.

Nilai rata-rata RMSE dari seluruh data eksperimen penelitian ini menunjukkan bahwa secara umum penerapan model awPSO-ANN memiliki akurasi yang lebih baik dibanding model ANN.

SIMPULAN

Berdasarkan pengujian penerapan model awPSO-ANN dalam prediksi laju inflasi yang dilakukan dengan 147 eksperimen yang terbagi dalam empat variasi jumlah neuron pada satu hidden layer yaitu 1 neuron, 5 neuron, 10 neuron dan 15 neuron ditemukan bahwa rata-rata nilai RMSE penerapan model awPSO-ANN dalam penelitian ini lebih kecil dibandingkan dengan penerapan model ANN dan berdasarkan analisis uji beda menunjukkan bahwa ada perbedaan yang signifikan pada nilai RMSE penerapan model awPSO-ANN dengan penerapan model ANN sehingga dapat disimpulkan bahwa penerapan model awPSO-ANN

meningkatkan akurasi dalam prediksi laju inflasi berdasarkan atribut laju inflasi dan indeks harga konsumen (IHK) empat kota besar yaitu Jakarta, Surabaya, Bandung dan Medan.

DAFTAR PUSTAKA

- [1] Raghbendra Jha and Tu Dang, "Inflation Variability and The Relationship Between Inflation and Growth," Canberra, Australia, 2011.
- [2] Adnan Haider and Muhammad Nadeem Hanif, "Inflation Forecasting in Pakistan Using Artificial Neural Network," Karachi, Pakistan, 2007.
- [3] Jane M Binner, Alicia M Gazely, Shu Heng Chen, and Bin Tzong Chie, "Financial Innovation and Divisia Money In Taiwan : Comparative Evidence From Neural Network and Vector Error-Correction Forecasting Models," *Contemporary Economic Policy*, Vol 22, pp. 213-224, 2004.
- [4] Jane M Binner et al., "A Comparison of Linier Forecasting Models and Neural Networks : An Application to Euro Inflation and Euro Divisia," *Applied Economics*, 37, pp. 665-680, 2005.
- [5] Saeed Moshiri and Norman Cameron, "Neural Network Versus Econometric MOdels in Forecasting Inflation," *Journal of Forecasting* 19, pp. 201-217, 2000.
- [6] Emi Nakamura, "Inflation Forecasting Using A Neural Network," 2005.
- [7] Massimiliano Marcellino, "A Comparison of Time Series Model for Forecasting GDP Growth and Inflation," *Journal of Forecasting*, 2009.
- [8] A. Nazif Catik and Mehmet Karaquka, "A Comparative Analysis of Alternative Univariate Time Series Models in Forecasting Inflation," Dusseldorf Germany, 2011.
- [9] Recep Duzgun, "Generalized Regression Neural Network for Inflation Forecasting," *International Research Journal Of Finance and Economic*, pp. 59-70, 2010.
- [10] Y.H Zweiri, J.F Whidborne, and L.D Sceviratne, "A three-term backpropagation algorithm," *Neurocomputing*, vol. 50, pp. 305-318, 2003.
- [11] Haza Nuzly Abdull Hamed, "Particle Swarm Optimization For Neural Network Learning Enhancement," Malaysia, 2006.
- [12] S.-C. Chu, Roddick J. F., and Pan J. S., "Ant colony system with communication strategies," *Information Sciences*, vol. vol.167, pp. pp.63-76.
- [13] Xiaouhui Hu and Russell Eberhart, "Solving Contstrained Nonlinier Optimization Problems With Particle Swarm Optimization," in *6th World Multiconference on Systemics*, Orlando, USA, 2002.
- [14] Rania Hassan, Babak Cohanim, and Olivier De Weck, "A Comparison of Particle Swarm Optimization and The Genetic Algorithm," *American Institute of Aeronautics and Astronautics*, 2004.
- [15] Andreas Windisch, Stefan Wappler, and Joachim Wegener, "Applying Particle Swarm Optimization to Software Testing," in *Proceedings of the 2007 conference on Genetic and evolutionary computation*, London, England, United Kingdom, 2007.

- [16] G. P. Zhang, "Neural Network in Business Forecasting," *Idea Group Publishing*, 2003.
- [17] F. Van Den Berg, "An Analysis of Particle Swarm Optimization ," *Faculty of Natural and Agriculture Science, University of Pretoria.*, 2001.
- [18] J. Malik, R. Mishra, and I Singh, "PSO-ANN Approach for Estimating Drilling Induced Damage In CFRP Laminates," *Advances In Production Engineering & Management*, vol. Vol. 6 No. 2, pp. 95-104, 2011.
- [19] N. T. Danh, H.N. Phien, and A.D. Gupta, "Neural network models for river flow forecasting," *Water SA*, vol. Vol. 25 No. 1, pp. 33-39, 1999.
- [20] D. E. Rumelhart, G. E. Hinton, and R. Williams, "Learning internal representations by error propagation.," *Paralled Distributed Processing : Explorations in the Microstructure of Cognition*, vol. 1, pp. 318-362, 1986.
- [21] J Kennedy and R. Eberhart, *Swarm Intelligence*. San Francisco, CA: Morgan Kaufmann Publishers, Inc, 2001.
- [22] R. Eberhart and Y. Shi, *Particle Swarm Optimization : Development, Applications and Resources*. Indiana Polis, USA: Purdue School Of Engineering and Technology., 2001.
- [23] Dian Wahyuningsih, Ida Zuhroh, and Zainuri, "Prediksi Inflasi Indonesia Dengan Model Artificial Neural Network," *Journal of Indonesian Applied Economics*, vol. Vol. 2 No. 2, pp. p113-127, 2008.