

PREDIKSI HARGA EMAS MENGUNAKAN *FEED FORWARD NEURAL NETWORK* DENGAN METODE *EXTREME LEARNING MACHINE*

Nisa Afida Izati¹, Budi Warsito², Tatik Widiharih³
^{1,2,3}Departemen Statistika FSM Universitas Diponegoro
budiwrst2@gmail.com

ABSTRACT

The prediction of gold price aims to find out the gold price in the future on the basis of historical data on gold prices in the past, so it can be used as a consideration by gold investors to investing in gold. Prediction methods that do not require assumptions, one of which is Artificial Neural Networks. In this study, using Artificial Neural Networks, Feed Forward Neural Network with Extreme Learning Machine (ELM). ELM is a non-iterative algorithm so ELM has advantages in process speed. The input weight and bias for this method are determined randomly. After that, to find the final weight using the Moore-Penrose Generalized Inverse calculation on the hidden layer output matrix. The best model selection criteria uses the Mean Absolute Percentage Error (MAPE). This study shows that the results of the training and testing process from the model 1 input neuron and 7 hidden neurons are very good, because it produces MAPE training = 0.6752% and MAPE testing = 0.8065%. Also gives a very good prediction result because it has MAPE = 0.5499%

Keywords: gold price, Extreme Learning Machine, MAPE

1. PENDAHULUAN

Logam mulia emas memiliki ketahanan yang tinggi untuk disimpan dalam waktu yang lama sehingga tidak mudah rusak walau beberapa tahun berlalu, lunak, tahan korosi, dan mudah ditempa. Emas memiliki *supply* yang terbatas dan tidak mudah didapat, sementara permintaan terhadap emas tidak pernah berkurang. Hal ini menjadikan nilai atau harga emas cenderung stabil dan naik, sangat jarang turun.

Kenaikan harga emas ini mendorong investor untuk lebih memilih berinvestasi emas daripada pasar modal. Pengetahuan mengenai keuntungan dan risiko yang didapat ketika terjun di bidang investasi diperlukan oleh investor sebelum memasuki dunia investasi agar tujuan investasi tercapai. Salah satu pengetahuan penting berinvestasi emas adalah prediksi harga emas. Prediksi harga emas bertujuan untuk mengetahui peluang investasi harga emas di masa yang akan datang sehingga dapat digunakan sebagai pertimbangan oleh investor emas untuk mengetahui perubahan harga emas.

Metode peramalan saat ini sangat banyak dan memerlukan asumsi yang harus dipenuhi, namun terdapat metode yang tidak memerlukan asumsi, salah satunya adalah jaringan syaraf tiruan (JST). JST merupakan sistem pemrosesan informasi yang memiliki karakteristik mirip dengan jaringan syaraf pada makhluk hidup (Fausett, 1994). Arsitektur jaringan syaraf tiruan yang banyak digunakan dalam penelitian saat ini yaitu *Feed Forward Neural Network* (FFNN), FFNN merupakan model yang sederhana jika digunakan untuk menyelesaikan masalah data time series. Dalam FFNN, *input layer* dihubungkan dengan *hidden layer* lalu *hidden layer* dihubungkan dengan *output layer*. *Input layer* tidak memiliki hubungan langsung dengan *output layer*.

Peramalan dengan metode-metode JST FFNN yang telah ada sebelumnya membutuhkan waktu komputasi yang relatif lebih lama karena parameter jaringannya ditentukan secara iteratif menggunakan metode pelatihan. Huang *et al* (2004) menemukan sebuah metode baru dalam JST bernama *Extreme Learning Machine* (ELM). Bobot awal (bobot *input nodes* ke *hidden nodes* dan bobot bias ke *hidden nodes*) pada metode *Extreme Learning Machine* ditentukan secara random. Sehingga ELM memiliki kelebihan dalam hal waktu. Bobot akhir pada metode ELM ditentukan dengan melakukan perhitungan *Moore-Penrose Generalized Invers*. Berdasarkan latar belakang yang telah diuraikan, maka penulis mengambil topik penelitian prediksi harga emas menggunakan *Feed Forward Neural Network* (FFNN) dengan metode *Extreme Learning Machine*.

2. TINJAUAN PUSTAKA

2.1. Harga Emas Dunia

Harga emas yang dijadikan standar harga emas seluruh negara di dunia adalah harga emas yang ditentukan oleh *London Bullion Market Association* (LBMA) dengan sistem *London Gold Fixing*. LBMA menentukan harga emas dua kali dalam sehari setiap hari kerja, yaitu pukul 10.30 (*Gold A.M*) dan pukul 15.00 (*Gold P.M*) dengan mata uang Dolar Amerika Serikat per *troy ounce*. Harga yang digunakan sebagai patokan harga emas seluruh negara di dunia adalah harga Gold P.M. (LBMA, 2015).

2.2. Analisis Runtun Waktu

Runtun waktu merupakan urutan serangkaian pengamatan yang beruntun (Wei, 2006). Analisis runtun waktu menyatakan pengamatan pada waktu t (Z_t) bergantung pada satu atau beberapa pengamatan sebelumnya (Z_{t-k}).

1. Autocorrelation Function (ACF)

ρ_k dinamakan fungsi autokorelasi (ACF) pada lag k waktu yang mempresentasikan kovarian dan korelasi antara Z_t dan Z_{t-k} dari proses yang sama hanya dipisahkan dengan k lag waktu. Fungsi autokorelasi dirumuskan sebagai berikut:

$$\rho_k = \frac{\text{cov}(Z_t, Z_{t+k})}{\text{cov}(Z_t, Z_t)} \text{ dengan garis Batas ACF} = 2 \times \sqrt{\text{Var}(\hat{\rho}_k)}$$

2. Partial Autocorrelation Function (PACF)

Menurut Wei (2006), fungsi autokorelasi parsial (PACF) dapat diturunkan berdasarkan model regresi dengan Z_{t+k} sebagai variabel terikat dari sebuah proses stasioner dengan rata-rata nol yang diregresikan dengan variabel $Z_{t+k-1}, Z_{t+k-2}, \dots, Z_t$ pada lag ke- k Fungsi autokorelasi parsial dirumuskan sebagai berikut:

$$\hat{\phi}_{kk} = \frac{\begin{vmatrix} 1 & \rho_1 & \rho_2 & \dots & \rho_{k-2} & \rho_1 \\ \rho_1 & 1 & \rho_1 & \dots & \rho_{k-3} & \rho_2 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \dots & \rho_1 & \rho_k \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 & \rho_2 & \dots & \rho_{k-2} & \rho_{k-1} \\ \rho_1 & 1 & \rho_1 & \dots & \rho_{k-3} & \rho_{k-2} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \dots & \rho_1 & 1 \end{vmatrix}} \text{ dengan garis batas PACF} = 2 \times \sqrt{\text{Var}(\hat{\phi}_{kk})}$$

2.3. Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan adalah suatu sistem yang memiliki kemiripan dengan jaringan syaraf makhluk hidup dalam memproses informasi. Obyek pengamatan dikenali dengan melakukan suatu pelatihan yang terstruktur dan terus menerus sampai sistem jaringan tersebut mampu mengenali obyek tersebut. Secara garis besar JST mempunyai dua tahap pemrosesan informasi yaitu tahap pelatihan dan tahan pengujian.

Tahap pelatihan (*training*) dimulai dengan memasukkan pola-pola pelatihan (data latih) ke dalam jaringan. Dengan menggunakan pola-pola ini jaringan akan mengubah bobot menjadi penghubung antar neuron. Tahap pengujian (*testing*) merupakan pengujian suatu pola masukan yang belum pernah dilatihkan sebelumnya (data uji) menggunakan bobot-bobot yang telah dihasilkan pada tahap pelatihan.

Susunan dari neuron-neuron dalam lapisan dan pola keterhubungan antar lapisan disebut sebagai arsitektur jaringan. JST memiliki beberapa bagian utama yang menyusunnya antara lain *input*, bias, fungsi aktivasi, *hidden layer*, dan *output*. *Input* merupakan bagian dari jaringan syaraf tiruan yang berperan dalam memberikan informasi/masukkan data yang akan disampaikan ke neuron dari informasi tersebut akan diterima dan diproses oleh neuron dan akan diberikan nilai bobot. Menurut Flores, *et al* (2012) penentuan *input* pada JST dapat dilakukan berdasarkan lag yang signifikan pada plot ACF dan PACF.

Bias adalah nilai masukan (*input*) yang ditambahkan ke dalam suatu jaringan syaraf tiruan dimana nilainya selalu sama dengan 1 (Siang, 2005). Bias memiliki fungsi untuk mengubah nilai threshold menjadi = 0 (bukan = θ), sedangkan *threshold* adalah sebuah konstanta yang berfungsi sebagai penimbang dalam suatu hubungan dari suatu unit tertentu (Warsito, 2009).

Fungsi aktivasi dalam JST dipakai untuk menentukan *output* suatu neuron. Argumen fungsi aktivasi adalah net *input* (kombinasi linier *input* dan bobotnya). Warsito (2009) mengemukakan beberapa fungsi aktivasi yang biasa digunakan pada JST sebagai berikut:

- Fungsi Identitas (Linier/Purelin)

Fungsi identitas akan menghasilkan *output* yang sama dengan *input*nya, fungsi identitas biasa digunakan untuk kasus dengan *output* jaringan berupa bilangan riil (bukan hanya pada *range* [0,1] atau [-1,1]), dengan fungsi sebagai berikut:

$$f(x) = x, \text{ untuk } -\infty < x < \infty \quad (1)$$

- Fungsi Sigmoid Biner (Logistic Sigmoid)

Fungsi sigmoid biner bernilai antara 0 sampai 1, fungsi sigmoid biner digunakan apabila *output* yang diinginkan merupakan bilangan riil antara 0 sampai 1. Fungsi sigmoid biner dinyatakan sebagai berikut:

$$f(x) = \frac{1}{1+e^{-x}} \quad \text{untuk } -\infty < x < \infty \quad (2)$$

Hidden layer berfungsi untuk menyalurkan informasi dari *input layer* ke *output layer* dimana sebelumnya informasi tersebut sudah dilakukan pemrosesan untuk memberikan *output* yang optimal. Hasil keluaran proses pada *hidden layer* disebut *output hidden layer*. Informasi *output hidden layer* ini selanjutnya diteruskan ke *output layer*. *Output* merupakan hasil dari semua perhitungan *input* dan bobot-bobot yang diberikan pada neuron yang saling berhubungan dengannya. Pada dasarnya *output* merupakan hasil akhir pemrosesan.

2.4. Pre-processing dan Post-processing

Pre-processing merupakan proses mengubah skala data sebelum dilakukan tahap pelatihan, *pre-processing* bertujuan agar data dapat diproses sesuai dengan fungsi aktivasi yang digunakan (Warsito, 2009). Fungsi aktivasi yang digunakan pada penelitian ini adalah fungsi sigmoid biner. Menurut Siang (2005), jika menggunakan fungsi aktivasi sigmoid biner maka data harus ditransformasikan dulu. Keluaran fungsi aktivasi sigmoid biner adalah bilangan riil [0,1] maka data ditransformasikan ke interval [0,1]. Tapi akan lebih baik jika ditransformasikan ke interval yang lebih kecil, misal pada interval [0.1, 0.9]. Ini mengingat fungsi sigmoid merupakan fungsi asimtotik yang nilainya tidak pernah mencapai 0 ataupun 1.

Post-processing adalah proses penskalaan setelah dilakukan tahap pelatihan. *Post-processing* dimaksudkan agar data hasil peramalan dengan jaringan syaraf dapat dilihat secara mudah dalam nilai yang sama dengan asalnya. Menurut Siang (2005) berikut rumus normalisasi data ke dalam interval [0.1,0.9]:

$$x'_t = \frac{0,8(x_t - \min(x))}{(\max(x) - \min(x))} + 0,1 \quad (3)$$

dimana x'_t : data ke-t hasil normalisasi

x_t : data ke-t sebelum dinormalisasi

$\min(x)$: nilai minimum dari data sebelum dinormalisasi

$\max(x)$: nilai maksimum dari data sebelum dinormalisasi

Rumus denormalisasi data dari interval [0.1,0.9]:

$$\hat{x}_t = \frac{(\hat{x}'_t - 0,1)(\max(x) - \min(x))}{0,8} + \min(x) \quad (4)$$

dimana \hat{x}'_t : data prediksi ke-t normalisasi

\hat{x}_t : data prediksi ke-t hasil denormalisasi

$\min(x)$: nilai minimum dari data sebelum dinormalisasi

$\max(x)$: nilai maksimum dari data sebelum dinormalisasi

2.5. Extreme Learning Machine (ELM)

ELM merupakan metode pembelajaran baru dari jaringan syaraf tiruan. Metode ini pertama kali diperkenalkan oleh Huang, *et al* (2004). ELM merupakan jaringan syaraf tiruan *feedforward* dengan *single hidden layer* atau biasa disebut dengan *Single Hidden Layer Feedforward neural Networks* (SLFNs) dengan m unit pada *hidden layer*. Metode pembelajaran ELM dibuat untuk mengatasi kelemahan-kelemahan dari jaringan syaraf tiruan *feedforward* terutama dalam hal *learning speed*.

Semua bobot pada jaringan JST FFNN (bobot *input nodes* ke *hidden nodes*, bobot bias ke *hidden nodes*, dan bobot *hidden nodes* ke *output*) ditentukan secara iteratif dengan menggunakan metode pelatihan. Pada ELM bobot awal (bobot *input nodes* ke *hidden nodes* dan bobot bias ke *hidden nodes*) dipilih secara *random*, setelah itu untuk mencari bobot akhir (bobot *hidden nodes* ke *output*) dilakukan dengan melakukan perhitungan analitis yaitu menggunakan *Moore-Penrose Generalized Invers*.

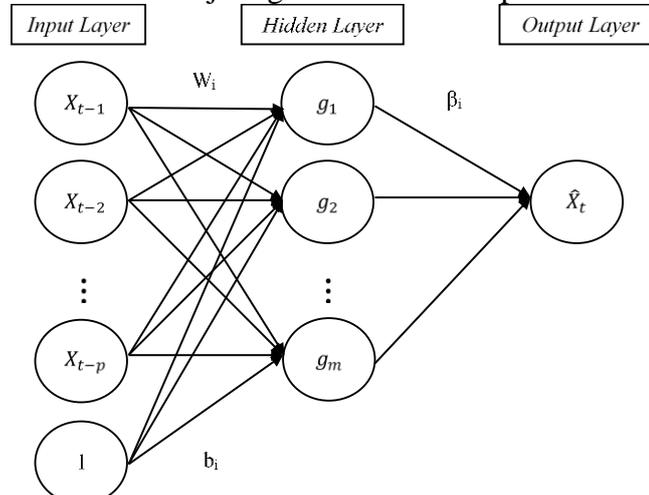
Arsitektur jaringan metode ELM terdiri p unit pada *input layer*, m unit pada satu *hidden layer* (sehingga disebut *single hidden layer*), satu bias, dan satu *output*. Juga terdapat bobot penghubung antara *input nodes* dan *hidden nodes*, antara bias dan *hidden nodes*, serta antara *hidden nodes* dan *output*. Berikut adalah model runtun waktu *Extreme Learning Machine* (ELM) dengan m unit pada satu *hidden layer*:

$$\hat{x}_t = \sum_{i=1}^m \beta_i g(w_i \cdot x_j + b_i)$$

dimana:

\hat{x}_t adalah data prediksi, β_i adalah bobot yang menghubungkan neuron *hidden* dengan neuron *output* dengan $\beta_i = (\beta_1, \beta_2, \dots, \beta_m)$, w_i adalah bobot yang menghubungkan neuron *input* dengan neuron *hidden* dimana $w_i = (w_{1i}, w_{2i}, \dots, w_{pi})$, x_j merupakan vektor data *input* dengan $x_j = (x_{j(t-1)}, x_{j(t-2)}, \dots, x_{j(t-p)})$, k merupakan neuron *input* dengan $k = 1, 2, \dots, p$, i merupakan neuron *hidden* dimana $i = 1, 2, \dots, m$, j merupakan sampel data dengan $j = 1, 2, \dots, n$, b_i adalah bobot bias ke neuron *hidden* dengan $b_i = (b_1, b_2, \dots, b_m)$

Berikut disajikan gambar arsitektur jaringan metode ELM pada Gambar 1.



Gambar 1 Arsitektur Jaringan Extreme Learning Machine

Moore-Penrose Generalized Invers merupakan perluasan dari konsep invers matriks. Invers *Moore - Penrose* ada untuk setiap matriks baik matriks bujur sangkar maupun yang tidak bujur sangkar dan yang singular maupun yang non singular (Britz, 2007). Gustasffon (2018) mengemukakan bahwa untuk matriks H ukuran $n \times m$, invers *Moore-Penrose* dari matriks H didefinisikan sebagai matriks H^+ ukuran $m \times n$.

Jika matriks H memiliki kolom yang bebas linier ($n \geq m$) maka

$$H^+ = (H^T H)^{-1} H^T \quad (5)$$

Jika matriks H memiliki baris yang bebas linier ($n \leq m$) maka

$$H^+ = H^T (H H^T)^{-1} \quad (6)$$

Berikut algoritma metode Extreme Learning Machine:

Langkah 1 : Membuat plot ACF dan PACF

Langkah 2 : Membagi data menjadi 2 yaitu *training* dan *testing*

Algoritma Training

Langkah 3 : *Input* data target proses *training*

Langkah 4 : Melakukan *pre-processing* data berupa normalisasi data

Langkah 5 : *Input* data *training* yang sudah *dipre-processing* berdasarkan lag yang signifikan pada plot ACF dan PACF

Langkah 6 : Inisialisasi bobot awal (bobot *input nodes* ke *hidden nodes* dan bobot bias ke *hidden nodes*) dengan bilangan acak interval tergantung fungsi aktivasi yang digunakan. Pada kasus ini menggunakan fungsi aktivasi sigmoid biner sehingga bobot awal harus di dalam interval $[0, 1]$.

Langkah 7 : Setiap unit *input* menerima sinyal input dan meneruskan sinyal tersebut pada seluruh unit lapisan di atasnya (unit *hidden*).

Langkah 8 : Menghitung semua keluaran di unit tersembunyi menggunakan fungsi aktivasi sigmoid biner maka

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_m \cdot x_1 + b_m) \\ g(w_1 \cdot x_2 + b_1) & \cdots & g(w_m \cdot x_2 + b_m) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_n + b_1) & \cdots & g(w_m \cdot x_n + b_m) \end{bmatrix}_{n \times m}$$

$w_i \cdot x_j$ merupakan hasil kali dalam (*inner product*), w_i adalah vektor bobot yang menghubungkan neuron *input* dengan neuron *hidden* dimana $w_i = (w_{1i}, w_{2i}, \dots, w_{pi})$, x_j adalah vektor data *input*, $x_j = (x_{j(t-1)}, x_{j(t-2)}, \dots, x_{j(t-p)})$

$$g(w_i \cdot x_j + b_i) = \frac{1}{1 + e^{-(w_i \cdot x_j + b_i)}}$$

Langkah 9 : Menghitung bobot yang menghubungkan neuron *hidden* dengan neuron *output*

$$\beta = H^+ x_t$$

$$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{\bar{N}} \end{bmatrix}_{\bar{N} \times 1} = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_{\bar{N}} \cdot x_1 + b_m) \\ g(w_1 \cdot x_2 + b_1) & \cdots & g(w_{\bar{N}} \cdot x_2 + b_m) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_n + b_1) & \cdots & g(w_{\bar{N}} \cdot x_n + b_m) \end{bmatrix}_{m \times n}^+ \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_{n \times 1}$$

dimana β_i adalah vektor bobot yang menghubungkan neuron *hidden* dengan neuron *output* $\beta_i = (\beta_1, \beta_2, \dots, \beta_m)$, H adalah matriks yang tersusun dari output *hidden layer*, H^+ adalah *Moore-Penrose Generalized Invers* dari matriks H , x_t merupakan vektor data target dimana dalam JST target merupakan data aktual $x_t = (x_1, x_2, \dots, x_n)$

Langkah 10 : Menghitung semua keluaran di unit *output*

$$\hat{x}_t = H \beta \quad (7)$$

Langkah 11 : Melakukan *post-processing* data

Langkah 12 : Menghitung nilai *error*

Algoritma Testing

Langkah 13 : *Input* data target proses *testing*

Langkah 14 : Melakukan *pre-processing* data

Langkah 15 : *Input* data *testing* yang sudah di *pre-processing* berdasarkan lag yang signifikan pada plot ACF dan PACF

Langkah 16 : Inisialisasi bobot (bobot *input nodes* ke *hidden nodes*, bobot bias ke *hidden nodes*, dan bobot *hidden nodes* ke *output*) yang diperoleh dari langkah 6

Langkah 17 : Lakukan seperti langkah 7 hingga langkah 8 untuk mendapatkan *output hidden layer*

Langkah 18 : Lakukan seperti langkah 10 hingga langkah 12 untuk mendapatkan nilai prediksi dan nilai error

2.6. Kriteria Pemilihan Model Terbaik

Dalam penelitian ini, pemilihan model terbaik dilihat berdasarkan nilai *Mean Absolute Percentage Error* (MAPE). MAPE merupakan nilai tengah kesalahan presentase *absolute*, jadi pada MAPE nilai positif atau negatif *error* tidak dihiraukan. Nilai MAPE dapat dihitung menggunakan rumus:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|x_t - \hat{x}_t|}{x_t} \times 100\% \quad (8)$$

Dimana x_t : data aktual ke-t, \hat{x}_t : data prediksi ke-t, n : ukuran pengamatan

Semakin kecil nilai MAPE menunjukkan bahwa persentase kesalahan yang dihasilkan oleh model juga semakin kecil. Menurut Chang, *et al* (2007) dalam Halimi, *et al* (2013) nilai evaluasi yang dihasilkan mempunyai kriteria MAPE seperti berikut:

- MAPE < 10% : kemampuan peramalan sangat baik
- 10% ≤ MAPE < 20% : kemampuan peramalan baik
- 20% ≤ MAPE < 50% : kemampuan peramalan cukup
- MAPE ≥ 50% : kemampuan peramalan buruk

3. METODOLOGI PENELITIAN

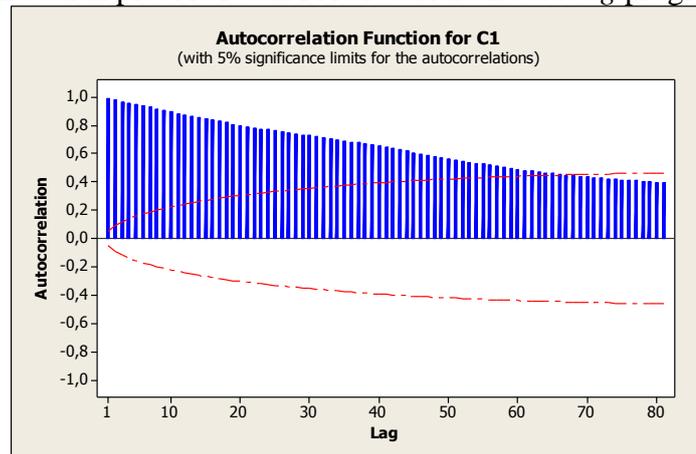
Data yang digunakan dalam penelitian ini merupakan data sekunder yang bersumber dari *website* Investing (<https://id.investing.com/>). Data tersebut merupakan data harian harga emas dunia dalam USD/*troy ounce* pada tahun 2014-2018 dengan menggunakan hari aktif sebanyak 1319 data. Penentuan variabel *input* yang akan digunakan dengan mengidentifikasi data menggunakan plot ACF dan PACF. Sebanyak 80% dari data digunakan untuk data *training* dan 20% dari data digunakan untuk data *testing*. Program komputer yang digunakan dalam pengolahan data penelitian ini adalah *software* MATLAB 2017b, Minitab 16, dan Microsoft Excel 2013. Berikut adalah langkah-langkah analisis data:

- Mempersiapkan data harga emas dunia dalam USD per *troy ounce*
- Membuat plot ACF dan PACF
- Membagi data menjadi 2 yaitu *training* sebanyak 80% dan *testing* sebanyak 20%
- Input* data target untuk proses *training*
- Melakukan *pre-processing* data
- Input* data *training* yang sudah di *pre-processing* berdasarkan lag yang signifikan
- Inisialisasi bobot awal (bobot *input nodes* ke *hidden nodes* dan bobot bias ke *hidden nodes*) dengan bilangan acak di dalam interval [0, 1].
- Melakukan proses *training* metode *Extreme Learning Machine*
- Melakukan *post-processing* data
- Menghitung nilai MAPE proses *training*
- Input* data target proses *testing*
- Melakukan *pre-processing* data
- Input* data *testing* yang sudah di *pre-processing* berdasarkan lag yang signifikan

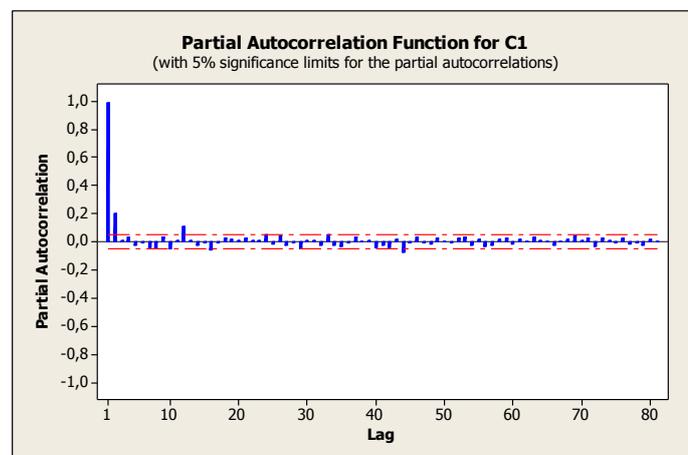
14. Melakukan proses *testing* metode *Extreme Learning Machine* menggunakan bobot yang diperoleh dari proses *training*
15. Melakukan *post-processing* data
16. Menghitung nilai MAPE proses *testing*
17. Melakukan peramalan harga emas dunia menggunakan model terbaik

4. ANALISIS DAN PEMBAHASAN

Penentuan variabel input jaringan dapat dilakukan dengan menggunakan bantuan plot ACF dan PACF, berikut ini adalah plot ACF dan PACF dari hasil *running* program Minitab 16:



Gambar 2 Plot ACF Data Harga Emas



Gambar 3 Plot PACF Harga Emas

Berdasarkan Gambar 2 bisa disimpulkan data runtun waktu harga emas dunia yang digunakan tidak bisa dimodelkan menggunakan model MA karena nilai ACF-nya terus turun secara eksponensial. Dilihat dari Gambar 3 bisa disimpulkan data runtun waktu harga emas dunia yang digunakan dapat dimodelkan secara autoregresif karena terpotong pada lag 1, 2 dan 12, sehingga dapat dikatakan bahwa jaringan terdiri atas X_t dipengaruhi oleh variabel X_{t-1} , X_{t-2} dan X_{t-12} .

Pada penelitian ini neuron *hidden* dibatasi sejumlah 5, 6, dan 7 berdasarkan *trial dan error*. Neuron dalam *hidden layer* tersebut dikombinasi dengan variabel *input* untuk membangun beberapa model. Hasil kombinasi tersebut menghasilkan 21 model FFNN. Dari 21 model tersebut akan dipilih satu model terbaik untuk digunakan model dalam peramalan harga emas berdasarkan kriteria MAPE terkecil. Semakin kecil nilai MAPE menunjukkan

semakin dekat nilai prediksi dengan nilai aktualnya. Berikut adalah 21 model kombinasi arsitektur jaringan:

Tabel 1. Kombinasi Neuron Input dan Neuron Hidden

Banyaknya Neuron Input	Neuron Input	Banyaknya Neuron Hidden
1	X_{t-1}	5
	X_{t-2}	5
	X_{t-12}	5
	X_{t-1}	6
	X_{t-2}	6
	X_{t-12}	6
	X_{t-1}	7
	X_{t-2}	7
	X_{t-12}	7
2	X_{t-1} dan X_{t-2}	5
	X_{t-1} dan X_{t-12}	5
	X_{t-2} dan X_{t-12}	5
	X_{t-1} dan X_{t-2}	6
	X_{t-1} dan X_{t-12}	6
	X_{t-2} dan X_{t-12}	6
	X_{t-1} dan X_{t-2}	7
	X_{t-1} dan X_{t-12}	7
	X_{t-2} dan X_{t-12}	7
3	X_{t-1}, X_{t-2} dan X_{t-12}	5
	X_{t-1}, X_{t-2} dan X_{t-12}	6
	X_{t-1}, X_{t-2} dan X_{t-12}	7

Data harga emas dunia dilakukan *pre-processing* dahulu sebelum masuk ke proses ELM. Nilai minimum data = 1050,80 dan nilai maksimum data = 1402,2 sehingga rumus *pre-processing* sebagai berikut: $x'_t = \frac{0,8 (x_t - 1050,8)}{(1402,2 - 1050,8)} + 0,1$

Berikut ini contoh perhitungan tahap *pre-processing* pada data runtun waktu harian harga emas dunia ke-1, ke-2, dan ke-1319:

$$x'_1 = \frac{0,8 (1205,4 - 1050,8)}{(1402,2 - 1050,8)} + 0,1 = 0,4520$$

$$x'_2 = \frac{0,8 (1225,0 - 1050,8)}{(1402,2 - 1050,8)} + 0,1 = 0,4966$$

$$\vdots$$

$$x'_{1319} = \frac{0,8 (1287,7 - 1050,8)}{(1402,2 - 1050,8)} + 0,1 = 0,6393$$

Berikut diilustrasikan proses *training dan testing* untuk model runtun waktu harga emas dengan arsitektur jaringan X_{t-1} sebagai variabel *input* dan jumlah *hidden nodes* sebanyak 7.

4.1. Proses Training Metode Extreme Learning Machine

Bobot awal (bobot *input nodes* ke *hidden nodes* dan bobot bias ke *hidden nodes*) ditentukan dengan dibangkitkan bilangan random dengan interval [0.1, 0.9] Berikut adalah bobot yang diperoleh berdasarkan *trial and error* untuk model runtun waktu harga emas dengan X_{t-1} sebagai variabel *input* dan jumlah *hidden nodes* sebanyak 7.

Tabel 2 Bobot Proses Training

Bobot	Keterangan
0,7302	Bobot <i>input</i> X_{t-1} ke <i>hidden nodes</i> 1
0,3439	Bobot <i>input</i> X_{t-1} ke <i>hidden nodes</i> 2
0,5841	Bobot <i>input</i> X_{t-1} ke <i>hidden nodes</i> 3
0,1078	Bobot <i>input</i> X_{t-1} ke <i>hidden nodes</i> 4
0,9063	Bobot <i>input</i> X_{t-1} ke <i>hidden nodes</i> 5
0,8797	Bobot <i>input</i> X_{t-1} ke <i>hidden nodes</i> 6
0,8178	Bobot <i>input</i> X_{t-1} ke <i>hidden nodes</i> 7
0,2607	Bobot bias ke <i>hidden nodes</i> 1
0,5944	Bobot bias ke <i>hidden nodes</i> 2
0,0225	Bobot bias ke <i>hidden nodes</i> 3
0,4253	Bobot bias ke <i>hidden nodes</i> 4
0,3127	Bobot bias ke <i>hidden nodes</i> 5
0,1615	Bobot bias ke <i>hidden nodes</i> 6
0,1788	Bobot bias ke <i>hidden nodes</i> 7

Perhitungan perkalian bobot *input* dengan variabel *input* ditambah bobot bias didapatkan *output hidden layer*. Berikut adalah *output hidden layer* proses *training*:

$$H = \begin{bmatrix} 0,6435 & 0,6026 & 0,6282 & 0,5767 & 0,6616 & 0,6589 & 0,6526 \\ 0,6510 & 0,6062 & 0,6343 & 0,5779 & 0,6706 & 0,6676 & 0,6608 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0,6766 & 0,6191 & 0,6554 & 0,5820 & 0,7013 & 0,6976 & 0,6890 \end{bmatrix}_{1054 \times 7}$$

Bobot akhir (bobot *neuron hidden* ke *output*) diperoleh dengan menghitung matriks *Moore-Penrose Generalized Invers* dari *Output Hidden Layer* dengan data aktual. Berikut adalah bobot akhir yang diperoleh dari proses *training*:

$$\beta = \begin{bmatrix} -1,2863 \times 10^7 \\ 0,4555 \times 10^7 \\ -0,0947 \times 10^7 \\ -0,2399 \times 10^7 \\ 0,1568 \times 10^7 \\ -1,1062 \times 10^7 \\ 2,0806 \times 10^7 \end{bmatrix}_{7 \times 1}$$

Output diperoleh dari perkalian matriks *Output Hidden Layer* dengan bobot akhir (bobot *neuron hidden* ke *output*). Berikut ini *output* proses *training* ELM:

$$\hat{x}'_t = \begin{bmatrix} 0,4539 \\ 0,4981 \\ \vdots \\ 0,6535 \end{bmatrix}_{1054 \times 1}$$

Output yang diperoleh dilakukan *post-processing* supaya dapat dibandingkan dengan data aslinya, dengan rumus:

$$\hat{x}_t = \frac{(\hat{x}'_t - 0,1)(1402,2 - 1050,8)}{0,8} + 1050,8$$

Berikut adalah *output* hasil *post-processing*

$$\hat{x}_1 = \frac{(0,4539 - 0,1)(1402,2 - 1050,8)}{0,8} + 1050,8 = 1206,27$$

$$\hat{x}_2 = \frac{(0,4981 - 0,1)(1402,2 - 1050,8)}{0,8} + 1050,8 = 1225,65$$

⋮

$$\hat{x}_{1054} = \frac{(0,6535 - 0,1)(1402,2 - 1050,8)}{0,8} + 1050,8 = 1293,91$$

Nilai MAPE proses *training* yang didapatkan adalah sebesar 0,6752%, nilai tersebut kurang dari 10%, maka dapat disimpulkan bahwa keakuratan peramalan pada proses *training* sangat baik.

4.2. Proses Testing Metode Extreme Learning Machine

Semua bobot (bobot *input* ke *hidden nodes*, bias ke *hidden nodes*, dan *hidden nodes* ke *output*) yang dipakai pada proses *testing* adalah bobot yang diperoleh dari dari proses *training*. Dengan melakukan perhitungan, berikut adalah *Output Hidden Layer* yang diperoleh pada proses *testing*:

$$H = \begin{bmatrix} 0,8186 & 0,5235 & 0,7070 & 0,3430 & 0,9532 & 0,9329 & 0,8856 \\ 0,7624 & 0,4970 & 0,6621 & 0,3348 & 0,8835 & 0,8652 & 0,8226 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0,7302 & 0,4818 & 0,6363 & 0,3300 & 0,8434 & 0,8263 & 0,7865 \end{bmatrix}_{264 \times 7}$$

Output didapatkan dengan menghitung semua keluaran di *output layer* menggunakan bobot akhir (bobot *hidden nodes* ke *output*) yang diperoleh dari proses *training*. Berikut *output* pada proses *testing* yang sudah dilakukan *post-processing*:

$$\begin{aligned} \hat{x}_1 &= 1338,73 \\ \hat{x}_2 &= 1307,67 \\ &\vdots \\ \hat{x}_{264} &= 1289,31 \end{aligned}$$

Nilai MAPE proses *testing* yang didapatkan adalah sebesar 0,8065%, nilai tersebut kurang dari 10%, maka keakuratan peramalan pada proses *testing* sangat baik.

4.3. Hasil Proses Training dan Testing

Berikut ini tabel hasil perhitungan MAPE proses *training* dan *testing* arsitektur FFNN menggunakan metode ELM pada 21 kombinasi model arsitektur jaringan.

Tabel 3 Hasil Perhitungan MAPE Data Training dan Testing

Lag Input	Neuron Hidden	MAPE training (%)	MAPE testing (%)
1	5	0,6770	0,7945
2	5	0,9506	0,9726
12	5	2,3828	1,9700
1	6	0,6766	0,7923
2	6	0,9495	0,9638
12	6	2,3820	1,9288
1	7	0,6752	0,8065
2	7	0,9467	0,9792
12	7	2,3840	2,1575
1 dan 2	5	0,6840	0,7651
1 dan 12	5	0,6830	0,7842
2 dan 12	5	0,9476	0,9720
1 dan 2	6	0,6830	0,7680
1 dan 12	6	0,6820	0,7723
2 dan 12	6	0,9476	0,9678
1 dan 2	7	0,6825	0,7717
1 dan 12	7	0,6816	0,7735
2 dan 12	7	0,9462	0,9617
1, 12, dan 12	5	4,7786	6,3701
1, 12, dan 12	6	4,7786	7,1923
1, 12, dan 12	7	4,7786	7,2518

Berdasarkan Tabel 3 dapat disimpulkan bahwa dari 21 model, yang memberikan nilai MAPE terkecil adalah pada model dengan arsitektur jaringan menggunakan x_{t-1} sebagai variabel *input* dengan jumlah *hidden nodes* sebanyak 7. Berikut disajikan grafik perbandingan *output* dan target proses *training* dan *testing* pada Gambar 4.



Gambar 4 Perbandingan Output dan Target Proses *Training* dan *Testing*

Berdasarkan Gambar 4 dapat dilihat bahwa proses *training* dan *testing* model arsitektur jaringan menggunakan x_{t-1} sebagai variabel *input* dengan jumlah *hidden nodes* sebanyak 7 telah memberikan hasil prediksi yang akurat yang ditunjukkan oleh kedekatan nilai aktual (garis berwarna merah) dengan nilai prediksinya (* berwarna biru).

4.4. Model Runtun Waktu Harga Emas

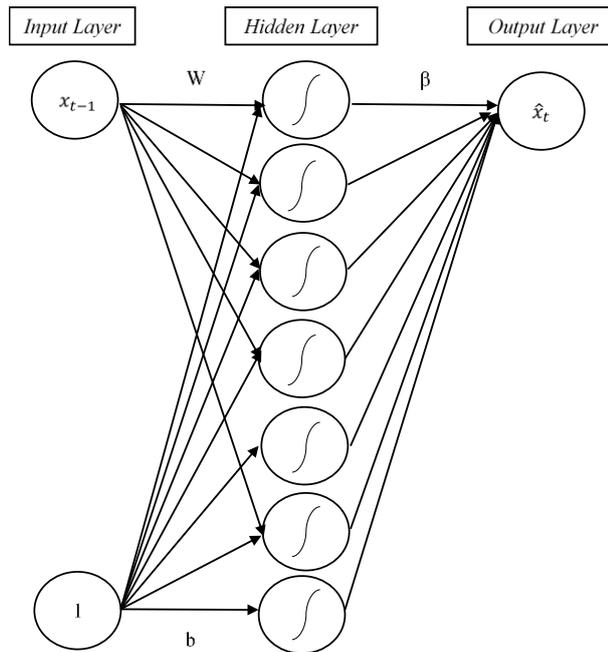
Model runtun waktu harga emas yang didapatkan adalah sebagai berikut:

$$\hat{x}_t = \sum_{i=1}^7 \beta_i g(w_i \cdot x_j + b_i)$$

$$\hat{x}_t = \beta_1 g(w_{11}x_{1(t-1)} + b_1) + \beta_2 g(w_{12}x_{1(t-1)} + b_2) + \beta_3 g(w_{13}x_{1(t-1)} + b_3) + \beta_4 g(w_{14}x_{1(t-1)} + b_4) + \beta_5 g(w_{15}x_{1(t-1)} + b_5) + \beta_6 g(w_{16}x_{1(t-1)} + b_6) + \beta_7 g(w_{17}x_{1(t-1)} + b_7)$$

$$\hat{x}_t = \left((-1,2863) \times 10^7 \left(\frac{1}{1 + \exp^{-(0,7302 x_{1(t-1)} + 0,2607)}} \right) \right) + \left((0,4555) \times 10^7 \left(\frac{1}{1 + \exp^{-(0,3439 x_{1(t-1)} + 0,5944)}} \right) \right) + \left(-0,0947 \times 10^7 \left(\frac{1}{1 + \exp^{-(0,5841 x_{1(t-1)} + 0,0255)}} \right) \right) + \left(-0,2399 \times 10^7 \left(\frac{1}{1 + \exp^{-(0,1078 x_{1(t-1)} + 0,4253)}} \right) \right) + \left(0,1568 \times 10^7 \left(\frac{1}{1 + \exp^{-(0,9063 x_{1(t-1)} + 0,3127)}} \right) \right) + \left((-1,1062) \times 10^7 \left(\frac{1}{1 + \exp^{-(0,8797 x_{1(t-1)} + 0,1615)}} \right) \right) + \left(2,0806 \times 10^7 \left(\frac{1}{1 + \exp^{-(0,8178 x_{1(t-1)} + 0,1788)}} \right) \right)$$

Secara visual arsitektur jaringan tersebut diilustrasikan pada Gambar 5 sebagai berikut:



Gambar 5 Arsitektur Jaringan Model Terbaik

4.5. Prediksi Harga Emas

Prediksi harga emas dunia untuk 10 periode ke depan dapat dilihat pada Tabel 4.

Tabel 2. Hasil Peramalan Emas Dunia 10 Periode ke Depan

Tanggal	Peramalan	Aktual	Error
1 Januari 2019	1.287,7646	1.283,35	4,4146
2 Januari 2019	1.287,8269	1.290,50	2,6731
3 Januari 2019	1.287,8872	1.301,20	13,3128
4 Januari 2019	1.287,9453	1.292,20	4,2547
7 Januari 2019	1.288,0014	1.296,40	8,3986
8 Januari 2019	1.288,0556	1.292,40	4,3444
9 Januari 2019	1.288,1079	1.298,50	10,3921
10 Januari 2019	1.288,1584	1.294,00	5,8416
11 Januari 2019	1.288,2071	1.296,10	7,8929
14 Januari 2019	1.288,2541	1.298,00	9,7459

Berdasarkan perhitungan, diperoleh MAPE sebesar 0,5499 %. Nilai MAPE yang didapatkan kurang dari 10%, maka dapat dikatakan keakuratan hasil peramalan sangat baik.

5. KESIMPULAN

Berdasarkan hasil analisis dan pembahasan pada bab sebelumnya, dapat diambil beberapa kesimpulan bahwa model terbaik harga emas dunia yang diperoleh adalah model FFNN dengan X_{t-1} sebagai variabel input dan dengan jumlah *hidden nodes* sebanyak 7. Model tersebut mampu memberikan hasil pemodelan yang sangat baik pada proses *training* yaitu dengan nilai MAPE sebesar 0,6752% dan mampu memberikan model dengan kemampuan peramalan yang sangat baik dengan MAPE 0,5499 %. Hasil peramalan harga emas dunia (USD/troy ounce) 10 periode ke depan dengan metode *Extreme Learning Machine* sangat baik karena mendekati dengan nilai aktualnya.

DAFTAR PUSTAKA

- Britz, T. 2007. *The Moore-Penrose Inverse Of A Free Matrix*. *Electronic Journal of Linear Algebra*, Vol 16, Hal : 208-215.
- Chang, P. C., Wang, Y. W. & Liu, C. H., 2007. *The Development of A Weighted Evolving Fuzzy Neural Network for PCB Sales Forecasting*. *Expert Systems with Applications*, Vol 32, No.1 : Hal 86-89.
- Fausett, L. 1994. *Fundamentals of Neural Networks; architectures, algorithms and applications*. Englewoods Cliffs, New Jersey: Prentice-Hall Inc
- Flores, J.H.F., Engel, P.M., & Pinto, R.C. 2012. *Autocorrelation and Partial Correlation Functions to Improve Neural Network Models on Univariate Time Series Forecasting*. *International Join Conference on Neural Network (IJNN)*. Brisbane, Australia
- Gustasffon, B. 2018. *Scientific Computing from a Historical Perspective*. Uppsala : Springer
- Halimi, R., Anggraeni, W. & Tyasnurita, R., 2013. *Pembuatan Aplikasi Peramalan Jumlah Permintaan Produk dengan Metode Time Series Exponential Smoothing Holts Winter di PT. Telekomunikasi Indonesia Tbk*. *Jurnal Teknik POMITS*, Vol. 1, No. 1 : Hal 1-6 .
- Huang, G.B., Zhu, Q.Y., Siew, C.K. 2004. *Extreme Learning Machine: A New Learning Scheme Of Feedforward Neural Network*. Universitas Teknologi Nanyang
- London Bullion Market Association. 2015. *LBMA Gold Price*. <http://www.lbma.org.uk/lbma-gold-price>. Diakses : 7 Maret 2019
- Siang, J. 2005. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab* . Yogyakarta: ANDI.
- Soejoeti, Z. 1987. *Materi Pokok Analisis Runtun Waktu*. Jakarta : Karunika
- Warsito, B. 2009. *Kapita Selektta Statistika Neural Network*. Semarang: BP Undip Semarang.
- Wei, W. 2006. *Time Analysis Univariate and Multivariate Methods, 2nd Edition*. USA: Addison Wesley Publishing Company, Inc
- Zhang, G. P. (1998). *Forecasting with Artificial Neural Networks : The State of the Art*. *Elsevier International Journal of Forecasting*, Vol. 14, No.1 : Hal 35-62