

OPTIMASI PARAMETER *NEURAL NETWORK* PADA DATA TIME SERIES UNTUK MEMPREDIKSI RATA-RATA KEKUATAN GEMPA PER PERIODE (Studi Kasus Gempa Bumi di Maluku Utara)

OPTIMIZATION PARAMETER OF *NEURAL NETWORK* FOR TIME SERIES DATA TO PREDICT THE MAGNITUDE OF PERIODICS EARTHQUAKE (Study Case Earthquake in North Maluku)

Muzakir Hi Sultan

Fakultas Matematika dan Ilmu Pengetahuan Alam, Program Studi Magister Matematika,
Universitas Brawijaya
Jl. Veteran Malang, Telp. (0341)554403. Email: zhasya91@yahoo.com

ABSTRAK

Gempa bumi merupakan suatu pergerakan tanah yang terjadi secara tiba-tiba hingga menimbulkan getaran, besarnya kekuatan gempa dapat mengakibatkan bencana baik kerusakan maupun korban jiwa. Untuk mengantisipasi bencana yang akan datang maka diperlukan suatu model khususnya untuk meramalkan besarnya kekuatan gempa. Pada penelitian ini, digunakan model *ARIMA* dan model kombinasi dari *Neural Network-Algorithm Genetik (NN-GA)* untuk memprediksi rata-rata kekuatan gempa bumi setiap bulan khususnya yang terjadi di wilayah Maluku Utara. Data yang digunakan adalah data kekuatan gempa berdasarkan skala richter yang diperoleh dari Badan Meteorologi, Klimatologi dan Geofisika (BMKG) kota Ternate. Sebagai input pada model *ARIMA* dan *NN-GA* digunakan rata-rata kekuatan gempa bumi 36 bulan dan rata-rata kekuatan gempa 36 bulan berikutnya digunakan sebagai target untuk prediksi. Untuk meng-update parameter (bobot) dari *Neural Network* digunakan metode *Gradient Descent* dan untuk mendapatkan parameter yang lebih optimal pada *layer Output*, maka di diterapkan *Algorithm Genetik*. Hasil peramalan dari kedua model kemudian dibandingkan dan model terbaik ditentukan dari nilai *Mean Square Error (MSE)* yang terkecil. dari hasil peramalan dengan model *ARIMA* diperoleh *MSE* sebesar 1.0125, sedangkan pada model *NN-GA* diperoleh *MSE* sebesar 0.9196. Nilai tersebut, menunjukkan bahwa model *NN-GA* lebih baik dari model *ARIMA* untuk peramalan rata-rata kekuatan gempa bumi beberapa bulan ke depan.

Kata kunci : Metode Peramalan Time Series *ARIMA*, *Neural Network (NN)*, *Gradient Descent* dan *Algorithm Genetik (GA)*.

ABSTRACT

Earthquake is a ground motion that occurs suddenly to cause vibration. The strength of earthquake magnitude can lead to serious damage and fatalities. To anticipate the impending disaster, then we need a model specifically for earthquake forecasting. This research, uses ARIMA model and a combination of Neural Network-Genetic Algorithm (NN-GA) to predict the average magnitude of earthquake in each periods. Particularly, in the area of North Maluku. The data used in this research is the average of earthquake measured by richter scale (SR) obtained from the Meteorology, Climatology and Geophysics (BMKG) Ternate city. The average magnitude of 36 months earthquake is used for input of the ARIMA model and the NN-GA models, whereas the next 36 months earthquake is used as target for the prediction. To update the parameters (weights) of Neural Networks the Gradient Descent method is used and to get an optimal parameters in the output layer, the Genetic Algorithm is applied. The forecasting results of the two models are then compared and the best model is determined from the smallest Mean Square Error (MSE). The result show that the MSE of forecasting based on ARIMA model is 1.0125. while the MSE of the NN-GA models is 0.9196. These values, indicate that the NN-GA models is better than ARIMA model to forecast the average of earthquake magnitude for the next several months.

Key words : Time Series *ARIMA* Forecasting, *Neural Network (NN)*, *Gradient Descent*, *Genetic Algorithm (GA)*

PENDAHULUAN

Secara geografis, kepulauan Indonesia terletak pada perbenturan tiga lempeng kerak

bumi yaitu lempeng Eruasia, lempeng Pasifik, dan lempeng India Australia. Ditinjau secara geologis, kepulauan Indonesia berada pada pertemuan dua jalur gempa utama, yaitu jalur gempa Sirkum

Pasifik dan jalur gempa Alpide Transasiatik sehingga menyebabkan wilayah Indonesia sangat rawan bahaya guncangan gempa bumi. Catatan Direktorat Vulkanologi dan Mitigasi Bencana Geologi (DVMBG) Departemen Energi dan Sumber Daya Mineral menunjukkan bahwa ada 28 wilayah di Indonesia yang dinyatakan rawan gempa dan tsunami salah satunya adalah wilayah maluku utara [1].

Gempa bumi adalah pergerakan tanah secara tiba-tiba yang terjadi di bumi hingga menimbulkan getaran yang disebut gelombang seismik. Besarnya kekuatan gempa yang terjadi pada Hiposentrum (pusat gempa) akan mengakibatkan terjadinya besaran getaran serta pengaruhnya yang berbeda pada daerah yang berbeda pula di permukaan bumi (Epiosentrum) [2]. Menurut [3] gempa bumi merupakan suatu kejadian yang tidak sepenuhnya tunggal, tetapi kejadian-kejadian tersebut saling berhubungan baik kekuatan, waktu dan tempat kejadian yang satu dengan yang lainnya. [4] menyatakan bahwa untuk menghitung atau memprediksi kondisi gempa pada masa yang akan datang dapat dilakukan dengan menggunakan pemodelan ANN (*Artificial Neural Network*). [5] mengaplikasikan ANN pada data time series untuk memprediksi sifat dari gempa bumi. Pengembangan model ANN telah banyak dilakukan dengan banyak cara seperti penggabungan dan optimalisasi struktur jaringan [6].

Neural Network (NN) telah digunakan secara luas dalam menyelesaikan permasalahan di bidang geofisika, kedokteran, meteorologi, sosial dan industri. [7] melakukan penelitian yang paling awal dalam penggunaan jaringan syaraf (*Neural Network*) untuk peramalan data time series. Kelebihan lain penggunaan *Neural Network* jika diterapkan pada data time series adalah bahwa data tidak perlu memenuhi asumsi sebaran seperti pada metode-metode statistika [8]. Terdapat dua tipe *Neural Network*, yaitu pelatihan dengan pengarahan (*supevised training*) dan pelatihan tanpa pengarahan (*Unsupevised training*). Salah satu metode *Neural Network* yang menggunakan algoritma pelatihan dengan pengarahan adalah *Neural Network* propagasi balik (*Backpropagation*) [9]. Untuk penentuan bobot pada hidden layer selama proses pelatihan menggunakan metode *Gradient Descent* [10]. Kemudian bobot di optimasi menggunakan *Algoritma Genetik (GA)* untuk menurunkan kesalahan yang terjadi pada hasil peramalan.

Tujuan penelitian ini adalah mengkaji cara kerja dari *ARIMA* dan *Neural Network - Algoritma Genetik*, memperoleh suatu output yang optimal pada *Neural Network* setelah parameternya (bobot) di optimasi menggunakan metode *Algoritma Genetik*, dan Membandingkan tingkat

akurasi yang dihasilkan oleh *ARIMA* dan *Neural Network - Algoritma Genetik* pada data *time series* dalam memprediksi rata-rata kekuatan terjadinya gempa bumi di wilayah Maluku Utara untuk n -periode kedepan.

DASAR TEORI

1. Analisis Time series

Time Series adalah serangkaian data pengamatan yang disusun menurut waktu, di mana data pengamatan tersebut bersifat acak dan saling berhubungan secara statistika. Analisis data deret waktu pada dasarnya digunakan untuk melakukan analisis data yang mempertimbangkan pengaruh waktu [11].

2. Pengujian Stasioneritas Time series

Suatu data dapat dikatakan stasioner apabila pola data tersebut berada pada kesetimbangan disekitar nilai rata-rata (*mean*) yang konstan dan variansi disekitar rata-rata tersebut konstan selama waktu tertentu [12].

a. Stasioner pada Varians

Bagi data yang tidak stasioner pada *varians* maka ditransformasikan terlebih dahulu. Transformasi data dapat dilakukan dengan metode transformasi Box-Cox [13], dengan bentuk sebagai berikut:

$$T(X_t) = X_t^{(\lambda)} = \begin{cases} \frac{X_t^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ \ln X_t, & \lambda = 0 \end{cases} \quad (1)$$

di mana :

λ : Parameter transformasi

$T(X_t)$: Data transformasi

X_t : Pengamatan pada waktu t

b. Stasioner pada Mean

Jika data tidak stasioner terhadap *mean* maka perlu dilakukan pembedaan (*differencing*). Operator *shift* mundur (*backward shift*) sangat tepat untuk menggambarkan proses *differencing* [12]. Penggunaan *backshift* adalah sebagai berikut:

$$BX_t = X_{t-1} \quad (2)$$

di mana:

X_t : nilai pengamatan pada waktu ke- t

X_{t-1} : nilai pengamatan pada waktu ke- $t-1$

B : *backshift*

Secara umum jika terdapat *differencing* sampai orde ke- d ditulis sebagai

$$X_t^d = \frac{(1-B)^d X_t}{1} \quad d \geq 1 \quad (3)$$

3. Koefisien Fungsi Autokorelasi (ACF)

Koefisien autokorelasi merupakan suatu alat penentu dari indentifikasi pola dasar yang menggambarkan data [14]. Menurut [15] proses

$$\rho_k = \frac{Cov(X_t, X_{t+k})}{\sqrt{Var(X_t)}\sqrt{Var(X_{t+k})}} = \frac{E(X_t - \bar{X})E(X_{t+k} - \bar{X})}{\sqrt{Var(X_t)}\sqrt{Var(X_{t+k})}} \quad (4)$$

Sedangkan untuk fungsi autokovariansi antara X_t dan X_{t+k} untuk *lag-k* sebagai berikut:

$$\begin{aligned} \gamma_k &= Cov(X_t, X_{t+k}) \\ &= E(X_t - \bar{X})E(X_{t+k} - \bar{X}) \end{aligned} \quad (5)$$

dimana γ_k disebut fungsi autokovariansi, untuk keadaan stasioner $Var(X_t) = Var(X_{t+k}) = \gamma_0$ sehingga persamaan (4) dan (5) menjadi:

$$\rho_k = \frac{\gamma_k}{\gamma_0} \quad (6)$$

Koefisien fungsi autokorelasi ρ_k diduga dengan koefisien autokorelasi data.

$$\rho_k = \frac{\sum_{t=1}^{n-k} (X_t - \bar{X})(X_{t+k} - \bar{X})}{\sum_{t=1}^{n-k} (X_t - \bar{X})^2} \quad (7)$$

4. Koefisien Fungsi Autokorelasi Parsial (PACF)

Autokorelasi parsial digunakan untuk mengetahui korelasi atau hubungan antara X_t dan X_{t+k} , apabila pengaruh dari *time lag* 1,2,3,...,k-j dianggap terpisah [12]. Menurut [11] taksiran dari PACF adalah berdasarkan koefisien autokorelasi pada persamaan *Yule-Walker* untuk *time lag-k* secara rekursif sebagai berikut:

$$\phi_{kk} = \frac{\rho_{kk} - \sum_{j=1}^{k-1} \phi_{k-1,j} \rho_{k-j}}{1 - \sum_{j=1}^{k-1} \phi_{k-1,j} \rho_j} \quad (8)$$

5. Pembentukan Model ARIMA

Model ini merupakan kombinasi dari model *Autoregressive* dan *Moving Average* ditambah dengan *Integrated* (Pembedaan).

Model *Autoregressive* (AR):

Model *Autogressive* dengan AR (p). Bentuk umum model AR (p) adalah:

stasioner suatu *time series* (X_t) diperoleh dari $E(X_t) = \bar{X}$ dan $Var(X_t) = (X_t - \bar{X})^2$ yang konstan dan kovariansi $Cov(X_t, X_{t+1})$. Dari koefisien autokorelasi antara X_t dan X_{t+1} untuk *lag-k* sebagai berikut:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + e_t \quad (9)$$

Persamaan (9) menggunakan operator B (*backshift*):

$$\phi_p(B)X_t = e_t \quad (10)$$

dengan $\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$.

Model *Moving Avarage* (MA):

Bentuk umum model MA(q) dapat ditulis sebagai berikut:

$$X_t = e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q} \quad (11)$$

Persamaan (11) menggunakan operator B (*backshift*):

$$X_t = \theta_q(B)e_t \quad (12)$$

dengan $\phi_q(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_q B^q$.

Model *Autoregresivve Moving Avarage* (ARMA):

Model merupakan suatu gabungan dari model AR (p) dan MA (q). bentuk umum model ARMA (p,q), yaitu

$$X_t = \phi_1 X_{t-1} + \dots + \phi_p X_{t-p} + e_t - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q} \quad (13)$$

Persamaan (13) menggunakan operator B (*backshift*):

$$\phi_p(B)X_t = \theta_q(B)e_t \quad (14)$$

Model *Autoregresivve Moving Avarage Integreted* (ARIMA):

Model ini merupakan gabungan dari model ARMA (p,q) dan *Differencing* (d), secara umum model ARIMA (p,d,q) untuk suatu data *time series* X_t adalah sebagai berikut [12]

$$\phi_p(B)(1 - B)^d X_t = \theta_q(B)e_t \quad (15)$$

dengan $\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$
 $\theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$

di mana:

$\phi_p(B)$: faktor *AR* (p)

$\theta_q(B)$: faktor *MA* (q)

$(1 - B)^d$: order *differencing*

6. Tahapan Pemodelan Autoregressive Integrated Moving Average (ARIMA)

Tahapan pemodelan metode *ARIMA* adalah:

1. Identifikasi Model

Langkah pertama dalam pembentukan model *ARIMA* adalah pembentukan plot data *time series*. Pembuatan plot data *time series* bertujuan untuk mendeteksi stasioneritas data *time series*.

2. Menentukan Orde Autoregressive (AR) dan Moving Average (MA)

Setelah data terbukti stasioner langkah selanjutnya adalah mengidentifikasi order *Autoregressive (AR)* dan *Moving Average (MA)* dapat dilakukan dengan cara melihat plot *ACF* dan *PACF* dari data tersebut.

3. Estimasi Parameter

Metode yang digunakan untuk mengestimasi parameter *ARIMA* ϕ dan θ yaitu metode kuadrat terkecil (*least square method*). Untuk mempermudah proses estimasi biasanya dilakukan dengan program Minitab.

4. Pemeriksaan Diagnostik

Setelah berhasil mengestimasi nilai-nilai parameter dari model *ARIMA* yang ditetapkan sementara, selanjutnya perlu dilakukan pemeriksaan diagnostik untuk membuktikan bahwa model tersebut cukup memadai dan menentukan model mana yang terbaik digunakan untuk peramalan [12].

a. Uji Signifikansi Parameter

Uji signifikansi parameter bertujuan untuk membuktikan bahwa model *ARIMA* tersebut cukup memadai untuk digunakan. uji signifikansi parameter model pada parameter yaitu dengan uji t .

b. Uji sisa White Noise

Suatu proses $\{e_t\}$ disebut *white noise* jika datanya terdiri dari variabel random yang tidak berkorelasi dan berdistribusi normal. Menguji sampel *residual* dari *ACF* tidak signifikan memakai statistik uji *Ljung Box* [14]

1. Hipotesis

$H_0 : \rho_1 = \rho_2 = \dots = \rho_K = 0$ (*white noise*)

$H_1 : \exists \rho_k \neq 0, k = 1, 2, \dots, K$ (tidak *white noise*)

2. Statistik uji yaitu uji Ljung Box

$$Q_K = n(n+2) \sum_{k=1}^K \frac{\rho_k^2}{(n-k)} \quad (15)$$

3. Kriteria Keputusan

Tolak H_0 jika $Q_K > \chi^2_{(\alpha, df)}$; $df = K - m$, di mana K banyaknya *lag* dan m banyaknya parameter atau p -value $< \alpha 0.05$, artinya e_t barisan yang tidak memiliki korelasi.

7. Konsep Dasar Pemodelan Artificial Neural Networks (ANN)

Neural Networks (NN) salah satu metode yang digunakan untuk pengenalan pola, *signal processing* dan peramalan, sebagaimana metode-metode lain yang terdapat dalam *NN* [16]. Model dari *Neural networks* terdiri atas 3 layer yaitu *layer input*, *hidden layer/kerne layer* dan *layer output* [17].

8. Metode Neural Network Backpropagation

Algoritma *backpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu [16].

Misalkan diberikan n inputan data yang berbeda x_i ($i = 1, 2, \dots, n$) yang dihubungkan dengan output data \hat{y}_k ($k = 1, 2, \dots, p$), maka interpolasi dari *Neural Networks* adalah:

$$\hat{y}_k = f^0 \left[\sum_{j=1}^m (w_j f_j^h \left(\sum_{i=1}^n v_{ji} x_i(k) + v_{0j} \right) + w_0 \right) \quad (16)$$

di mana:

x_i : variabel input sebanyak n

\hat{y}_k : nilai *Output Layer*

k : indeks banyak pasangan input-output

v_{ji} : bobot dari neuron ke- i pada *layer Input* yang menuju pada *Hidden Layer*

v_{0j} : bias dari neuron ke- j pada *Hidden Layer*

f_j^h : fungsi aktivasi neuron ke- j pada *Hidden Layer*

w_j : bobot dari neuron ke- j pada *Hidden Layer* yang menuju pada *Output Layer*

w_0 : bias pada neuron *Output Layer*

f^0 : fungsi aktivasi pada *Hidden Layer*

Fungsi aktivasi yang sering digunakan *Neural networks*, salah satunya fungsi *sigmoid biner* dengan interval di antara 0 dan 1 [10], sebagai berikut:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (17)$$

dengan turunannya

$$f(x)' = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1}{1+e^{-x}} \left(1 - \frac{1}{1+e^{-x}}\right) = f(x)(1-f(x)) \quad (18)$$

Langkah-langkah algoritma pelatihan untuk *Neural Network* adalah sebagai berikut:

1. Inisialisasi bobot (diberi nilai kecil secara acak)
2. Ulangi langkah 2 hingga 9 sampai kondisi iterasi terpenuhi.
3. Untuk masing-masing pasangan data pelatihan, lakukan langkah 3-8

Fase I : Propagasi Maju (*Feedforward*)

4. Masing-masing unit input $x_i (i = 1, 2, \dots, n)$ menerima sinyal masukan x_i dan sinyal tersebut diteruskan ke unit-unit bagian berikutnya.
5. Hitung semua output pada *hidden layer* $a_{j(k)}^h (k = 1, 2, \dots, n)$

$$g_{j(k)}^h = \sum_{i=1}^n v_{ji} x_{i(k)} + v_{0j} \quad (19)$$

kemudian menghitung fungsi aktivasi f_j^h pada *hidden layer*

$$a_{j(k)}^h = f_j^h(g_{j(k)}^h) = f_j^h \left(\sum_{i=1}^n v_{ji} x_{i(k)} + v_{0j} \right) \quad (20)$$

Bila fungsi aktivasi sigmoid maka:

$$f_j^h = \frac{1}{1 + e^{-g_{j(k)}^h}} \quad (21)$$

6. Hitung semua output pada *layer output* $a_{(k)}^0 (k = 1, 2, \dots, m)$

$$g_{(k)}^0 = \sum_{j=1}^m w_j a_{j(k)}^h + w_0 \quad (22)$$

kemudian menghitung fungsi aktivasi f^0 pada *layer output*

$$a_{(k)}^0 = f^0(g_{(k)}^0) = f^0 \left(\sum_{j=1}^m w_j a_{j(k)}^h + w_0 \right) \quad (23)$$

$$\text{Dengan } f^0 = \frac{1}{1 + e^{-g_{(k)}^0}}$$

Fase II : Propogasi Mundur (*Backpropagation*)

7. Masing-masing *layer output* $\hat{y}_k (k = 1, 2, \dots, p)$ menerima pola target (y_k) sesuai dengan pola input dan hitung nilai galatnya (δ_k^0).

$$\delta_k^0 = (y_k - \hat{y}_k) f^{0'}(g_{(k)}^0) \quad (24)$$

Karena fungsi $f^0(g_{(k)}^0) = \hat{y}_k$ menggunakan fungsi sigmoid maka:

$$f^{0'}(g_{(k)}^0) = f^0(g_{(k)}^0) (1 - f^0(g_{(k)}^0)) = \hat{y}_k (1 - \hat{y}_k) \quad (25)$$

Sehingga persamaan (24) di substitusikan ke persamaan (25) diperoleh:

$$\delta_k^0 = (y_k - \hat{y}_k) \hat{y}_k (1 - \hat{y}_k) \quad (26)$$

Hitung perubahan bobot w_{kj} dengan laju pembelajaran α .

$$\Delta w_{jk} = \alpha \delta_k^0 a_{j(k)}^h ; k = 1, 2, \dots, p ; j = 1, 2, \dots, m$$

Dan perubahan bobot bias w_{0k}

$$\Delta w_{0k} = \alpha \delta_k^0$$

8. Hitung faktor δ berdasarkan galat di setiap *hidden layer* $a_{j(k)}^h (k = 1, 2, \dots, n)$

$$\delta_j = \sum_{k=1}^p \delta_k^0 w_j \quad (27)$$

Faktor δ unit tersembunyi

$$\delta_{j(k)}^h = \delta_j f_j^{h'}(g_{j(k)}^h) = \delta_j a_{j(k)}^h (1 - a_{j(k)}^h) \quad (28)$$

Maka perubahan bobot v_{ji} dengan laju pembelajaran α .

$$\Delta v_{ji} = \alpha \delta_{j(k)}^h x_{i(k)} ; j = 1, 2, \dots, m ; i = 1, 2, \dots, n$$

Dan perubahan bobot bias v_{0j}

$$\Delta v_{0j} = \alpha \delta_{j(k)}^h$$

Fase III : Perubahan Bobot

9. Masing-masing *output layer* $\hat{y}_k (k = 1, 2, \dots, p)$ di *update* bobotnya. Setelah menurunkan algoritma *Gradient Descent*, diperoleh dua persamaan yang digunakan untuk mengupdate bobot w_j, w_0, v_{ji} dan v_{0j} sebagai berikut:

- a. Untuk *updating* bobot dan bias pada *Output Layer*

$$w_j(k+1) = w_j(k) + \alpha \sum_{k=1}^p \delta_k^0 a_{j(k)}^h \quad (29)$$

$$w_0(k+1) = w_0(k) + \alpha \sum_{k=1}^p \delta_k^0 \quad (30)$$

- b. Untuk *updating* bobot dan bias pada *Hidden Layer*

$$v_{ji}(k+1) = v_{ji}(k) + \alpha \sum_{k=1}^p \delta_{j(k)}^h x_{i(k)} \quad (31)$$

$$v_{0j}(k+1) = v_{0j}(k) + \alpha \sum_{k=1}^p \delta_{j(k)}^h \quad (32)$$

Dengan $\alpha > 0$ yang merupakan laju pembelajaran (*learning rate*)

9. Metode Algoritma Genetik (GA)

Algoritma Genetik (GA) merupakan suatu konsep komputasi yang pertama kali diutarakan oleh John Holland dari Universitas Michigan pada tahun 1975. *Algoritma Genetik* adalah algoritma yang dikembangkan dari proses pencarian solusi optimasi menggunakan pencarian acak, ini

terlihat pada proses pembangkitan populasi awal yang menyatakan sekumpulan solusi yang dipilih secara acak. Algoritma ini memanfaatkan proses seleksi secara alamiah yang dikenal dengan proses evolusi dan operasi genetika atas kromosom [18].

1. Inisialisasi

Langkah awal untuk memperoleh nilai parameter yang optimum dari *Neural networks* adalah dengan inisialisasi. Kromosom memiliki panjang variabel yang didefinisikan sebagai berikut:

$$chrom = \{x_{n1}, x_{n2}, \dots, x_{np}\} \tag{33}$$

di mana $x_{n1}, x_{n2}, \dots, x_{np}$ adalah kemungkinan calon-calon solusi sebanyak p dan $chrom$ dinamakan populasi. Dalam algoritma Genetika, x_{n1} dinamakan kromosom pertama dengan panjang n .

2. Fungsi Evaluasi

Fungsi evaluasi adalah fungsi yang menghitung nilai *fitness* di setiap kromosom, fungsi *fitness* adalah error antara target output dan saat output.

3. Fungsi Seleksi

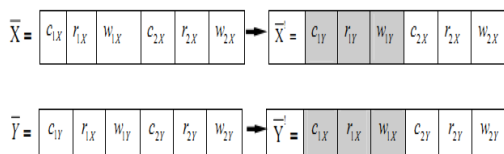
Untuk mendapatkan solusi yang terbaik, maka program harus menyeleksi solusi yang memiliki nilai *fitness* yang baik (*fitness* terbesar) dan mampu bertahan dalam populasi. Suatu metode seleksi yang umum digunakan adalah metode *roulette-wheel* (roda reulatte) [17]

4. Crossover dan Mutasi

Crossover adalah salah satu komponen paling penting dalam GA. Sebuah kromosom yang mengarah pada solusi yang bagus dapat diperoleh dari proses memindah-silangkan dua buah kromosom [18]. Persamaan crossover didefinisikan sebagai:

$$\begin{aligned} \bar{X}' &= r\bar{X} + (1-r)\bar{Y} \\ \bar{Y}' &= (1-r)\bar{X} + r\bar{Y} \end{aligned} \tag{33}$$

di mana \bar{X} dan \bar{Y} adalah dua vektor berdimensi- k yang dinotasikan individu (keluarga) dari populasi dengan $r \in [0,1]$. Dari persamaan ini, dapat disajikan dalam bentuk:

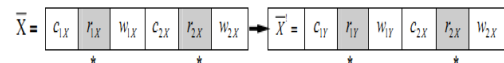


Gambar 1. Proses hitung crossover tiga titik dalam dua neutron

Mutasi berfungsi untuk menggantikan kromosom yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali kromosom yang tidak muncul pada inisialisasi populasi, salah satu metode mutasi yang digunakan adalah mutasi seragam sebagai berikut:

$$x_i = \begin{cases} U(a_i, b_i) & \text{jika } i = j \\ x_i & \text{yang lain} \end{cases} \tag{34}$$

di mana a_i batas bawah dan b_i batas atas untuk setiap variabel i . berikut adalah gambar yang menyajikan proses mutasi yang muncul di antara parameter dari *NN*.



Gambar 2. Mutasi seragam dari dua titik dalam dua neutron

METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini adalah:

1. Pengumpulan Data

Data yang digunakan dalam penelitian ini diperoleh dari Badan Meteorologi, Klimatologi dan Geofisika (BMKG) kota Ternate berupa kekuatan gempa dalam *skala richter* dari setiap kejadian gempa dengan kurun waktu Januari 2007 sampai dengan Desember 2012.

2. Variabel Penelitian

Berikut ini adalah variabel yang digunakan dalam penelitian ini.

1. Variabel input (x_i) adalah data rata-rata kekuatan gempa per periode pada waktu sebelumnya ($y_{t-1}, y_{t-2}, \dots, y_{t-n}$).
2. Variabel output (y_t) adalah data rata-rata kekuatan gempa per periode yang dijadikan data aktual/target.

3. Metode Analisis

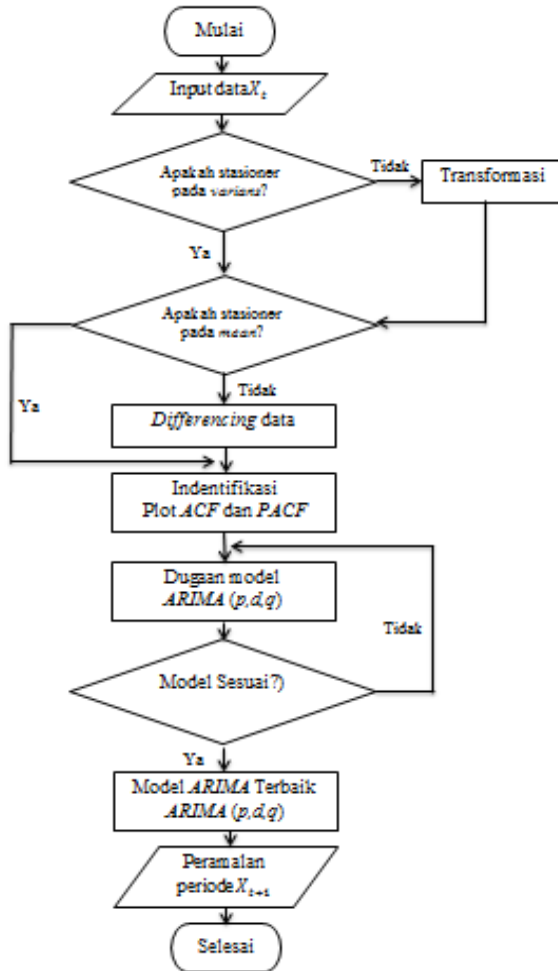
Peramalan dengan ARIMA

Adapun tahapan pengujian yang harus dilakukan untuk mendapatkan model *ARIMA* terbaik adalah:

- Membuat plot data yang ada secara grafis.
- Melakukan pengujian kestasioneran, apabila diketahui bahwa data tidak stasioner dalam *mean* maka perlu distasionerkan dengan cara pembedaan (*differencing*). Sedangkan apabila data tidak stasioner dalam *varians* maka dilakukan transformasi data. Jika data sudah stasioner maka dicari nilai *ACF* dan *PACF* dan merumuskan model umum *ARIMA* (p,d,q).
- Estimasi parameter model.

- *Diagnosis checking* terhadap model:
 - Uji signifikansi model dengan uji *t*
 - Uji kesesuaian model dengan asumsi kecukupan *white noise* dengan *Ljung Box*
- Jika model belum sesuai maka diulangi sampai kondisi terpenuhi.
- Menetapkan model terbaik untuk peramalan berdasarkan nilai *MSE* yang terkecil.

Setelah model yang tepat telah ditemukan maka peramalan untuk beberapa periode yang akan datang dapat ditentukan.



Gambar 3. Flowchart proses peramalan model ARIMA

Peramalan dengan Neural Network (NN)- Algoritma Genetik (GA)

Untuk mendapatkan model peramalan menggunakan model *Neural Network Backpropagation-Algoritma Genetik (GA)* dilakukan pelatihan (*training*) dan Peramalan (*testing*) dengan tahapan-tahapan sebagai berikut:

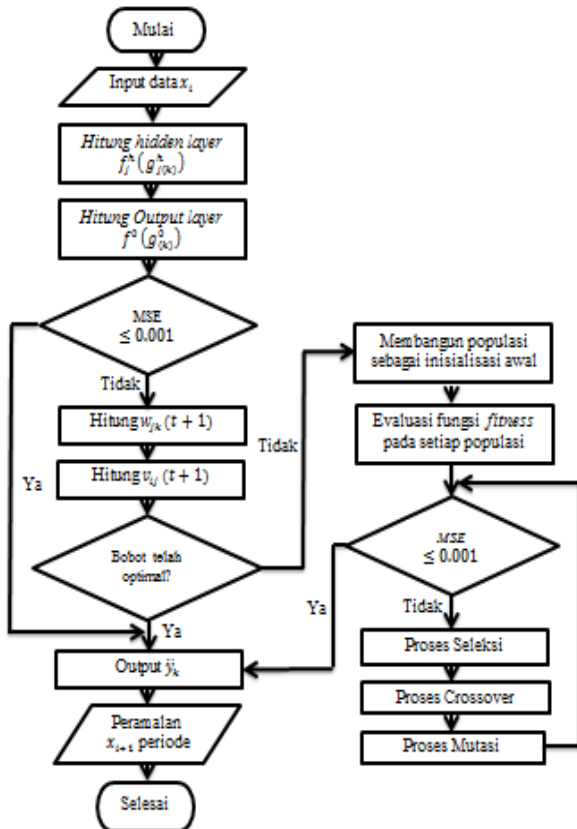
- Input yang digunakan adalah semua data rata-rata kekuatan gempa per periode.
- Algoritma *Backpropagation* (Propagasi Balik)

- Inisialisasi bobot
- *Feedforward* pola data input hingga mencapai lapisan output.
- Hasil pada lapisan output akan dibandingkan dengan nilai target dan di hitung nilai *Mean Square Error (MSE)*. Jika sudah terpenuhi sesuai dengan toleransi kesalahan, maka proses perhitungan dihentikan. Namun sebaliknya, jika kriteria belum terpenuhi maka dilanjutkan ke langkah berikutnya.

- Bergerak dari lapisan output kembali ke lapisan hidden dan melakukan penyesuaian bobot serta mengulangi.

- Optimasi parameter *Neural Network Backpropagation* dengan Algoritma Genetik, sebagai berikut:

- Inisialisasi Parameter
 - Inisialisasi populasi kromosom berdasarkan model peramalan.
- Fungsi Evaluasi
 - Evaluasi populasi kromosom untuk menentukan nilai kecocokan (*fitness*) suatu kromosom (solusi).
- Seleksi
 - Seleksi Kromosom solusi sesuai dengan nilai *fitnessnya* menggunakan metode *Roulette-Wheel* (roda roulette).
- Crossover dan Mutasi
 - Crossover kromosom induk untuk meningkatkan populasi menggunakan metode *one-cut point*. Mutasi gen untuk mendapatkan kromosom baru menggunakan metode mutasi seragam.
- Hingga maksimum generasi tercapai dengan toleransi kesalahan sebesar $\alpha = 0.001$.
- Hitung *Mean Square Error (MSE)* sebagai pengujian *Neural Network Backpropagation* yang merupakan selisih antara *output* jaringan dengan target dan simpan semua hasil pelatihan yang nantinya digunakan untuk peramalan.
- Membandingkan nilai *Mean Square Error (MSE)* hasil peramalan dari *Neural Network Backpropagation- Algoritma Genetik* dengan metode *ARIMA* dan *Neural Network Backpropagation* secara individu.
- Hasil Peramalan terbaik.



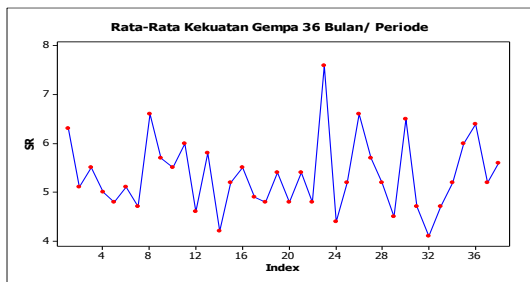
Gambar 4. Flowchart proses peramalan model NN-GA

HASIL DAN PEMBAHASAN

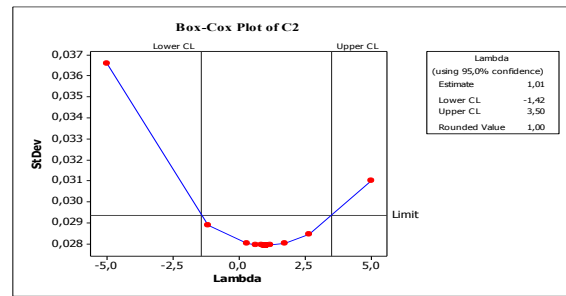
1. Peramalan Menggunakan ARIMA untuk 36 Bulan/Periode Kedepan

1. Identifikasi Model

Langkah pertama yang harus dilakukan untuk menganalisis data *time series* adalah membuat plot data rata-rata kekuatan gempa seiring waktu. Plot ini dibuat dengan tujuan untuk mendeteksi kestasioneran data baik dalam *mean* maupun *varians* secara grafis. Juga dapat digunakan untuk mengetahui kecenderungan rata-rata kekuatan gempa yang terjadi pada beberapa periode sebelumnya.



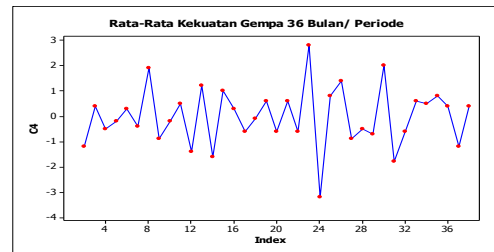
(a)



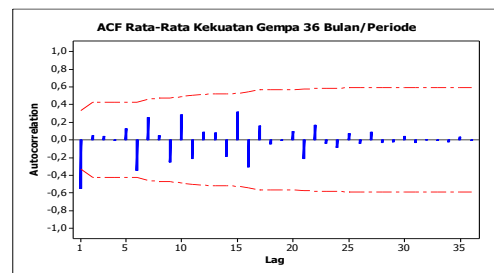
(b)

Gambar 5. (a) Plot *time series* Kekuatan Gempa (b). Plot *Box-Cox*

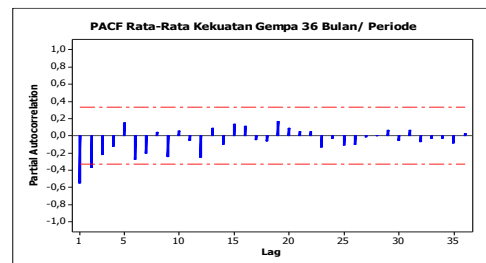
Pada Gambar 5 (a) terlihat bahwa terdapat perbedaan fluktuasi data rata-rata kekuatan gempa pada beberapa periode waktu, sehingga dicurigai bahwa terdapat sifat ketidastasioneran terhadap *varians* pada data tersebut. Akan tetapi nilai $\lambda = 1$ pada transformasi *Box-Cox* Gambar 5 (b), menunjukkan bahwa data masih memiliki sifat stasioneritas pada *varians*. Karena secara grafis terdapat perubahan rata-rata dari waktu ke waktu, maka perlu dilakukan *differencing* (pembedaan) untuk memenuhi sifat stasioneritas terhadap *mean* seperti pada gambar 6 (a).



(a)



(b)



(c)

Gambar 6. (a) Plot *time series* Kekuatan Gempa satu kali *differencing* (b). Plot *ACF* satu kali *differencing* (c) Plot *ACF* satu kali *differencing*

Berdasarkan plot *Autokorelasi Function (ACF)* dan plot *Partial Function autokorelasi (PACF)* pada Gambar 6 (b) dan 6 (c) terlihat bahwa nilai *autokorelasi* dari keduanya mempunyai satu *lag* yang signifikan yaitu pada *lag* ke-1. Sehingga dugaan sementara model *ARIMA (p,d,q)* berdasarkan plot *ACF* dan *PACF* adalah *ARIMA (1,1,0)*, *ARIMA (0,1,1)* dan diperoleh juga model yang lain yang merupakan gabungan dari kedua model tersebut yaitu *ARIMA (1,1,1)*.

2. Uji Signifikansi Parameter

Uji signifikansi parameter harus memenuhi syarat untuk mendukung ketepatan dalam pemilihan model, sehingga dilakukan pengujian sebagai berikut:

Tabel 1. Uji parameter untuk model *ARIMA (1,1,0)*, *ARIMA (0,1,1)* dan *ARIMA (1,1,1)*

Berdasarkan uji signifikansi pada Tabel 1 diketahui bahwa semua nilai parameter $|t_{hitung}| > \frac{t_{\alpha}}{2}$ atau $p\text{-value} < \alpha = 0.05$, maka H_0 ditolak (parameter tidak signifikan). Sehingga semua model memiliki nilai parameter yang signifikan.

3. *Diagnostic Checking Model*

Suatu model dikatakan sesuai apabila deret residualnya bersifat *white noise*. Pengujian *white noise* menggunakan statistik uji *Ljung-Box* dengan hipotesis H_0 adalah residual memenuhi syarat *white noise*.

Tabel 2. Uji *white noise* untuk model *ARIMA (1,1,0)*, *ARIMA (0,1,1)* dan *ARIMA (1,1,1)*

Berdasarkan Tabel 2 dapat di lihat bahwa nilai $Q_K < \chi^2_{(\alpha,df)}$ atau $p\text{-value} > \alpha = 0.05$ untuk setiap *lag*, maka H_0 di terima. Sehingga di simpulkan bahwa residual memenuhi syarat bersifat *white noise*.

4. Pemilihan Model Terbaik

Berdasarkan pengujian syarat-syarat di atas, dapat dijelaskan bahwa model dugaan *ARIMA (1,1,0)*, *ARIMA (0,1,1)*, dan *ARIMA (1,1,1)* adalah model yang sesuai. Langkah selanjutnya adalah membandingkan nilai *Mean Square Error (MSE)* dari ketiga model tersebut.

Tabel 3. Uji *MSE* untuk model *ARIMA (1,1,0)*, *ARIMA (0,1,1)* dan *ARIMA (1,1,1)*

Model <i>ARIMA</i>	<i>MSE</i>
<i>ARIMA (1,1,0)</i>	1,2320

<i>ARIMA (0,1,1)</i>	1,0607
<i>ARIMA (1,1,1)</i>	0,6505

Berdasarkan Tabel 3 maka dapat disimpulkan bahwa model *ARIMA (1,1,1)* adalah model yang terbaik untuk meramalkan nilai data beberapa bulan/periode ke depan dengan nilai *MSE (Mean Square Error)* terkecil yaitu 0,6505.

5. Peramalan *ARIMA* pada Rata-Rata Kekuatan Gempa Bumi

Setelah diperoleh model terbaik yaitu model *ARIMA (1,1,1)*, selanjutnya akan dilakukan peramalan rata-rata kekuatan gempa bumi untuk 36 Bulan/periode ke depan di wilayah Maluku Utara. Model *ARIMA (1,1,1)* secara matematis

Model	Koefisien	$ t_{hitung} $	$t_{\frac{\alpha}{2},df}$	<i>P-value</i>	Ket
<i>ARIMA (1,1,0)</i>	$\phi_1 = 0,5611$	2,94	2,021	0,008	Signifikan
<i>ARIMA (0,1,1)</i>	$\theta_1 = 0,8940$	5,16	2,021	0,000	Signifikan
<i>ARIMA (1,1,1)</i>	$\phi_1 = 0,1176$	2,47	2,021	0,043	Signifikan
	$\theta_1 = 0,9638$	4,43	2,021	0,000	Signifikan

dapat ditulis sebagai berikut:

$$\begin{aligned}
 (1 - \phi_1 B)(1 - B)X_t &= (1 - \theta_1 B)e_t \\
 (1 - \phi_1 B)(1 - B)X_t &= (1 - \theta_1 B)e_t \\
 (1 - \phi_1 B)(X_t - BX_t) &= e_t - \theta_1 Be_t \\
 (1 - \phi_1 B)(X_t - BX_t) &= e_t - \theta_1 Be_t \\
 X_t - BX_t - \phi_1 BX_t + \phi_1 B^2 X_t &= e_t - \theta_1 Be_t \\
 X_t - X_{t-1} - \phi_1 X_{t-1} + \phi_1 X_{t-2} &= e_t - \theta_1 e_{t-1} \\
 X_t = (1 + 0,1176)X_{t-1} - 0,1176X_{t-2} + e_t &- (0,9638e_{t-1})
 \end{aligned}$$

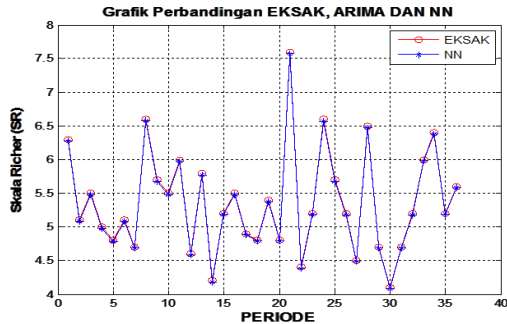
Model	lag	df	$\chi^2_{hitung} (Q_K)$	$\chi^2_{(\alpha,df)}$	<i>P-value</i>	Ket
<i>ARIMA (1,1,0)</i>	12	9	6,7	16,919	0,666	<i>White noise</i>
	24	21	7,0	32,671	0,998	<i>White noise</i>
<i>ARIMA (0,1,1)</i>	12	9	7,1	16,919	0,627	<i>White noise</i>
	24	21	13,1	32,671	0,905	<i>White noise</i>
<i>ARIMA (0,1,1)</i>	12	7	17,4	14,017	0,115	<i>White noise</i>
	24	19	26,3	30,144	0,122	<i>White noise</i>

2. Peramalan Menggunakan *NN-GA* untuk 36 Bulan/Periode Kedepan

1. Data training menggunakan *NN* dan *NN-GA* Untuk rata-rata gempa bumi 36 bulan/periode

Data *training* bertujuan untuk mempelajari pola data yang akan digunakan selanjutnya pada data *testing* (peramalan). Untuk menggambarkan

hasil simulasi data *training* di atas akan ditampilkan grafik antara data eksak dengan data hasil *training* dari *NN* dan *NN-GA* serta lamanya iterasi. Berikut ini adalah grafik hasil data training rata-rata gempa bumi 36 periode kedepan menggunakan metode *NN* secara individu.



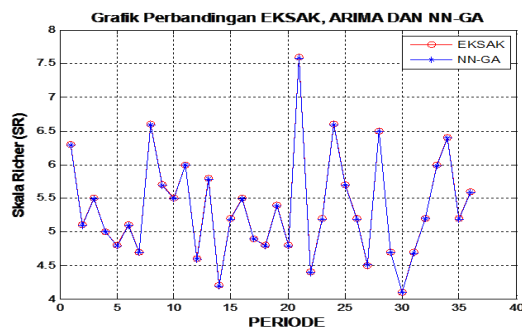
(a)



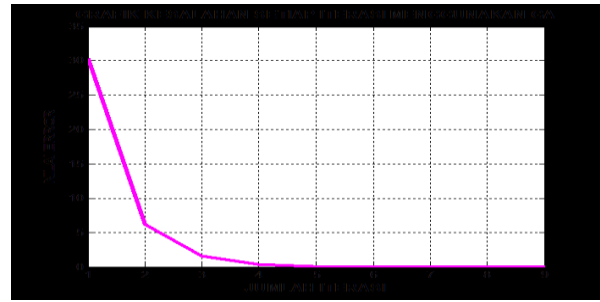
(b)

Gambar 7. (a) Perbandingan hasil data *training* rata-rata gempa 36 bulan/periode menggunakan model *NN* (b). Nilai *MSE* dari model *NN* pada setiap iterasi

Berikut ini adalah grafik hasil data *training* rata-rata gempa bumi untuk 36 periode menggunakan metode kombinasi *NN-GA*.



(a)



(b)

Gambar 8. (a) Perbandingan hasil data *training* rata-rata gempa 36 bulan/periode menggunakan model *NN-GA* (b). Nilai *MSE* dari model *NN-GA* pada setiap iterasi.

Gambar 7 (a) dan Gambar (b) menampilkan perbandingan kesalahan dari model *NN* dan *NN-GA*. Nilai kesalahan sendiri diperoleh dari nilai mutlak selisih prediksi yang dihasilkan oleh ke dua metode dengan nilai eksak. Tujuannya untuk melihat kesalahan hasil data prediksi dari ke dua metode tersebut, yang sebenarnya bisa juga dilihat melalui Tabel 4 karena lebih mudah untuk melihat kedekatan hasil prediksi yang dihasilkan dari ke dua metode dengan nilai eksak. Semakin kecil nilai kesalahan dari kedua model maka semakin dekat dengan nilai eksak yang diperoleh untuk 36 bulan/periode berikutnya.

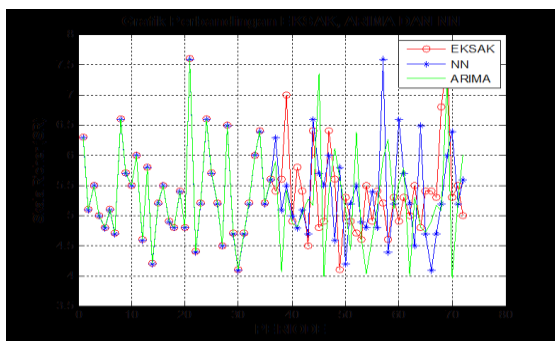
Berdasarkan hasil pada Tabel 5.4 diperlihatkan bahwa untuk data training rata-rata kekuatan gempa 36 bulan/periode, kombinasi model *NN-GA* lebih baik dibandingkan dengan hasil model *NN* secara individu. Hal ini dapat dilihat nilai *MSE* pada *NN-GA* sebesar 0.00016 yang secara keseluruhan lebih kecil dibandingkan dengan nilai *MSE* dari *NN* sebesar 0.000093, serta jumlah iterasi yang berpengaruh terhadap lamanya waktu komputasi, perbandingan iterasi dari kedua model tersebut yaitu *NN-GA* hanya membutuhkan 9 kali iterasi sedangkan *NN* membutuhkan 12 kali iterasi.

2. Data *Testing* menggunakan metode *ARIMA*, *NN* dan *NN-GA* untuk peramalan rata-rata gempa bumi 36 bulan/periode kedepan

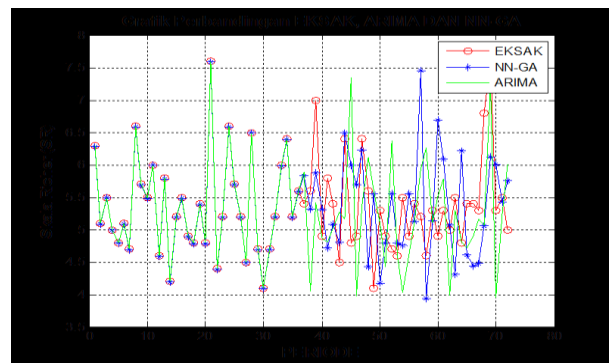
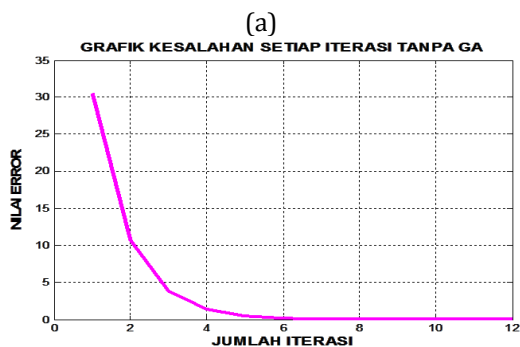
Untuk menggambarkan hasil simulasi di atas akan ditampilkan grafik perbandingan antara data aktual dengan data hasil prediksi dari *ARIMA*, *NN* dan *NN-GA*. Berikut ini adalah grafik data hasil peramalan rata-rata gempa bumi 36 bulan /periode kedepan tanpa menggunakan *GA*.

Tabel 4. Perbandingan hasil prediksi dan nilai kesalahan dari model *NN* dan *NN-GA* untuk data *training* 36 bulan/periode

Hasi Training 12 Bulan/ Periode							
Per	Akt	<i>NN</i>	<i>NN-GA</i>	Per	Akt	<i>NN</i>	<i>NN-GA</i>
1	6.3	6.28	6.29	19	5.4	5.38	5.39
2	5.1	5.08	5.091	20	4.8	4.78	4.79
3	5.5	5.48	5.489	21	7.6	7.58	7.588
4	5.0	4.99	4.992	22	4.4	4.39	4.390
5	4.8	4.78	4.789	23	5.2	5.18	5.192
6	5.1	5.08	5.094	24	6.6	6.58	6.588
7	4.7	4.69	4.686	25	5.7	5.68	5.685
8	6.6	6.58	6.590	26	5.2	5.18	5.192
9	6.7	5.68	5.690	27	4.5	4.48	4.493
10	5.5	5.48	5.489	28	6.5	6.48	6.485
11	6.0	5.98	5.986	29	4.7	4.68	4.692
12	4.6	4.58	4.592	30	4.1	4.08	4.093
13	5.8	5.78	5.789	31	4.7	4.68	4.691
14	4.2	4.19	4.195	32	5.2	5.19	5.189
15	5.2	5.18	5.192	33	6.0	5.98	5.989
16	5.5	5.48	5.493	34	6.4	6.81	6.387
17	4.9	4.88	4.888	35	5.2	5.18	5.188
18	4.8	4.78	4.793	36	5.6	5.58	5.591
MSE	<i>NN</i>			<i>NN-GA</i>			
	0.00016			0.000093			



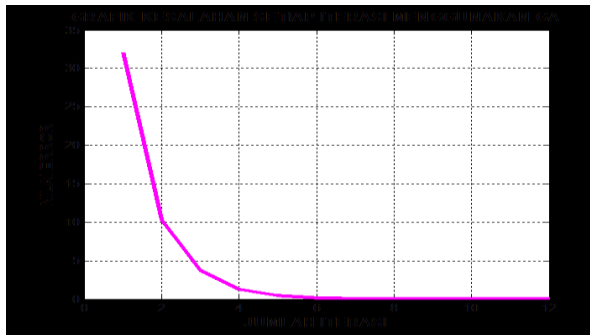
Gambar 9. (a) Perbandingan data hasil peramalan rata-rata gempa 36 bulan/periode kedepan menggunakan model *ARIMA* dan *NN* (b). Nilai *MSE* dari model *ARIMA* dan *NN* pada setiap iterasi



(b)

(a)

Gambar 9 (a) dan Gambar 10 (a) menampilkan perbandingan kesalahan dari model yaitu *ARIMA* dan *NN-GA*. Nilai kesalahan sendiri diperoleh dari nilai mutlak selisih prediksi yang dihasilkan oleh kedua metode dengan nilai eksak.



(b)

Gambar 10. (a) Perbandingan hasil peramalan rata-rata gempa 36 bulan/periode kedepan menggunakan model *ARIMA* dan *NN-GA* (b). Nilai *MSE*

dari model *ARIMA* dan *NN-GA* pada setiap iterasi.

Tujuannya untuk melihat kesalahan hasil prediksi dari kedua metode tersebut, yang sebenarnya bisa juga dilihat melalui Tabel 5 karena lebih mudah untuk melihat kedekatan hasil prediksi yang dihasilkan dari ke tiga metode dengan nilai eksak. Semakin kecil nilai kesalahan dari kedua model maka semakin dekat dengan prediksi yang diperoleh untuk 36 periode berikutnya.

Gambar 9 (b) dan 10 (b) menampilkan perbandingan rata-rata waktu komputasi oleh masing-masing algoritma untuk menyelesaikan tiap-tiap data uji coba. Dapat dilihat bahwa algoritma *NN-GA* mendapatkan waktu komputasi yang sama dengan *NN-GA*, namun menghasilkan prediksi yang berbeda. Hal ini disebabkan karena solusi yang didapat dengan menggunakan *NN-GA* lebih baik dibandingkan dengan *NN* secara individu.

Tabel 5 Perbandingan hasil prediksi dan kesalahan dari model *ARIMA* dan *NN-GA* untuk data *testing* (peramalan) 36 bulan/periode kedepan

Hasi Peramalan 36 Bulan/Periode							
Per	Akt	<i>ARIMA</i>	<i>NN-GA</i>	Per	Akt	<i>ARIMA</i>	<i>NN-GA</i>
37	5.4	5.58	6.28	55	4.9	4.62	5.38
38	5.6	4.05	5.08	56	5.4	5.10	4.78
39	7.0	5.41	5.48	57	5.2	5.94	5.75
40	4.9	4.86	4.98	58	4.6	6.25	4.38
41	5.8	4.80	4.78	59	5.3	5.11	5.18
42	5.4	5.01	5.08	60	4.9	5.52	6.58
43	4.5	5.26	4.68	61	5.3	5.78	5.68
44	6.4	5.16	6.57	62	5.0	3.98	5.19
45	4.8	7.34	5.68	63	5.5	5.30	4.48
46	4.9	3.98	5.48	64	4.8	4.81	6.47
47	6.4	5.35	5.98	65	5.4	4.71	4.69
48	5.6	6.11	4.58	66	5.4	4.90	4.08
49	4.1	5.61	5.78	67	5.3	5.16	4.68
50	5.3	5.08	4.19	68	6.8	5.07	5.18
51	4.9	4.43	5.18	69	7.6	7.22	5.98
52	4.7	6.36	5.48	70	5.3	3.95	6.38
53	4.6	4.61	4.88	71	5.5	5.26	5.18
54	5.5	4.03	4.78	72	5.0	6.00	5.58
<i>MSE</i>	<i>ARIMA</i>			<i>NN-GA</i>			
	1.0125			0.9196			

Berdasarkan hasil pada Tabel 5.5 diperlihatkan bahwa untuk data peramalan rata-rata kekuatan gempa bumi 36 periode berikutnya, kombinasi model *NN-GA* lebih baik dibandingkan dengan hasil model *ARIMA* dan *NN* secara individu. Hal ini dapat dilihat nilai *MSE* pada *NN-GA* sebesar 0.9196 yang secara keseluruhan lebih kecil dibandingkan dengan nilai *MSE* dari *ARIMA* sebesar 1.0125 dan *NN* sebesar 0.9664 secara individu serta jumlah iterasi yang sama yaitu 12 kali iterasi.

KESIMPULAN

Kesimpulan yang didapatkan pada penelitian ini adalah model *ARIMA* yang terbaik adalah model *ARIMA* (1,1,1) dan berdasarkan hasil *training* model kombinasi *Neural Network- Algoritma Genetik* lebih baik daripada model *Neural Network* untuk peramalan. Pemilihan parameter-parameter dan jumlah iterasi dapat berpengaruh terhadap performa dari *Neural Network - Algoritma Genetik*. Performa yang dimaksud adalah ketepatan prediksi berdasarkan nilai *MSE* yang kecil dan lamanya waktu komputasi. Hasil yang diperoleh menunjukkan bahwa kombinasi dari metode *Neural Network - Algoritma Genetik* dalam memprediksi rata-rata Gempa Bumi di wilayah Maluku Utara pada bulan/periode berikutnya secara keseluruhan lebih baik dari pada metode *ARIMA* secara individu berdasarkan kriteria nilai *MSE* terkecil.

BIBLIOGRAPHY

- [1] "Badan Meteorologi, Klimatologi dan Geofisika."
- [2] N. Morris and S. Handoko, *Gempa bumi*. Elex Media Komputindo, 2002.
- [3] Y. Altinok and D. Kolcak, "An application of the semi-Markov model for earthquake occurrences in North Anatolia, Turkey," *J. Balk. Geophys. Soc.*, vol. 2, no. 4, pp. 90-99, 1999.
- [4] A. C. C. Ronen, S. Schultz, P.S.Hattori M., *Seismic-guide estimation of log properties*. 1994, p. Part 1, 2, and 3.
- [5] B. H. Russell, L. R. Lines, and D. P. Hampson, "Application of the radial basis function neural network to the prediction of log properties from seismic attributes," *Explor. Geophys.*, vol. 34, no. 1/2, pp. 15-23, 2003.
- [6] Y. Zhangang, C. Yanbo, and K. W. E. Cheng, "Genetic algorithm-based RBF neural network load forecasting model," in *Power Engineering Society General Meeting*, 2007, pp. 1-6.
- [7] A. Lapedes and R. Farber, "Nonlinear signal processing using neural networks," 1987.
- [8] C. M. Bishop, "Neural networks for pattern recognition," 1995.
- [9] E. Yoan, *Pengembangan Metode Jaringan Syaraf Tiruan dengan Fungsi Radial Basis (FBR) Fuzzy dan Aplikasinya*. Depok: Universitas Indonesia.
- [10] D. R. Sari, "ANALISA ALGORITMA BACK PROPAGATION DENGAN METODE GRADIEN DESCENT UNTUK BEBAN LISTRIK JANGKA PENDEK."
- [11] J. D. Cryer, *Time series analysis*. Wadsworth Publ. Co., 1986.
- [12] S. Makridakis, S. C. Wheelwright, and V. E. McGee, "Metode dan aplikasi peramalan," *Bin. Aksara. Jakarta*, 1999.
- [13] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time series analysis: forecasting and control*. John Wiley & Sons, 2013.
- [14] L. Arsyad, "Peramalan Bisnis," *Ghalia Indones. Jakarta*, 1995.
- [15] W. W.-S. Wei, *Time series analysis*. Addison-Wesley publ, 1994.
- [16] J. J. Siang, "Jaringan syaraf tiruan dan pemrogramannya menggunakan matlab," *Penerbit Andi, Yogyakarta*, 2005.
- [17] Suhartono, *Feedforward Neural Network untuk Peramalan Time Series*. Yogyakarta: Penerbit Andi, 2007.
- [18] P. Andrey, "Genetic Algorithm For Optimization. Programs For Matlab,," UGM Yogyakarta.