Comparison of ARCH / GARCH model and Elman Recurrent Neural Network on data return of closing price stock

Vania Orva Nur Laily¹, Budi Warsito¹, Di Asih I Maruddani¹

¹Department of Statistics, Faculty of Science and Mathematics, Diponegoro University
Jl. Prof. Soedharto, SH, Tembalang, Semarang 50275, Indonesia
E-mail: vaniaorva@gmail.com

Abstract. Time series data can be modeled using the Autoregressive Intergrated Moving Average (ARIMA) model. ARIMA model requires the assumption of homoscedasticity. But on financial data, like stock data, has a high fluctuation so that the variant is heteroscedastic. To solve this problem, Autoregressive Conditional Heteroscedasticity (ARCH) or Generalized Autoregressive Conditional Heteroscedasticity (GARCH) model is used. The ARCH process introduced in Engle allows the conditional variance to change over time as a function of past errors leaving the unconditional variance constant. The ARCH model was generalized to the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model and introduced in Bollerslev allowing for a much more flexible lag structure. Another model that can be used is the Elman Recurrent Neural Network model (ERNN) which is a nonlinear model. The purpose of this research is to determine the best model between ARCH / GARCH model and ERNN model based on MSE value. The data used is the return of daily closing price of BRI stocks period of January 1st, 2015 to 17th February 2017. In this study, the best model obtained is GARCH (1,1) which has the smallest MSE value.

Keywords: Closing price of stock, return, heteroscedasticity, ARCH/GARCH, ERNN

1. Introduction

Time series data can be modeled with the model Autoregressive Moving Average (ARMA). ARMA model requires homoscedasticity assumptions to be met. But the financial data, such as the stock data, have high fluctuations so that the variants are heteroscedastic. One of time series model to resolve such issues is Autoregressive Conditional Heteroscedasticity (ARCH). ARCH model is generalized into a Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model by Bollerslev in 1986, GARCH used to overcome high order in ARCH models.

Another models that can be used is a neural network (NN) model. Generally, there are three kinds of NN based of the architecture that is single layer neural network, multi-layer neural network, and recurrent neural network. According to Kusumadewi [6], recurrent network is a network that accommodates output to be input on that network again in order to generate the next output. One of recurrent network model is Elman Recurrent Neural Network. It is a network that uses a semi-recursive backpropagation algorithm to find data patterns [4]. Elman Recurrent Neural Networks add a layer to receive feedback from the hidden layers to slow down the system so that the system can adapt to the variation in time which directly reflects the characteristics of a dynamic process. Given the dynamics of stock price movements, Elman Neural Networks suitable to build a non-linear models for time series data of stock prices.
Research using ARIMA only limited to providing mean models in a time series process while in the process of time series also consider variants. As has been explained previously that the financial data, such as stocks, have high fluctuations so that the variants are heteroscedastic, Therefore, researcher is interested in using GARCH and Elman Recurrent Neural Network to model returns the closing price of BRI stocks indicated heterocedastic effects and choose the best model based on the smallest of Mean Square Error (MSE).

2. Generalized Autoregressive Conditional Heteroscedasticity (GARCH)

In ARCH models, residual variance of time series data is not only influenced by independent variables but also influenced by the residual value of the variable itself [11]. Generally it can be expressed in the following equation:

$$\sigma_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \alpha_2 a_{t-2}^2 + \cdots + \alpha_p a_{t-p}^2$$  \hspace{1cm} (1)

Furthermore, the model developed by Bollerslev in 1986 became the Generalized Autoregressive Conditional Heteroscedasticity (GARCH) model. GARCH is one of the time series models can be used to describe the dynamic of the volatility function (standard deviation) of the data [8]. The general form GARCH (r, s) is as follows:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{r} \alpha_i a_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2$$  \hspace{1cm} (2)

3. Elman Recurrent Neural Network (ERNN)

According to Kusumadewi [5] Artificial Neural Network (ANN) is one of the artificial representation of the human brain that trying to stimulate the learning process of the human brain. According to Kusumadewi [5], the structure of the neural network neurons in Figure 1 is similar to a biological neuron cells in humans. The artificial neurons work in the same way with biological neurons. Information (input) will be sent to the neurons with a specific weight of arrival. This input will be processed by a propagation function which will add up the values of all weights. The sum will be compared to a threshold value through the activation function of each neuron. If the neuron is activated, it will send the output neurons by weights to all the neurons associated with it.

$$\sum \text{Input from the input layer neurons} \rightarrow \text{Weight} \rightarrow \text{Activation Function} \rightarrow \text{Output for the neurons in the next layer}$$

**Figure 1.** Structure of Neuron Neural Network

Recurrent Neural Network is a neural network having a feedback connection. According to Haykin [3], Elman Recurrent Neural Network is a neural network that has a simple connection feedback from the hidden layer to the layer of context unit consisting of operators of time delaysunit. According to Sani [7], a common model of the RNN network with one hidden layer, with q input unit and p unit in the hidden layer is

$$\hat{Y} = f^0 \left[ \beta_0 + \sum_{j=1}^{p} \left( \beta_j f^h (y_{j0} + y_{adj} + \sum_{i=1}^{q} y_{ij} x_i) \right) \right]$$  \hspace{1cm} (3)

Where, $\beta_0$ is bias weight in the hidden layer, $\beta_j$ is weight for unit j in the hidden layer, $y_{ji}$ is weight of the $i^{th}$ inputs in the $j^{th}$ unit of the hidden layer, $y_{adj}$ is delay weight, $y_{j0}$ is bias weight in the input layer, $f^h(x)$ is activation function of the hidden layer, and $f^0(x)$ is activation function in the output layer. According to Haykin [3], the simple RNN architecture can be seen in Figure 2.
Figure 2. Architecture of Simple Recurrent Neural Network

Steps of the training with backpropagation algorithm that used on the ERNN are as follows:
1) Initialize all weights by using the smallest random value
2) Do the following steps if the epoch <Maximum Epoch and MSE> Target Error
   a) Phase 1 (Feedforward)
      Each neuron input \((x_k, k = 1, 2, 3 \ldots n)\) received signal \(x_k\) and forwards the signal to all the neurons in layer thereafter, the hidden layer. The signals received by hidden layer from neuron input layer is formulated in the following equation:
      \[
y_{in(1)p} = b_p + \sum_{k=1}^{n} x_k w_{kp}(x) \tag{4}
\]
      where, \(y_{in(1)p}\) is signal received by hidden layer from neuron input layer, \(b_p\) is weight bias on the \(k^{th}\) neuron in the hidden layer with \(p = 1, 2, \ldots l\), \(x_k\) is input variables with \(k = 1, 2, \ldots, n\), and \(w_{kp}(x)\) is weights from the \(k^{th}\) input layer towards \(p^{th}\) neuron in the hidden layer.

      The \(y_{in(1)p}\) signal output is calculated using the activation function defined in the following equation:
      \[
y_{(1)p} = f(y_{in(1)p}) \tag{5}
\]
      Each context neuron \((u_p, p = 1, 2, 3 \ldots l)\) receives the signal and forwards the signal to the hidden layer. The received signal by context neurons is the result of the activation of the hidden layer output signals as defined in the following equation:
      \[
u_p = y_{(1)p} \tag{6}
\]
      Then context neurons sends back a signal to the hidden layer neurons as defined in the following equation:
      \[
y_{in(2)l} = \sum_{p=1}^{l} u_p w_{pp(u)} \tag{7}
\]
      where, \(y_{in(2)p}\) is received signal by the hidden layer from context neurons, \(u_p\) is output signal (activated) received by the hidden layer from input neuron with \(p = 1, 2, \ldots, l\), and \(w_{pp(u)}\) is weights from the \(p^{th}\) input layer (neurons context) toward \(p^{th}\) neurons in the hidden layer.

      Each neuron in the hidden layer \((p = 1, 2, \ldots l)\) sums up the weighted input signals obtained from the input layer and neurons context. The sum of these signal formulated in the following equation:
      \[
y_{inp} = y_{in(1)p} + y_{in(2)p} = b_p + \sum_{k=1}^{n} x_k w_{kp}(x) + \sum_{p=1}^{l} u_p w_{pp(u)} \tag{8}
\]
      where, \(y_{inp}\) is received signal by the hidden layer from the input layer and context neurons, \(y_{in(1)p}\) is received signal by the hidden layer from neuron input layer, and \(y_{in(2)p}\) is received signal by the hidden layer from context neurons. Notation \(b_p\) is \(p^{th}\) weight bias in the hidden layer with \(p = 1, 2, \ldots, l\), \(x_k\) is input variables with \(k = 1, 2, \ldots, n\), \(w_{kp}(x)\) is weight of the \(k^{th}\) input layer heading into the \(p^{th}\) hidden, \(u_p\) is output signal (activated) that received by the hidden layer from input neuron \(p = 1, 2, \ldots, l\), and \(w_{pp(u)}\) is weights from the \(p^{th}\) input layer (context neurons) toward \(p^{th}\) neurons in the hidden layer.
The \( y_{in_p} \) output signal is calculated using the activation function defined in the following equation:

\[
y_p = f(y_{in_p})
\]  

(9)

then the signal is sent to the neurons in the output layer. Activation function used from the input layer to the hidden layer in this thesis is tansig, so that Equation (4) and Equation (9) becomes

\[
y_{(1)p} = \frac{1-e^{-\sigma y_{in}(1)p}}{1+e^{-\sigma y_{in}(1)p}} \quad \text{and} \quad y_p = \frac{1-e^{-\sigma y_{in_p}}}{1+e^{-\sigma y_{in_p}}}
\]

Each neuron output \((Z)\) sums up the weighted input signals from the hidden layer as defined in the following equation:

\[
z_{in} = b_0 + \sum_{p=1}^{l} v_p y_p
\]  

(10)

where, \( b_0 \) is bias on the output neurons, \( v_p \) is weight from \( p^{\text{th}} \) neuron in the hidden layer toward the output layer with \( p = 1, 2, \ldots, l \) and \( y_p \) is the output signal has been activated from a hidden layer with \( p = 1, 2, \ldots, l \).

The \( z_{in} \) output signal is calculated using the activation function defined in the following equation:

\[
z = f(z_{in})
\]  

(11)

Activation function that used from the input layer to the output layer in this thesis is purelin. So the equation (11) turns into the following equation:

\[
z = z_{in}
\]  

(12)

b) Phase 2 (Backpropagation)

1) Neuron output \((z)\) receiving a target pattern that associated with the input pattern of learning, then calculated the error information. The calculation of error value defined in the following equation:

\[
\delta_0 = (tr - z)f'(z_{in})
\]  

(13)

\( tr \) = Targets to be achieved

Then calculate correction of weights that will be used to improve the \( v_p \) value and the bias correction that will be used to correct the \( b_0 \) value. Calculation of weight correction is defined in the following equation:

\[
\Delta v_p = \alpha \delta_0 y_p
\]  

(14)

\[
\Delta b_0 = \alpha \delta_0
\]  

(15)

with \( \alpha \) is rate of acceleration. Then \( \delta_0 \) transmitted to the neurons in the previous layer (hidden layer). This step was taken as the number of neurons in the hidden layer.

2) Each neuron hidden \((y_p, p = 1, 2, \ldots, l)\) sums up the delta inputs from neurons that located in the upper layer (output layer) are formulated in the following equation:

\[
\delta_{in_p} = \delta_0 v_p
\]  

(16)

Then multiply the \( \delta_{in_p} \) value by the derivative of the activation function of the output signal. This multiplication is formulated in the following equation:

\[
\delta_p = \delta_{in_p} f'(y_{in_p})
\]  

(17)

Then calculate the correction of weights that will be used to improve the \( w_{kp(x)} \) and \( w_{pp(i)} \) weight value. Calculation of weight correction is defined in the following equation:

\[
\Delta w_{kp(x)} = \alpha \delta_p x_k
\]  

(18)

\[
\Delta w_{pp(i)} = \alpha \delta_p u_p
\]  

(19)

In addition, calculate the weight bias correction that will be used to correct the \( b_p \) value. The calculation of weight correction is defined in the following equation:

\[
\Delta b_p = \alpha \delta_p
\]  

(20)

c) Phase 3 (Weight Change)

1) Neuron output \((z)\) fix bias and weight by the formula in the following equation:

\[
v_p(new) = v_p(old) + \Delta v_p
\]  

(21)
\[ b_0(\text{new}) = b_0(\text{old}) + \Delta b_0 \]  

2) Each hidden layer neuron \((y_p, p = 1,2, \ldots l)\) fix bias and weight by the formula in the following equation:

\[
\begin{align*}
    w_{kp}(\text{new}) &= w_{kp}(\text{old}) + \Delta w_{kp} \\
    w_{pp}(\text{new}) &= w_{pp}(\text{old}) + \Delta w_{pp} \\
    b_{p}(\text{new}) &= b_{p}(\text{old}) + \Delta b_{p}
\end{align*}
\]

4. Measuring Error of Forecasting

Error of forecasting can be calculated by using the difference between the original value and its forecasting value. One of the best model selection criteria is the smallest MSE value with the following MSE formula [10].

\[
MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{Z}_i - Z_i)^2
\]

Where, \(\hat{Z}_i\) is a vector of \(n\) predictions and \(Z_i\) is the vector of observed values of the variable being predicted.

5. Results and Discussion

The data used is the return of daily closing price of BRI stocks period of 1\(^{st}\) January 2015 to 17\(^{th}\) February 2017. The data got from the site http://finance.yahoo.com/.

The initial step is to plot the data closing price of stock. The plot aimed to determine the characteristics of the data movement closing price of stock.

![Figure 3. Plot the data stock closing price](image)

Plot closing price of stock data can be seen in Figure 3. The next step is to plot the returns of closing price stock. Use of return gives a lot of information, especially in knowing the stock fluctuations.
Figure 4. Plots stock closing price return data

Plot of return the closing price stock can be seen in Figure 4. Further examination is check stationary of data with Augmented Dickey-Fuller test. Here are the results of Augmented Dickey-Fuller test:

| ADF count | critical Value | Probability |
|-----------|----------------|-------------|
| -20.4452  | -1.9411        | 0.001       |

Based on the calculation results in Table 1 it can be concluded that at the 5% significance level $H_0$ is rejected, which means the returns of closing price stock of BRI is stationary. Once the data is proven stationary then performed model identification.

To identify the ARIMA model that may be seen in the ACF and PACF plots. In the ACF plot shows that the lag-1 exceeds the standard limit of error, so that the MA model probably MA (1). The PACF plot shows that the lag-1 exceeds the standard limit of error, so that the AR model probably an AR (1). Then the Box Jenkins models identification are probably the AR (1), MA (1), and ARMA (1,1).

Once a suitable AR and MA orde is set to obtain a temporary forecasting model, the next step is estimate the parameters. The results of the estimation parameters such as the following:

| Model     | Parameter | Value        | t-statistics | Decision          |
|-----------|-----------|--------------|--------------|-------------------|
| AR (1)    | Constant  | -3.94481 x 10^-5 | -0.0471227  | $H_0$ is accepted |
|           | $\phi_1$ | 0.132089     | 3.70197     | $H_0$ is rejected |
| MA (1)    | Constant  | -4.39673 x 10^-5 | -0.0459706  | $H_0$ is accepted |
|           | $\theta_1$ | 0.142921   | 3.93982     | $H_0$ is rejected |
| ARMA (1,1)| Constant  | -4.66182 x 10^-5 | -0.0453479  | $H_0$ is accepted |
|           | $\phi_1$ | -0.0860492  | -0.360544   | $H_0$ is accepted |
|           | $\theta_1$ | 0.226043   | 0.969404    | $H_0$ is accepted |
Based on the calculation results in Table 3 is known that $H_0$ is rejected for the model parameters AR (1) and the model MA (1), which means each of the parameters of the AR model (1) and MA (1) is significant. As for the ARMA model (1.1) for all the parameters $H_0$ is accepted so that parameter is not significant to the model.

Then do the verification of the model on the model AR (1) and MA (1), which includes testing the independence of residuals and residual normality test. Based on residual independence test has been done it is concluded that at the 5% significance level there is no correlation between the lagged residuals for all models possible. Then the normality test is concluded that the residuals are not normally distributed for all models.

The next step is testing the effects of ARCH / GARCH on the residual model of an AR (1). According to Tsay [9], use the Lagrange Multiplier test with the following results:

| Table 3. Test Lagrange Multiplier |
|----------------------------------|
| value $F$ | critical Value | Probability |
|---------|----------------|-------------|
| 5.4988  | 3.8415         | .0190       |

Based on the results of the calculations in Table 3 obtained a decision that $H_0$ is rejected, which means there is the effect of ARCH / GARCH on the residual model of an AR (1). Next will be modeled in the form of ARCH / GARCH. Based lag cuts on the plot of ACF and PACF plot is a lag to the 1st, 2nd, and 3rd. So that models are temporarily formed ARCH (1), ARCH (2), ARCH (3), GARCH (1.1), GARCH (1.2), GARCH (2.1), GARCH (2.2), GARCH (3.1), GARCH (3.2).

After obtained some of temporary model then do parameter estimation and parameter significance test. Based on the significance test parameters is known that for ARCH models (1), ARCH (2), ARCH (3), GARCH (1.1), GARCH (1.2) $H_0$ is rejected for all the parameters so that the parameters are significant for all five model. In the GARCH (2.1) for the $\beta_1$ parameter, $H_0$ is accepted and $H_0$ is rejected for other parameters, which means at significance level of 5% parameters are significant of the models except the $\beta_1$ parameter. For GARCH (2.2) $H_0$ is accepted on the $\beta_1$ and $\alpha_2$ parameters while for other parameters $H_0$ is rejected, meaning that the GARCH (2.2) parameters are significant of the models except the $\beta_1$ and $\alpha_2$ parameters. Furthermore GARCH (3.1) $H_0$ accepted for the $\beta_1$ parameters and $H_0$ is rejected for other parameters, so the parameters are significant of the models except the $\beta_1$ parameter. Lastly for GARCH (3.2) $H_0$ is rejected for the $\alpha_1$ parameter and $H_0$ accepted for other parameters which means for the GARCH model (3.2) only the $\alpha_1$ parameter is significant to the model.

Because the model is still more than one, then the best model selection based on significant parameters and models with the smallest MSE value. Based on the calculation of the MSE obtained the smallest MSE value is GARCH (1.2). But GARCH (1.2) has one parameter which is eliminated from the model because the coefficient value is treated as zero. So in this study selected the best ARCH / GARCH model that is GARCH (1.1).This is because in the MATLAB function there is a statement that when there is a model with one of its parameters is zero indicates if a simpler model than the model is sufficient. GARCH (1.1) has the following equation.

$$
\sigma_t^2 = 0.0000194884 + 0.09322292 \sigma_{t-1}^2 + 0.857834 \sigma_{t-1}^2
$$

(27)

Next do the modeling with Elman Recurrent Neural Network (ERNN). The first step in modeling ERNN is to build a network for mean model. Mean model using the original data which returns the closing price of stocks to be modeled and analyzed. Based on the plot of PACF data return closing stock price is known that the lag that exceeds the standard error limit is the 1st lag, the 12th, and 33rd so that the three lags become variable input components. The input data starts from the 34th data so that the data used for input is only 513. Next data is divided into 80% for the training data and 20% for the testing data. So 410 data as training data and 103 data as testing data. Activation function used is a bipolar sigmoid function for processing of inputs to the hidden layer. As for the processing from hidden to output using a linear function.
After the learning process which consists of the training and testing, we get the value of MSE for each network with the number of different hidden layer neurons. MSE value is used to choose the best network. MSE results of each network can be seen as follows:

| Number of layer neurons | MSE training       | MSE testing       |
|-------------------------|--------------------|-------------------|
| 3-1-1                   | $0.4401 \times 10^{-3}$ | $0.2723 \times 10^{-3}$ |
| 3-2-1                   | $0.4298 \times 10^{-3}$ | $0.2680 \times 10^{-3}$ |
| 3-3-1                   | $0.4309 \times 10^{-3}$ | $0.2595 \times 10^{-3}$ |
| 3-4-1                   | $0.4332 \times 10^{-3}$ | $0.2679 \times 10^{-3}$ |
| 3-5-1                   | $0.5000 \times 10^{-3}$ | $0.2000 \times 10^{-3}$ |
| 3-6-1                   | $0.4529 \times 10^{-3}$ | $0.2609 \times 10^{-3}$ |
| 3-7-1                   | $0.5250 \times 10^{-3}$ | $0.2486 \times 10^{-3}$ |
| 3-8-1                   | $0.4709 \times 10^{-3}$ | $0.3876 \times 10^{-3}$ |
| 3-9-1                   | $0.4556 \times 10^{-3}$ | $0.2964 \times 10^{-3}$ |
| 3-10-1                  | $0.4531 \times 10^{-3}$  | $0.3279 \times 10^{-3}$  |

Based on the calculation of MSE value obtained the smallest MSE testing value of $0.2000 \times 10^{-3}$. So the best network for the mean model is a network with three neurons of input layer, five neurons of hidden layer, and 1 neuron of output layer. Plot data from training and testing on the best network can be seen in Figure 5.

![Figure 5. Plot Output and Target of Mean Model](image)

In Figure 5, the predicted (output) result indicated by the green star symbol is about zero. Although the output value does not coincide with the target, but the results are sufficiently accurate predictions. Because the output values close to the target (blue line).

Next build a network for variant models. The model variant uses the residual testing of the best network on mean model. Based on the output of the process of training and testing data obtained residual of training and testing. Furthermore, the data residual testing were tested to determine whether it contains residual effects of ARCH / GARCH or not. The test uses Lagrange Multiplier test. The Lagrange Multiplier test is performed by `archtest(ts)` where `ts` is the residual of the data testing.
The test shows output $h = 1$ which means reject $H_0$ so it can be concluded that there is ARCH / GARCH effect in residual.

In the model of variants input component identified based on the PACF lag of the residual squared data it is known that the lag that exceeds the standard error limit is the $1^{st}$ lag and the $3^{rd}$ lag so that the lags become the input component variable. The input data starts from the $4^{th}$ data so that the data used for input is only 100. Next data is divided into 80% for the training data and 20% for the testing data. So 80 data as training data and 20 data as testing data. Activation function used is a bipolar sigmoid function for processing of inputs to the hidden layer. As for the processing from hidden to output using a linear function.

After the learning process which consists of the training and testing, we get the value of MSE for each network with the number of different hidden layer neurons. MSE value is used to choose the best network. MSE value of each network can be seen in Table 5.

Based on Table 5 obtained the smallest MSE testing value of $0.0920 \times 10^{-3}$. So the best network on the variant model is a network with two neurons of input layer, two neurons of hidden layer, and one neuron of output layer.

| Number of layer neurons | MSE training | MSE testing |
|-------------------------|--------------|-------------|
| 2-1-1                   | $0.2946 \times 10^{-3}$ | $0.8306 \times 10^{-3}$ |
| 2-2-1                   | $0.4105 \times 10^{-3}$ | $0.0920 \times 10^{-3}$ |
| 2-3-1                   | $0.2980 \times 10^{-3}$ | $8.6900 \times 10^{-4}$ |
| 2-4-1                   | $0.2556 \times 10^{-3}$ | $0.0921 \times 10^{-3}$ |
| 2-5-1                   | $0.2231 \times 10^{-3}$ | $0.1199 \times 10^{-3}$ |
| 2-6-1                   | $0.3096 \times 10^{-3}$ | $0.1041 \times 10^{-3}$ |
| 2-7-1                   | $0.2472 \times 10^{-3}$ | $0.1657 \times 10^{-3}$ |
| 2-8-1                   | $0.3960 \times 10^{-3}$ | $0.0964 \times 10^{-3}$ |
| 2-9-1                   | $0.2664 \times 10^{-3}$ | $0.1265 \times 10^{-3}$ |
| 2-10-1                  | $0.3000 \times 10^{-3}$ | $0.1406 \times 10^{-3}$ |

The plot of training and testing data of the best network for variant model is as follows:
In Figure 6, the predicted (output) result indicated by the green star symbol is about zero. Although the output value does not coincide with the target, but the results are sufficiently accurate predictions. Because the output values close to the target (blue line).

To know the best model based on the smallest MSE value then between model ARCH / GARCH with ERNN model compared. Based on the models obtained earlier then the comparison is done as in Table 6.

| Model       | MSE         |
|-------------|-------------|
| GARCH (1,1) | $1.9662 \times 10^{-7}$ |
| ERNN (2,2,1)| $0.0920 \times 10^{-3}$ |

Based on Table 6 it is known that the smallest MSE value is $1.9662 \times 10^{-7}$. So we can conclude that the best model for forecasting the residual variance of return data of closing price of the BRI stock is GARCH (1.1).

6. Conclusion
This study investigate the volatility dynamics of return of closing price of the BRI stock. It examined two approaches for forecasting with conditional volatility –models based on statistics, that is ARCH/GARCH models and ERNN models.

The result of best ARCH / GARCH model is GARCH (1.1) by this equation $\sigma_t^2 = 0.0000194884 + 0.09322292 \sigma_{t-1}^2 + 0.857834 \sigma_{t-1}^2$ which means the residual variance of return data of closing price of the BRI's stock t-period is influenced by the quadratic residual period to t-1 for 0.0932229, the t-1 period variant of 0.857834 and the constant value of 0.0000194884.

In ERNN mean model obtained the best model is the network with three neurons of input layer, 5 neurons of hidden layer, and 1 neuron of output layer. Residual of the best network is known to contain the effects of ARCH / GARCH so that further build a network for variant model. In ERNN variant models obtained the best model is the network with two neurons of input layer, two neurons of hidden layer, and one neuron of output layer.

In this experiment the ARCH/GARCH model has smaller MSE value than ERNN model. So that the best model for forecasting the residual variance of return data of closing price of the BRI stock is ARCH/GARCH model which is GARCH (1.1) model, best of ARCH/GARCH model in this study.

References
[1] Bollerslev, T. 1986. Generalized Autoregressive Conditional Heteroscedasticity (GARCH). Journal of Econometrics 31, 307-327.
[2] Engle R., 1982. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. Journal of Econometrics 50, 987-1007
[3] Haykin, S. 2009. Neural Networks and Learning Machines. New Jersey: Prentice-Hall, Inc.
[4] Jalal, M. E. et al. 2016. Forecasting Incoming Call Volumes in Call Centers with Recurrent Neural Networks. Journal of Business Research 69 : page. 4811 – 4814.
[5] Kusumadewi, S. 2003. Artificial Intelligence (Teknik dan Aplikasinya). Yogyakarta : Graha Ilmu.
[6] Kusumadewi, S. 2004. Membangun Jaringan Syaraf Tiruan menggunakan MATLAB & Excel Link. Yogyakarta : Graha Ilmu.
[7] Sani, D. L. 2014. Penerapan Elman-Recurrent Neural Network Pada Peramalan Konsumsi Listrik Jangka Pendek Di PT. PLN App Malang. Jurnal Mahasiswa Statistik Vol. 2 No. 6 : Hal. 441-444.
[8] Rosadi, D. 2012. Ekonometrika & Analisis Runtun Waktu Terapan dengan Eviews. Yogyakarta: Andi Offset.
[9] Tsay, R. S. 2005. *Analysis of Financial Time Series* Second Edition. New York: A John Wiley & Sons, Inc.

[10] Wei, W. W. S. 2006. *Time Series Analysis: Univariate and Multivariate Methods*. Second Edition. Boston: Pearson Education Inc.

[11] Winarno, W. W. 2007. *Analisis Ekonometrika dan Statistika dengan Eviews*. Yogyakarta: Unit Penerbit dan Percetakan Sekolah Tinggi Ilmu Manajemen YKPN.