

PEMODELAN VOLATILITAS *RETURN* PORTOFOLIO SAHAM MENGUNAKAN FEED FORWARD NEURAL NETWORK (Studi Kasus :PT Bumi Serpong Damai Tbk. Dan PT H.M Sampoerna Tbk.)

Rizki Pradipto Widyantomo¹, Abdul Hoyyi², Tatik Widiharih³

^{1,2,3} Departemen Statistika, Fakultas Sains dan Matematika, Universitas Diponegoro
e-mail : hoyyistat@gmail.com

ABSTRACT

Time series analysis is an analysis used to predict a time-observed data, one of which is the ARIMA model. ARIMA model assumes a constant residual variance (homogeneous). While financial data usually produce ARIMA model with variance error that is not constant. If the assumption of homogeneity of the residual variance is not met, then the method that can be used is ARCH or GARCH model. Another method that can be used on the data assuming the homogeneity of the variance error is not met is the Neural Network model. In this model we use Neural Network model with variance and residual as the input variables that obtained from ARCH / GARCH model. The data used are BSDE and HMSP asset portfolio returns from November 14, 2016 to January 18, 2018. In this study the selected input variables are from ARIMA (1.0.1) GARCH (1,1) model. The best Neural Network model obtained is Neural Network model with 10 hidden layers with MSE value 6.58×10^{-10} with model train evaluation which is MAPE value 1.14441%.

Keywords: Time series Analysis, ARCH / GARCH, Neural Network, Return.

1. PENDAHULUAN

Setiap investor melakukan investasi guna mendapatkan *return* yang semaksimal mungkin dengan risiko tertentu dalam berinvestasi. Hal tersebut dapat dicapai dengan melakukan alokasi investasi yang benar. Alokasi investasi merupakan tindakan menetapkan bobot investasi atau proporsi instrumen keuangan tak berisiko (*risk free asset*) dan instrumen keuangan (*risky asset*) atau yang biasa disebut dengan membentuk portofolio. *Return* merupakan pendapatan yang dinyatakan dalam persentase dari modal awal investasi yang meliputi keuntungan jual beli saham baik *capital gain* maupun *capital loss*. Secara teknis, semakin besar *return* yang diharapkan maka risiko yang dihadapi oleh investor juga akan semakin besar, karena risiko dan *return* memiliki pola hubungan yang searah dan linier (Tandelilin, 2001). Oleh karena itu, pembentukan portofolio dari saham-saham tertentu bertujuan untuk memperoleh suatu investasi yang memberikan *return* yang sama namun memberikan risiko lebih rendah, atau dengan risiko yang sama namun memberikan *return* yang lebih tinggi. Besarnya risiko dapat dilihat dari nilai deviasi standar, semakin kecil deviasi standar yang akan terjadi, maka tingkat risikonya semakin kecil.

Berdasarkan model Markowitz (1952) diasumsikan bahwa *return* berdistribusi normal dan bahwa tingkat risiko dapat diukur sebagai ukuran dari standar deviasi return aset. Namun, karena pada umumnya data *time series* finansial bersifat heteroskedastisitas, tantangannya adalah untuk memperkirakan variansi bersyaratnya. Menurut Engle (1982) variansi bersyarat *error* adalah fungsi dari *error* bersyarat periode sebelumnya dimana *error* bersyarat yang diprediksi perlu di auto regresi dengan data periode sebelumnya untuk perkiraan varian yang lebih baik. Maka dari itu, ramalan yang dilakukan pada kondisi autoregresif dan memenuhi asumsi heteroskedastisitas dapat dilakukan dengan membentuk model runtun waktu *Autoregressive Conditional Heteroscedasticity* yang diperkenalkan oleh Engle (1982) dan model pemodelan yang lebih fleksibel untuk pemodelan variansi yang tidak konstan yaitu model *Generalized Autoregressive Conditional Heteroscedasticity* yang diperkenalkan oleh Bollerslev (1986).

Model *Autoregressive Conditional Heteroscedasticity* dan *Generalized Autoregressive Conditional Heteroscedasticity* dapat digunakan untuk meramalkan deviasi standar atau volatilitas dari *return* saham. Model lain yang dapat digunakan untuk memodelkan data yang mempunyai fluktuasi tinggi dan tidak tetap adalah *Artificial Neural Network* (ANN). Model *Artificial Neural Network* menawarkan peningkatan potensial pada model sebelumnya karena ANN memiliki kemampuan untuk belajar, dan memanipulasi *error*, data hilang atau input data fuzzy. Menurut Warsito (2009), ANN dapat dibagi dua berdasarkan jumlah lapisannya, yaitu jaringan lapis tunggal dan jaringan multi lapis. Jaringan multi lapis merupakan jaringan yang sering digunakan karena jaringan multi lapis dapat menyelesaikan masalah yang lebih kompleks dibanding jaringan lapis tunggal yang hanya dapat menyelesaikan masalah sederhana. *Feedforward neural network* merupakan jaringan multi lapis yang sering digunakan yang dapat memodelkan hubungan antara variabel input dengan satu atau lebih variabel respon. Berdasarkan uraian diatas peneliti ingin melakukan penelitian menggunakan metode alternatif lain yaitu model *Feedforward Neural Network* dengan peubah input model ARCH/GARCH untuk memodelkan volatilitas *return* portofolio 2 aset saham PT. Bumi Serpong Damai Tbk dan PT. H.M. Sampoerna Tbk yang terindikasi model heteroskedastisitas.

2. TINJAUAN PUSTAKA

2.1. Investasi

Menurut Tandelilin (2001), investasi adalah komitmen atas sejumlah dana atau sumber daya lainnya yang dilakukan pada saat ini, dengan tujuan memperoleh keuntungan di masa depan. Terdapat dua macam investasi yaitu investasi pada aset finansial dan investasi pada aset riil. Aset finansial dapat dilakukan di pasar modal dalam bentuk saham, obligasi, waran, opsi, dan sebagainya. Aset riil dapat berbentuk pembelian aset produktif (seperti tanah dan rumah), pendirian pabrik, dan sebagainya.

2.2. Saham

Saham merupakan tanda penyertaan atau kepemilikan seseorang atau badan dalam suatu perusahaan (Darmadji dan Fakhruddin, 2001).

2.3. Portofolio

Risiko dalam teori portofolio didefinisikan sebagai deviasi standar tingkat keuntungan (σ). Hal ini disebabkan karena σ menunjukkan seberapa jauh kemungkinan nilai yang diperoleh menyimpang dari nilai yang diharapkan (*expected value*). Semakin besar nilai σ semakin besar kemungkinan nilai riil menyimpang dari yang diharapkan, yang berarti semakin tinggi risikonya (Husnan, 2001).

2.4. Return Portofolio

Perhitungan *return* masing-masing aset dapat dilakukan dengan cara berikut :

$$R_{it} = \ln \frac{P_{it}}{P_{i(t-1)}}$$

dengan :

R_{it} merupakan *return* aset ke-i pada waktu ke-t, P_{it} merupakan harga aset ke-i pada waktu ke t, $P_{i(t-1)}$ merupakan harga aset ke-i pada waktu ke t.

Menurut Maruddani dan Purbowati (2009), untuk memperoleh portofolio yang optimal, dilakukan pengoptimalan dengan cara meminimumkan varian *return* portofolio

menggunakan metode *Mean Variance Efficient Portofolio* (MVEP) agar bobot alokasi dana (\mathbf{w}) optimal. Secara matematis dapat ditulis sebagai berikut:

$$\sum_{i=1}^n w_i = [1 \ 1 \ \dots \ 1] \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \mathbf{1}_N^T \mathbf{w} = 1$$

Untuk memperoleh nilai yang dapat meminimumkan varian *return* potofolio dapat diselesaikan menggunakan metode *Lagrange* sebagai berikut:

$$L(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \Sigma \mathbf{w} - \lambda (\mathbf{w}^T \mathbf{1}_N - 1)$$

Fungsi *Lagrange* harus diturunkan terhadap untuk mendapatkan nilai atau dapat ditulis sebagai berikut:

$$L'(\mathbf{w}) = \frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = 0$$

$$L'(\mathbf{w}) = \frac{\partial}{\partial \mathbf{w}} \left[\frac{1}{2} \mathbf{w}^T \Sigma \mathbf{w} - \lambda (\mathbf{w}^T \mathbf{1}_N - 1) \right] = 0$$

$$\mathbf{w} = \lambda \Sigma^{-1} \mathbf{1}_N$$

Nilai pengali *Lagrange* dapat dicari dengan cara sebagai berikut :

$$\lambda = (\mathbf{1}_N^T \Sigma^{-1} \mathbf{1}_N)^{-1}$$

Nilai \mathbf{w} disubstitusikan dengan nilai pengali *Lagrange*:

$$\mathbf{w} = \frac{\Sigma^{-1} \mathbf{1}_N}{\mathbf{1}_N^T \Sigma^{-1} \mathbf{1}_N}$$

Dalam mencari bobot menggunakan MVEP diperlukan pemenuhan asumsi normal multivariat antar dua *return* saham yang dapat dicari menggunakan jarak mahalanobis antar *return* dan kuantil chi-kuadrat. Asumsi normal multivariat dapat terpenuhi jika plot antara jarak mahalanobis dan kuantil chi-kuadrat mengikuti garis lurus dengan *slope* 1.

Return portofolio dapat dihitung berdasarkan persamaan berikut (Tsay, 2002) :

$$R_{pt} = \sum_{i=1}^N w_i R_{it}$$

dengan adalah w_i bobot aset ke- i , adalah R_{it} *return* aset ke- i pada waktu ke- t , adalah R_{pt} *return* portofolio pada waktu ke- t .

2.5. Analisis Runtun Waktu

Menurut Chatfield (2000) *time series* atau runtun waktu adalah himpunan observasi data yang diukur secara berurutan dalam waktu. Metode *time series* adalah salah satu metode peramalan yang menggunakan analisa pola hubungan antara variabel yang akan diperkirakan dengan variabel waktu.

2.6. Fungsi Autokorelasi dan Autokorelasi Parsial

Menurut Wei (2006), fungsi autokorelasi merupakan fungsi yang menyatakan korelasi antara deret waktu Z_t dan Z_{t-k} . Fungsi autokorelasi (ACF) dapat diduga sebagai berikut :

$$\rho_k = \frac{\sum_{t=k+1}^n (Z_t - \mu)(Z_{t-k} - \mu)}{\sum_{t=1}^n (Z_t - \mu)^2}$$

Menurut Wei (2006) autokorelasi parsial merupakan korelasi antara Z_t dan Z_{t+k} dengan mengabaikan ketidakbebasan $Z_{t+1}, Z_{t+2}, \dots, Z_{t+k-1}$.

Secara umum nilai fungsi autokorelasi parsial (PACF) pada lag ke- k adalah:

$$\phi_{kk} = \frac{\begin{vmatrix} 1 & \rho_1 & \rho_2 & \dots & \rho_{k-2} & \rho_1 \\ \rho_1 & 1 & \rho_1 & \dots & \rho_{k-3} & \rho_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \dots & \rho_1 & \rho_k \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 & \rho_2 & \dots & \rho_{k-2} & \rho_{k-1} \\ \rho_1 & 1 & \rho_1 & \dots & \rho_{k-3} & \rho_{k-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \dots & \rho_1 & 1 \end{vmatrix}}$$

2.7. Model Time Series Box Jenkins

Model Box-Jenkins merupakan pemodelan data runtun waktu diperkenalkan oleh Box dan Jenkins pada tahun 1970 sebagai berikut :

a. Model *Autoregressive* (AR)

$$Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} + a_t$$

b. Model *Moving Average* (MA)

$$Z_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \dots - \theta_q a_{t-q}$$

c. Model *Autoregressive Moving Average* (ARMA)

$$Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \dots - \theta_q a_{t-q}$$

d. Model *Autoregressive Integrated Moving Average* (ARIMA)

$$\phi_p(B)(1-B)^d Z_t = \theta_q(B)a_t$$

Tahapan pemodelan ARIMA adalah yang pertama adalah mengidentifikasi model, mengestimasi parameter, dan melakukan uji asumsi model dengan pengujian independensi residual dan normalitas residual.

2.8. Uji Lagrange Multiplier

Uji *Lagrange Multiplier* (LM) merupakan suatu uji terhadap kehadiran unsur *heteroskedasticity* (Tsay,2002):

Hipotesis :

$H_0 : \alpha_1 = \alpha_2 = \dots = \alpha_m = 0$ (tidak ada efek ARCH/GARCH dalam residual sampai lag ke- m)

H_1 : minimal ada satu nilai $\alpha_i \neq 0$, $i = 1, 2, \dots, m$ (ada efek ARCH/GARCH dalam residual sampai lag ke- m)

Statistik Uji :

$$LM = NR^2 \sim \chi_m^2$$

dengan

N merupakan banyaknya pengamatan, R^2 merupakan nilai koefisien determinasi, m merupakan banyaknya lag yang diuji.

Kriteria uji :

Tolak H_0 jika nilai $LM > \chi_{(\alpha,m)}^2$ atau $p\text{-value} < \alpha$

2.9. Model Heteroskedastisitas ARCH/GARCH

a. Model ARCH

$$\sigma_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \dots + \alpha_m a_{t-m}^2$$

dengan α_0 merupakan konstanta, α_i merupakan parameter dari ARCH, a_t merupakan residual ARIMA pada waktu t .

b. Model GARCH

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^p \alpha_i a_{t-i}^2 + \sum_{j=1}^s \beta_j \sigma_{t-j}^2$$

Dengan β_j merupakan parameter dari GARCH, σ_{t-j}^2 : variansi dari residual pada saat t-j, a_t merupakan residual ARIMA pada waktu t.

Koefisien-koefisien dari model GARCH(s,m) bersifat :

1. $\alpha_0 > 0$
2. $\alpha_i \geq 0$ untuk $i = 1, 2, \dots, s$
3. $\beta_j \geq 0$ untuk $j = 1, 2, \dots, m$
4. $\sum_{i=1}^m \sum_{j=1}^s \alpha_i + \beta_j < 1$

Tahapan pemodelan ARCH/GARCH yang dilakukan sama seperti pada pemodelan ARIMA.

2.10 Jaringan Syaraf Tiruan

Menurut Kusumadewi (2004), jaringan syaraf adalah merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran.

Neuron-neuron dalam *neural network* disusun dalam grup, yang disebut dengan *layer* (lapis). Susunan *neuron-neuron* dalam lapis dan pola koneksi di dalam dan antarlapis disebut dengan arsitektur jaringan. Arsitektur ini merupakan salah satu karakteristik penting yang membedakan *neural network*. Secara umum ada tiga lapis yang membentuk *neural network*:

1. Lapis input
Unit-unit di lapisan input disebut unit-unit input. Unit-unit input tersebut menerima pola inputan dari luar yang menggambarkan suatu permasalahan.
2. Lapis tersembunyi (*hidden layer*)
Unit-unit dalam lapisan tersembunyi disebut unit-unit tersembunyi, dimana outputnya tidak dapat diamati secara langsung..
3. Lapis output
Unit-unit dalam lapisan output disebut unit-unit output. Output dari lapisan ini merupakan solusi *Neural Network* terhadap suatu permasalahan.

Menurut Warsito (2009), *neural network* dapat dibagi menjadi dua berdasarkan jumlah lapisan dalam jaringan tersebut yaitu *jaringan lapisan tunggal (single layer)* dan *jaringan multi lapis (multi layer)*.

1. Jaringan Lapisan Tunggal (*Single Layer*)
Sebuah jaringan disebut lapis tunggal jika jaringan tersebut tidak mempunyai lapisan tersembunyi atau hanya mempunyai satu lapisan bobot terkoneksi (Fausset, 1994).
2. Jaringan Multilapis (*Multilayer*)
Jaringan multilapis merupakan perluasan dari jaringan *single layer* (Siang, 2005). Jaringan multilapis terdiri dari lapisan input (*input layer*), lapisan tersembunyi (*hidden layer*) dan lapisan output (*output layer*). Lapisan tersembunyi terletak diantara lapisan input dan lapisan output.

Pada penelitian ini akan digunakan algoritma *Conjugate Gradient* formula Polak dan Ribiere sebagai algoritma pelatihan. Nilai β_t pada formula ini dapat ditentukan dengan rumus :

$$\beta_t = \frac{\mathbf{g}_{t+1}^T [\mathbf{g}_{t+1} - \mathbf{g}_t]}{\mathbf{g}_t^T \mathbf{g}_t}$$

Menurut Warsito (2009), algoritma *Conjugate Gradient* dapat dideskripsikan sebagai berikut :

Langkah 1 : a. Inisialisasi bobot awal dengan bilangan acak kecil

b. Inisialisasi epoch = 0, MSE=0

c. Tetapkan maksimum epoch dan target error.

Langkah 2 : Hitung $g_j = \frac{\partial E}{\partial w_j}, g_{b2} = \frac{\partial E}{\partial b2}, g_{ij} = \frac{\partial E}{\partial v_{ij}}, g_{b1j} = \frac{\partial E}{\partial b1j}$

\mathbf{g}_0 merupakan vektor gradient fungsi kinerja pada waktu t yang berisi $g_{j(0)}, g_{b2(0)}, g_{ij(0)}$ dan $g_{b1j(0)}$.

Langkah 3: Hitung $\mathbf{d}_0 = -\mathbf{g}_0$

Langkah 4 : Hitung $\alpha_t = -\frac{\mathbf{g}_t^T \mathbf{d}_t}{\mathbf{d}_t^T \mathbf{Q} \mathbf{d}_t}$ didefinisikan \mathbf{Q} adalah matriks definit positif berukuran

$p \times p$ dengan $\mathbf{Q}^T = \mathbf{Q}$.

Langkah 5 : Ubah vektor bobot baru dengan rumus : $\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha_t \mathbf{d}_t$ dengan $\mathbf{d}_t = -\mathbf{g}_t$

Langkah 6 : Jika MSE < Target error atau epoch = maksimum epoch, maka kondisi penghentian terpenuhi. Jika tidak lakukan langkah 7.

Langkah 7 : Jika $t+1 >$ maksimum epoch, maka

a. $\mathbf{w}_0 = \mathbf{w}_{t+1}$

b. $t = 1$

c. dan lakukan langkah 2.

Jika $t+1 \leq$ maksimum epoch, maka

a. $t = t+1$

b. Hitung \mathbf{g}_{t+1} yang merupakan vektor yang berisi $g_{j(t+1)}, g_{b2(t+1)}, g_{ij(t+1)}$ dan $g_{b1j(t+1)}$.

c. Maksimum β_t sesuai dengan formula yang akan digunakan.

d. Hitunglah arah pencarian yang baru dengan persamaan : $\mathbf{d}_{t+1} = -\mathbf{g}_{t+1} + \beta_t \mathbf{d}_t$ lanjutkan ke langkah 4.

2.11 Model Feed Forward Neural Network Dengan Input GARCH

Model *Feed Forward Neural Network* dengan input GARCH merupakan model kombinasi dari hasil indentifikasi model GARCH yang digunakan sebagai input dalam pemodelan jaringan syaraf tiruan. Model varian *Feedforward Neural Network* sebagai berikut:

$$\hat{\sigma}_t^2 = f^0 \left(\sum_{j=1}^q w_j^0 f_j^h \left(\sum_{i=1}^p v_{j,i}^h \sigma_{t-1}^2 + v_{j,i}^h a_{t-1}^2 + b_j^h \right) + b^0 \right)$$

2.12 Pemilihan Model Terbaik

Untuk memilih model dengan arsitektur terbaik pada *neural network*, digunakan *Mean Square Error (MSE)* dengan persamaan sebagai berikut :

$$MSE = \frac{\sum_{t=1}^n (Z_t - \hat{Z}_t)^2}{n}$$

dengan Z_t merupakan nilai data periode ke- t , \hat{Z}_t merupakan nilai *forecasting* periode ke- t , n merupakan ukuran sampel.

Ukuran ketepatan lain yang dapat digunakan untuk mengevaluasi kinerja model data runtun waktu, salah satunya adalah *MAPE* (*Mean Absolute Percentage Error*), dengan persamaan sebagai berikut :

$$MAPE = \frac{\sum_{t=1}^n \frac{|e_t|}{Z_t} \times 100\%}{n} = \frac{\sum_{t=1}^n \frac{|Z_t - \hat{Z}_t|}{Z_t} \times 100\%}{n}$$

3. METODE PENELITIAN

3.1. Sumber Data dan Variabel Penelitian

Data yang digunakan dalam penelitian Tugas Akhir ini adalah data sekunder harga penutupan (*closing price*) saham harian. Data diperoleh dari situs resmi finance.yahoo.com sebagai salah satu situs penyedia data historis pergerakan harga saham seluruh emiten termasuk saham anggota Bursa Efek Indonesia.

3.2. Variabel Penelitian

Variabel yang digunakan adalah harga saham harian dari beberapa saham anggota LQ45 periode Agustus 2017 hingga Januari 2018. Saham perusahaan yang akan diteliti adalah saham PT Bumi Serpong Damai Tbk, dan PT H. M. Sampoerna Tbk. Harga saham harian yang digunakan adalah harga penutupan pada periode 14 November 2016 hingga 18 Januari 2018.

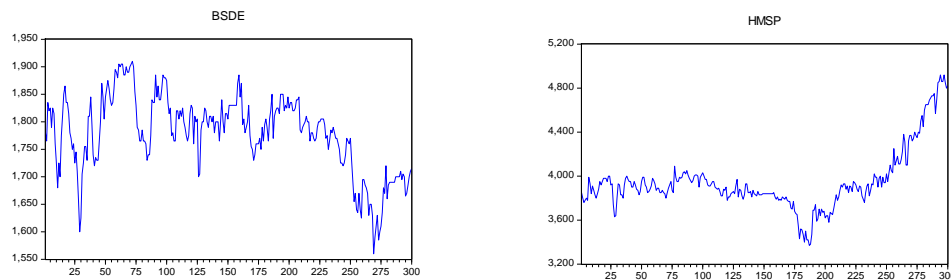
3.3. Tahapan Analisis Data

1. Mencari nilai *return* aset tunggal, *return* portofolio 2 aset dari saham Bumi Serpong Damai Tbk, dan H. M. Sampoerna Tbk.
2. Melakukan uji normalitas multivariat data *return* portofolio.
3. Menghitung *return* portofolio dengan MVEP.
4. Melakukan uji stasioneritas data dalam mean dan varian.
5. Mengidentifikasi model ARIMA.
6. Melakukan estimasi parameter model ARIMA.
7. Melakukan verifikasi model ARIMA.
8. Melakukan uji *Lagrange Multiplier* untuk mengetahui apakah ada efek ARCH/GARCH dalam model.
9. Mengidentifikasi model ARCH/GARCH.
10. Melakukan estimasi parameter model ARCH/GARCH.
11. Melakukan uji signifikansi parameter model ARCH/GARCH.
12. Menentukan input dan target berdasarkan model GARCH terbaik.
13. Menentukan data *training* dan data *testing*
14. Menentukan fungsi aktivasi pada *hidden layer* dan *output layer*, target error, banyak epoch yang akan diujikan, serta *learning rate*.
15. Melakukan proses *training* dan *testing*.
16. Memilih model NN terbaik berdasarkan MSE dan MAPE data *testing*.
17. Membuat plot *training* dan *testing* berdasarkan model yang telah didapat. Hasil dilihat dari plot data *output* dan target (data aktual), jika sudah saling berimpit atau bahkan sama maka model yang dihasilkan bisa dikatakan cukup baik.

4. HASIL DAN PEMBAHASAN

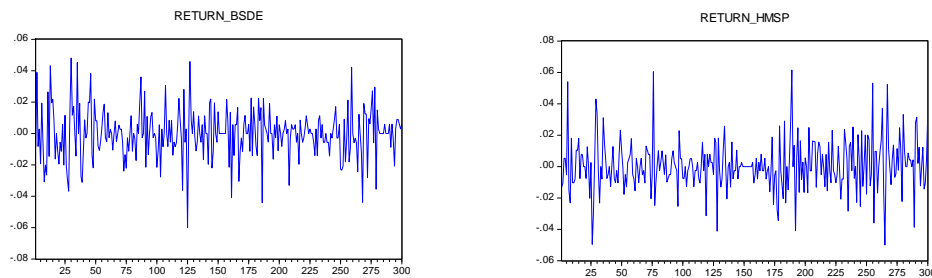
4.1 Deskripsi data

Data yang digunakan dalam penelitian ini adalah data *closing price* 2 aset saham yaitu PT. Bumi Serpong Damai Tbk (BSDE) dan PT. H.M. Sampoerna Tbk (HMSP) periode 14 November 2016 hingga 18 Januari 2018 selama hari aktif. Analisis akan dilakukan terhadap data *return* kedua aset saham yang telah dibentuk portofolio nya terlebih dahulu. Grafik data kedua aset saham dapat dilihat pada Gambar 1 yang menunjukkan plot data runtun waktu masing-masing aset saham tidak stasioner karena adanya fluktuasi naik turun dalam periode 1 tahun terakhir.



Gambar 1. Plot Runtun Waktu Data Saham

Data harga masing-masing aset saham selanjutnya diubah ke dalam bentuk *return* dan menghasilkan plot sebagai berikut :



Gambar 2. Plot Return Harga Saham

Berdasarkan data penelitian. dapat dituliskan tabel 1 sebagai berikut :

Tabel 1. Nilai *Expected Return* dan *Varian Return* Saham

Saham	<i>Expected Return</i>	<i>Varian Return</i>
BSDE	-0.000104	0.00024
HMSP	0.000751	0.00025

Dilakukan pembentukan *return* portofolio dengan metode *Mean Variance Efficient Portfolio* untuk mendapatkan *expected return* yang positif dengan tingkat risiko yang lebih rendah. Didapatkan bobot aset BSDE sebesar 46.3% dan bobot aset HMSP 53.6%.

4.2 Identifikasi Model ARIMA

Identifikasi model ARIMA dapat dilakukan dengan melihat plot ACF (fungsi autokorelasi) dan plot PACF (fungsi autokorelasi parsial). Pada plot ACF dan PACF terlihat bahwa lag ke-3, melebihi batas standar error. Sehingga model yang memungkinkan adalah AR ([3]), MA([3]) dan ARMA ([3,3]). Akan tetapi untuk mendapatkan model terbaik dapat dilakukan underfitting dan overfitting pada lag lainnya. Pada penelitian ini dilakukan overfitting pada lag 1 dan 2 sehingga model yang memungkinkan adalah ARIMA ([3],0,0), MA(0,0,[3]), ARMA([3],0,[3]), ARIMA(1,0,0), ARIMA(0,0,1),

ARIMA(2,0,0), ARIMA(0,0,2), ARIMA(3,0,0), ARIMA(0,0,3), ARIMA(1,0,1), ARIMA(1,0,2), ARIMA(1,0,3), ARIMA(2,0,1), ARIMA(2,0,2), ARIMA(2,0,3), ARIMA(3,0,1), ARIMA(3,0,2) dan ARIMA(3,0,3).

Setelah penentuan model yang mungkin, maka dilakukan uji asumsi model dan didapatkan model-model yang terindikasi heteroskedastisitas dalam residual model yaitu model ARIMA (0,0,3) dan ARIMA(1,0,1).

4.3 Uji Lagrange Multiplier

Uji *Lagrange Multiplier* untuk melihat apakah model ARIMA mempunyai efek ARCH/GARCH atau tidak. Hasil uji LM dapat dilihat pada Tabel 2.

Tabel 2. Uji *Lagrange Multiplier* Model ARIMA

Model	Probabilitas	Keputusan
ARIMA (0,0,3)	0.0012	H ₀ ditolak
ARIMA (1,0,1)	0.0012	H ₀ ditolak

Berdasarkan Tabel 2, kedua model ARIMA dari data return portofolio BSDE dan HMSP menolak H₀ karena mempunyai probabilitas < 0.05 sehingga dapat disimpulkan kedua model memiliki efek ARCH/GARCH.

4.4 Pemodelan ARCH/GARCH

Model awal yang terbentuk untuk *return* portofolio BSDE dan HMSP adalah ARIMA (0,0,3) ARCH(1), ARIMA (0,0,3) ARCH(2), ARIMA (0,0,3) GARCH(1,1), ARIMA (0,0,3) GARCH(1,2) dan ARIMA (0,0,3) GARCH(2,1) serta ARIMA (1,0,1) ARCH(1), ARIMA (1,0,1) ARCH(2), ARIMA (1,0,1) GARCH(1,1), ARIMA (1,0,1) GARCH(1,2) dan ARIMA (1,0,1) GARCH(2,1). Lalu dilakukan uji signifikansi parameter dan uji asumsi model ARCH-GARCH, maka didapatkan model terbaik sebagai berikut :

ARIMA (1,0,1) GARCH (1,1)

$$\sigma_t^2 = 2.16 \times 10^{-5} + 0.145850 a_{t-1}^2 + 0.693439 \sigma_{t-1}^2$$

Langkah selanjutnya membuat data series residual ARIMA dan data series varian GARCH dari model ARIMA (1,0,1) GARCH (1,1) yang akan digunakan sebagai input pada *neural network*.

4.5 Pemodelan Feed Forward Neural Network

Pemodelan *Feed Forward Neural Network* yang digunakan dalam analisis volatilitas *return* portofolio BSDE dan HMSP menggunakan 2 variabel input (σ_{t-1}^2 dan a_{t-1}^2) dan 1 variabel target (σ_t^2) yang didapatkan dari model ARIMA (1,0,1) GARCH(1,1).

Jumlah *Hidden Layer* yang digunakan sebanyak satu lapis, dengan jumlah unit yang dicobakan sebanyak 1 sampai 10 unit *hidden layer*. Sedangkan untuk parameter yang digunakan dalam *training* dan *testing* adalah sebagai berikut :

Tabel 3. Parameter Uji *Neural Network*

Parameter	Nilai Uji
Maksimum <i>epoch</i>	10 ³
<i>Performance goal</i>	10 ⁻⁵
<i>Learning rate</i> (α)	10 ⁻²

Algoritma Pembelajaran *Conjugate Gradient* yang digunakan adalah *traincgp*. Pembagian data *training* dan *testing* pada penelitian ini digunakan sebanyak 90% data *training* dan 10% data *testing*. Fungsi aktivasi yang digunakan pada jaringan ini adalah fungsi sigmoid biner (*logsig*) dan linear (*purelin*).

4.6 Pemilihan Model *Feed Forward Neural Network* Terbaik

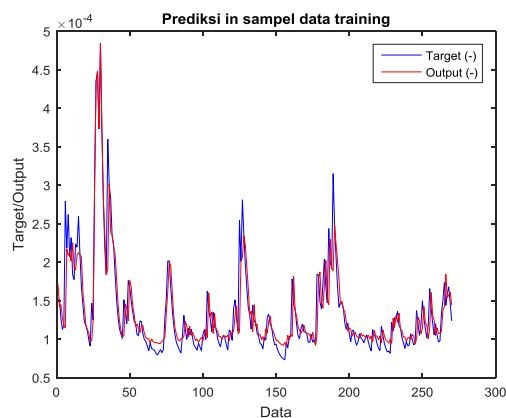
Setelah proses *training* dan *testing* telah dilakukan dengan unit *hiddenlayer* sebanyak 1 hingga 10, maka diperoleh nilai *MSE training* dan *MAPE training* untuk masing-masing model dengan jumlah *hidden layer* yang berbeda beda. Untuk mendapatkan model dengan arsitektur jaringan terbaik maka perlu dipilih model dengan nilai *MSE training* dan *MAPE training* terkecil.

Tabel 4. Nilai *MSE* dan *MAPE Training*

Jumlah Unit <i>Hidden Layer</i>	<i>MSE Training</i>	<i>MAPE Training</i>
1 Unit <i>Hidden Layer</i>	1.11×10^{-9}	4.9845
2 Unit <i>Hidden Layer</i>	1.05×10^{-9}	4.8690
3 Unit <i>Hidden Layer</i>	9.35×10^{-10}	4.3561
4 Unit <i>Hidden Layer</i>	9.11×10^{-10}	4.0189
5 Unit <i>Hidden Layer</i>	8.47×10^{-10}	3.8366
6 Unit <i>Hidden Layer</i>	7.94×10^{-10}	3.5489
7 Unit <i>Hidden Layer</i>	8.17×10^{-10}	3.6805
8 Unit <i>Hidden Layer</i>	8.04×10^{-10}	3.6820
9 Unit <i>Hidden Layer</i>	7.90×10^{-10}	3.5705
10 Unit <i>Hidden Layer</i>	6.58×10^{-10}	2.9789

Berdasarkan Tabel 4 maka model dengan arsitektur terbaik yang terpilih adalah model dengan jumlah *hidden layer* 10 dengan *MSE training* sebesar $6,58764 \times 10^{-10}$ dan *MAPE training* sebesar 2,9789. Untuk mendapatkan bobot yang optimal, maka dapat dilakukan *running* ulang dengan model dengan arsitektur yang menggunakan 10 *hidden layer*.

Setelah mendapatkan model arsitektur terbaik, lalu dilakukan perulangan untuk mendapatkan model dengan bobot optimal. Dilakukan perulangan sebanyak 10 kali dari model dengan arsitektur terbaik, didapatkan bobot optimal pada perulangan ke 10 dengan *MSE training* 6.38×10^{-10} . Plot data hasil *training* dari model dengan arsitektur terbaik dan bobot optimal ditunjukkan pada gambar sebagai berikut :

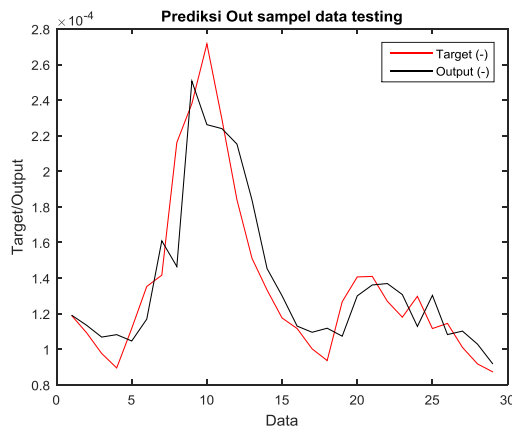


Gambar 3. Output dan Target Varian Data *Training*

Berdasarkan Gambar 3 diketahui bahwa plot antara *output* dan target pada data *training* saling berhimpitan, maka model yang dibentuk sangat baik. Untuk melihat apakah model cukup baik untuk digunakan dalam peramalan, maka dilakukan evaluasi kinerja model berdasarkan nilai *MAPE* data *testing*.

4.7 Evaluasi Kinerja Model dan Model Terbaik

Setelah diperoleh model *FFNN* dengan bobot optimal maka dilakukan evaluasi kinerja model berdasarkan nilai *Mean Absolute Percentage Error (MAPE)* untuk mengetahui apakah model yang terbentuk dapat digunakan untuk peramalan.



Gambar 4. *Output dan Target Varian Data Testing*

Berdasarkan Gambar 4. dapat terlihat bahwa plot antara *target* dan *output* data *testing* saling berdekatan walaupun tidak berhimpitan. Hasil tersebut cukup akurat dibuktikan dengan nilai *MAPE* data *testing* model sebesar 1.14441. Karena nilai *MAPE* *testing* < 10%, maka model memiliki kemampuan peramalan sangat baik.

$$\sigma_t^2 = \left(\left(\frac{0.5545}{1 + \exp(-(-0.9592\sigma_{t-1}^2 - 2.7168a_{t-1}^2 + 17.7615))} \right) + \left(\frac{0.8103}{1 + \exp(-(-2.6177\sigma_{t-1}^2 - 0.8022a_{t-1}^2 + 14.8032))} \right) + \dots \right) + \left(\frac{-0.9031}{1 + \exp(-(-1.7593\sigma_{t-1}^2 + 2.216a_{t-1}^2 + 2.5514))} \right) + 0.3358$$

Model *Feed Forward Neural Network* terbaik yang terpilih adalah model dengan arsitektur yang terdiri dari dua unit lapisan input σ_{t-1}^2 dan a_{t-1}^2 , sepuluh unit di lapisan tersembunyi, dan satu unit lapisan output.

5. KESIMPULAN

Berdasarkan hasil analisis dan pembahasan pada penelitian ini, kesimpulan yang didapatkan adalah sebagai berikut.

1. Pemodelan yang terpilih untuk data *return* portofolio BSDE dan HMSP adalah *neural network* dengan input ARIMA (1,0,1) GARCH (1,1). Model ARIMA-GARCH dipilih berdasarkan dengan model yang telah memenuhi asumsi *white noise* residual dan mengandung heteroskedastisitas. Untuk pemodelan *neural network* digunakan 2 buah input σ_{t-1}^2 dan a_{t-1}^2 serta 1 buah target σ_t^2 yang didapatkan dari model ARIMA GARCH.

Persamaan model ARIMA GARCH yang digunakan sebagai input dan target adalah sebagai berikut:

$$\sigma_t^2 = 2.16 \times 10^{-5} + 0.145850a_{t-1}^2 + 0.693439\sigma_{t-1}^2$$

Persamaan model *neural network* terbaik adalah sebagai berikut :

$$\sigma_t^2 = \left(\left(\frac{0.5545}{1 + \exp(-(-0.9592\sigma_{t-1}^2 - 2.7168a_{t-1}^2 + 17.7615))} \right) + \left(\frac{0.8103}{1 + \exp(-(-2.6177\sigma_{t-1}^2 - 0.8022a_{t-1}^2 + 14.8032))} \right) + \dots \right) + \left(\frac{-0.9031}{1 + \exp(-(-1.7593\sigma_{t-1}^2 + 2.216a_{t-1}^2 + 2.5514))} \right) + 0.3358$$

2. Pemilihan model *neural network* terbaik dilakukan dengan memilih model yang memiliki nilai *MSE training* terkecil untuk memilih arsitektur jaringan terbaik dan model dengan bobot optimal. Lalu dilihat nilai *MAPE testing* untuk mengevaluasi kinerja model *neural network* terbaik.

DAFTAR PUSTAKA

- Bollerslev, T. 1986. *Generalized Autoregressive Conditional Heteroscedasticity*. *Jurnal Ekonometrika*. Vol. 31, Hal. 307-327.
- Chatfield, C. 2000. *Time Series Forecasting*. Boca Raton, Florida: CRC Press
- Darmadji, T. dan Fakhruddin, H. 2001. *Pasar Modal di Indonesia Pendekatan Tanya Jawab*. Jakarta : Salemba Empat.
- Engle, R.F. 1982. *Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation*. *Econometrica*. Vol. 50, No. 4 : Hal. 987-1000.
- Fakhruddin, M. & Hadianto, M. 2001. *Perangkat dan Model Analisis Investasi di Pasar Modal*. Jakarta: Gramedia.
- Fausset, L. 1994. *Fundamental of Neural Network : Architecture Algorithms, and Application*. New Jersey: Prentice-Hall.
- Gitman, L. J. & Chad, J. Z. 2012. *Principle of Managerial Finance 13th edition*. Global Edition: Pearson Education Limited.
- Johnson, R. A, and Wichern, D. 2007. *Applied Multivariate Statistical Analysis*. Prentice Hall: Inc. United States of America
- Jorion, P. 2002. *Value at Risk: A New Benchmark for Managing Financial Risk*. Singapore: McGraw Hill.
- Kusumadewi, S. 2004. *Membangun Jaringan Syaraf Tiruan menggunakan Matlab & Excel Link*. Yogyakarta : Graha Ilmu.
- Markowitz, H. 1952. *Portfolio Selection*. *Journal of Finance*, Vol.7 No. 1 (Mar., 1952), pp. 77-91
- Maruddani, D.I.A dan Purbowati, A. 2009. *Pengukuran Value at Risk pada Aset Tunggal dan Portofolio dengan Simulasi Monte Carlo*. *Jurnal Media Statistika*. Vol. 2(2): 93-104. Undip: Semarang
- Sungkawa, I. 2011. *Penerapan Ukuran Ketepatan Nilai Ramalan Data Deret Waktu Dalam Seleksi Model Peramalan Volume Penjualan Pt Satriamandiri Citramulia*. *ComTech*. Vol. 2, No. 2 Hal : 636-645.
- Rosadi, D. 2010. *Analisis Ekonometrika dan Runtun Waktu Terapan dengan R*. Yogyakarta : Andi Offset.
- Soejoeti, Z. 1987. *Analisis Runtun Waktu*. Jakarta: Karunika.
- Tandelilin, E. 2001. *Analisis Investasi dan Manajemen Portofolio Edisi 1*. BPFE: Yogyakarta
- Warsito, B. 2009. *Kapita Selekta Statistika Neural Network* . Semarang: BPUNDIP Semarang.
- Wei, W.W.S. 2006. *Time Series Analysis, Univariate and Multivariate Methods*. Canada: Addison Wesley Publishing Company.