

**Perbandingan Metode *Autoregressive Integrated Moving Average* dengan Metode *Hybrid Autoregressive Integrated Moving Average* Jaringan Saraf Tiruan Pada Peramalan Harga Emas**

**SKRIPSI**



**RESKI ULANDARI**

**H 121 16 502**

**PROGRAM STUDI STATISTIKA DEPARTEMEN STATISTIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS HASANUDDIN  
MAKASSAR**

**2023**

**Perbandingan Metode *Autoregressive Integrated Moving Average* dengan Metode *Hybrid Autoregressive Integrated Moving Average* Jaringan Saraf Tiruan Pada Peramalan Harga Emas**



**Diajukan sebagai salah satu syarat memperoleh gelar Sarjana Sains pada Program Studi Statistika Departemen Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin**

**RESKI ULANDARI**

**H12116502**

**PROGRAM STUDI STATISTIKA DEPARTEMEN STATISTIKA FAKULTAS  
MATEMATIKA DAN ILMU PENGETAHUAN ALAM UNIVERSITAS  
HASANUDDIN**

**MAKASSAR**

**AGUSTUS 2023**

## LEMBAR PERNYATAAN KEOTENTIKAN

Saya yang bertanda tangan dibawah ini menyatakan dengan sungguh-sungguh bahwa skripsi yang saya buat dengan judul:

**Perbandingan Metode *Autoregressive Integrated Moving Average* dengan Metode *Hybrid Autoregressive Integrated Moving Average* Jaringan Saraf Tiruan Pada Peramalan Harga Emas**

adalah benar hasil karya saya sendiri, bukan hasil plagiat dan belum pernah dipublikasikan dalam bentuk apapun

Makassar, 18 Agustus 2023



**RESKI ULANDARI**  
**NIM H12116502**

**Perbandingan Metode *Autoregressive Integrated Moving Average* dengan Metode *Hybrid Autoregressive Integrated Moving Average* Jaringan Saraf Tiruan Pada Peramalan Harga Emas**

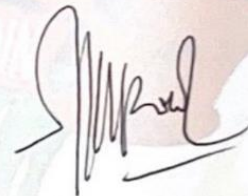
Disetujui oleh:

Pembimbing Utama

Pembimbing Pertama



**Sitti Sahriman, S.Si, M.Si**  
NIP.19881018 2015 04 2 002



**Dr. Nurtiti Sunusi, S.Si, M.Si**  
NIP.19720117 199703 2 002

Ketua Program Studi



**Dr. Anna Islamiyati, S.Si, M.Si**  
NIP.19770808 200501 2002

Pada 18 Agustus 2023

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Reski Ulandari  
NIM : H12116502  
Program Studi : Statistika  
Judul Skripsi : Perbandingan Metode *Autoregressive Integrated Moving Average* dengan Metode *Hybrid Autoregressive Moving Average* Jaringan Saraf Tiruan Pada Peramalan Harga Emas

Telah berhasil diperhatikan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

### DEWAN PENGUJI

1. Ketua : Sitti Sahrinan, S.Si, M.Si (.....)
2. Sekretaris : Dr.Nurtiti Sunusi, S.Si, M.Si (.....)
3. Anggota : Sri Astuti Thamrin, S.Si, M.Stat. Ph.D (.....)
4. Anggota : Dr.Georgina Maria Tinungki, M.Si (.....)

Ditetapkan di : Makassar

Tanggal : 18 Agustus 2023

## KATA PENGANTAR

*Assalamu'alaikum Warahmatullahi Wabaraktuh*

Puji syukur penulis panjatkan kepada Allah *Subhanahu Wa Ta'ala* atas segala limpahan rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi ini. Shalawat dan salam senantiasa tercurahkan kepada baginda Rasulullah *Shallallahu 'Alaihi Wa Sallam* beserta keluarga dan para sahabatnya. *Alhamdulillahirobbil'alamin*, berkat nikmat kemudahan dan pertolongan yang diberikan oleh Allah *Subhanahu Wa Ta'ala*, penulis dapat menyelesaikan skripsi yang berjudul "**Perbandingan Metode *Autoregressive Integrated Moving Average* dengan Metode *Hybrid Autoregressive Integrated Moving Average* Jaringan Saraf Tiruan Pada Peramalan Harga Emas**" yang disusun sebagai salah satu syarat akademik untuk memperoleh gelar sarjana pada Program Studi Statistika Departemen Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Penulis menyadari bahwa dalam penyelesaian skripsi ini tidak lepas dari bantuan dan dorongan dari berbagai pihak yang senantiasa turut membantu dalam bentuk moril maupun materil sehingga dengan segala keterbatasan kemampuan dan pengetahuan, penulis dapat menyelesaikan skripsi ini. Oleh karena itu, penulis menyampaikan ucapan terima kasih yang setulus-tulusnya serta penghargaan yang setinggi-tingginya kepada Ayahanda **A.Syafruddin** dan Ibunda **A.Suriati** yang telah memberikan dukungan penuh, pengorbanan luar biasa, limpahan cinta dan kasih sayang, kesabaran hati, serta dengan ikhlas telah menemani setiap langkah penulis dengan doa dan restu mulianya. Ucapan terima kasih juga penulis haturkan kepada Kakak tersayang **Ayu Andriani** yang senantiasa memberikan bantuan dan mendengarkan segala keluh kesah penulis dan memberi semangat kepada penulis dalam menyelesaikan skripsi ini, serta keluarga besar penulis, terima kasih atas doa mulia dan dukungannya selama ini.

Penghargaan yang tulus dan ucapan terima kasih dengan penuh keikhlasan dan ketulusan juga penulis ucapkan kepada:

1. **Bapak Prof. Dr. Ir. Jamaluddin Jompa, M.Sc.**, selaku Rektor Universitas Hasanuddin beserta seluruh jajarannya.
2. **Bapak Dr. Eng. Amiruddin**, selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin beserta seluruh jajarannya.
3. **Ibu Dr. Anna Islamiyati S.Si., M.Si.**, selaku Ketua Departemen Statistika beserta seluruh jajarannya..
4. **Ibu Sitti Sahrinan, S.Si., M.Si.** dan **Ibu Dr.Nurtiti Sunusi, S.Si, M.Si** selaku Pembimbing Utama dan Pembimbing Pertama yang dengan penuh kesabaran telah meluangkan waktu dan pemikirannya untuk senantiasa memberikan arahan, dorongan semangat, dan motivasi kepada penulis dari awal hingga selesainya penulisan tugas akhir ini.
5. **Ibu Sri Astuti Thamrin, S.Si., M.Stat. Ph.D** dan **Ibu Dr. Georgina Maria Tinungki, M.Si.**, selaku Tim Penguji yang telah meluangkan waktu dalam memberikan motivasi serta kritikan yang membangun kepada penulis dalam penyempurnaan tugas akhir ini.
6. Segenap Dosen Pengajar dan Staf yang telah memberikan ilmu dan kemudahan kepada penulis dalam berbagai hal selama menempuh pendidikan sarjana di Departemen Statistika.
7. Kepada Teman Angkatan **Statistika 2016**. Terima kasih atas bantuan, dorongan semangat dan nasehatnya kepada penulis.
8. Kepada seluruh pihak yang tidak dapat penulis sebutkan satu persatu terima kasih atas segala dukungan, partisipasi dan apresiasi yang diberikan kepada penulis

Penulis menyadari bahwa masih banyak kekurangan dalam penyusunan skripsi ini, untuk itu dengan segala kerendahan hati penulis memohon maaf. Akhir kata semoga tulisan ini dapat memberikan manfaat untuk berbagai pihak.

Makassar, 18 Agustus 2023

  
Reski Ulandari

**PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR  
UNTUK KEPENTINGAN AKADEMIK**

---

Sebagai civitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama : Reski Ulandari  
NIM : H12116502  
Program Studi : Statistika  
Departemen : Statistika  
Fakultas : Matematika dan Ilmu Pengetahuan Alam  
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Non-eksklusif (*Non-exclusive Royalty-Free Right*)** atas tugas akhir saya yang berjudul:

**“Perbandingan Metode *Autoregressive Integrated Moving Average* dengan Metode *Hybrid Autoregressive Integrated Moving Average* Jaringan Saraf Tiruan Pada Peramalan Harga Emas”**

Beserta perangkat yang ada (jika diperlukan). Terkait dengan hal diatas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (database), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar tanggal 18 Agustus 2023.

Yang menyatakan,



(Reski Ulandari)



## ABSTRAK

Peramalan data deret waktu adalah serangkaian pengamatan terhadap suatu variabel yang diambil dan dicatat secara berurutan dari waktu ke waktu. Salah satu model dalam peramalan adalah *Autoregressive Integrated Moving Average* (ARIMA). Namun, dalam model ARIMA memiliki kelemahan yaitu hanya mampu mengikuti pola deret waktu yang linier. Salah satu metode peramalan yang dapat digunakan untuk mengatasi pola deret waktu nonlinier yaitu menggunakan metode Jaringan Saraf Tiruan, namun terdapat kekurangan yaitu sering ditemui adanya permasalahan *overfitting*. Tetapi masalah tersebut dapat diminalisir dengan menggunakan metode *Hybrid* ARIMA Jaringan Saraf Tiruan. Tujuan dari penelitian ini adalah untuk membandingkan akurasi peramalan *Hybrid* ARIMA Jaringan Saraf Tiruan dengan model ARIMA dan memprediksi harga emas pada tahun 2020. Hasil yang diperoleh model ARIMA terbaik adalah ARIMA (1,1,0) dengan MAPE 14,7%. Dengan model ARIMA terbaik, diperoleh residual untuk *input* Jaringan Saraf Tiruan. Hasil yang diperoleh adalah arsitektur jaringan terbaik yaitu 4 *input* 20 *hidden layer* dan 1 *output* dengan nilai kesalahan terkecil. Dari hasil pengujian model *Hybrid* ARIMA Jaringan Saraf Tiruan terhadap data pengujian diperoleh nilai MAPE sebesar 10%.

**Kata Kunci:** Peramalan, ARIMA, *Hybrid* ARIMA Jaringan Saraf Tiruan, Harga Emas

**ABSTRACT**

*Forecasting time series data is a series of observations of a variable that is taken and recorded sequentially from time to time. One model in forecasting is the Autoregressive Integrated Moving Average (ARIMA). However, the ARIMA model has a weakness, namely it is only able to follow a linear time series pattern. One of the forecasting methods that can be used to overcome nonlinear time series patterns is using the Artificial Neural Network method, but there are drawbacks, namely overfitting problems are often found. But this problem can be minimized by using the Hybrid ARIMA Artificial Neural Network method. The purpose of this study is to compare the forecasting accuracy of Hybrid ARIMA Artificial Neural Networks with the ARIMA model and predict the price of gold in 2020. The results obtained by the best ARIMA model are ARIMA (1,1,0) with a MAPE of 14.7%. With the best ARIMA model, residuals are obtained for the input of the Neural Network. The results obtained are the best network architecture, namely 4 inputs, 20 hidden layers and 1 output with the smallest error value. From the test results of the ARIMA Hybrid Artificial Neural Network model for the test data, a MAPE value of 10% was obtained.*

**Keywords:** *Forecasting, ARIMA, Hybrid ARIMA Artificial Neural Networks, Gold Price*

**DAFTAR ISI**

<b>HALAMAN SAMPUL</b> .....	ii
<b>HALAMAN JUDUL</b> .....	ii
<b>LEMBAR PERNYATAAN KEOTENTIKAN</b> .....	iii
<b>HALAMAN PENGESAHAN</b> .....	v
<b>KATA PENGANTAR</b> .....	vi
<b>HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS</b> .....	viii
<b>ABSTRAK</b> .....	ix
<b>ABSTRACT</b> .....	x
<b>DAFTAR ISI</b> .....	xi
<b>DAFTAR GAMBAR</b> .....	xiii
<b>DAFTAR TABEL</b> .....	xiv
<b>DAFTAR LAMPIRAN</b> .....	xv
<b>BAB 1 PENDAHULUAN</b> .....	1
<b>1.1 Latar Belakang</b> .....	1
<b>1.2 Rumusan Masalah</b> .....	4
<b>1.3 Batasan Masalah</b> .....	4
<b>1.4 Tujuan Penelitian</b> .....	4
<b>1.5 Manfaat Penelitian</b> .....	4
<b>BAB 2 TINJAUAN PUSTAKA</b> .....	5
<b>2.1 Peramalan</b> .....	5
<b>2.2 Analisis Deret Waktu</b> .....	5
<b>2.3 Metode ARIMA</b> .....	6
<b>2.4 Metode Jaringan Saraf Tiruan (JST)</b> .....	9
<b>2.4.1 Arsitektur Jaringan Saraf Tiruan</b> .....	10

2.4.2	Fungsi Aktivasi .....	12
2.4.3	<i>Backpropogation</i> .....	14
2.5	Metode <i>Hybrid</i> ARIMA-JST .....	17
2.6	Ukuran Ketepatan Metode Peramalan .....	17
2.7	Emas .....	18
<b>BAB 3 METODOLOGI PENELITIAN .....</b>		<b>20</b>
3.1	Sumber Data .....	20
3.2	Identifikasi Variabel .....	20
3.3	Metode Analisis .....	20
3.3.1	ARIMA .....	20
3.3.2	Jaringan Saraf Tiruan .....	21
3.3.3	Metode <i>Hybrid</i> ARIMA Jaringan Saraf Tiruan .....	22
<b>BAB 4 HASIL DAN PEMBAHASAN .....</b>		<b>23</b>
4.1	Peramalan Menggunakan Model ARIMA .....	23
4.1.1	Identifikasi Kestasioneran .....	23
4.1.2	Identifikasi Model ARIMA .....	26
4.1.3	Pemeriksaan Diagnostik Model ARIMA .....	27
4.1.4	Pemilihan Model Terbaik .....	28
4.2	Peramalan Menggunakan <i>Hybrid</i> Jaringan Saraf Tiruan .....	29
4.2.1	Normalisasi Data .....	29
4.2.2	Perancangan Arsitektur Jaringan Saraf Tiruan .....	29
4.3	Perbandingan Hasil Ramalan .....	42
5.1	Kesimpulan .....	44
5.2	Saran .....	44
<b>DAFTAR PUSTAKA .....</b>		<b>45</b>
<b>LAMPIRAN .....</b>		<b>47</b>

**DAFTAR GAMBAR**

**Gambar 2. 1** Input dan Bobot..... 10

**Gambar 2. 2** Jaringan *Single layer* ..... 11

**Gambar 2. 3** Jaringan *Multi Layer* ..... 12

**Gambar 2. 4** Jaringan *Backpropogation* ..... 15

**Gambar 3. 1** Arsitektur Sistem peramalan *hybrid* ARIMA-JST .....20

**Gambar 4. 1** Plot Deret Waktu Data Harga Emas..... 23

**Gambar 4. 2** ACF dan PACF Data Harga Emas ..... 24

**Gambar 4. 3** Plot Data Harga Emas *Differencing* 1 ..... 25

**Gambar 4. 4** Plot ACF dan PACF setelah *Differencing* 1 ..... 26

**Gambar 4. 5** Perbandingan nilai prediksi dan aktual 4 *input*, 20 *hidden layer*.... 31

**Gambar 4. 6** Grafik Perbandingan Data Aktual dengan Hasil Peramalan..... 43

## DAFTAR TABEL

<b>Tabel 4. 1</b> Pengujian Stasioner Data Harga Emas .....	24
<b>Tabel 4. 2</b> Pengujian Stasioner Data Harga Emas <i>Differencing</i> 1 .....	25
<b>Tabel 4. 3</b> Uji Signifikansi Parameter Model ARIMA .....	27
<b>Tabel 4. 4</b> Uji Kolmogorov-smirnov.....	28
<b>Tabel 4. 5</b> Nilai MAPE dan RMSE Pada Model ARIMA.....	28
<b>Tabel 4. 6</b> Hasil Transformasi Harga Emas.....	29
<b>Tabel 4. 7</b> Dimensi Matriks Harga Emas .....	30
<b>Tabel 4. 8</b> Bias dan bobot awal pada <i>Input Layer</i> Terhadap <i>Hidden Layer</i> .....	31
<b>Tabel 4. 9</b> Bobot dan bias awal pada <i>Hidden Layer</i> Terhadap <i>Output Layer</i> .....	32
<b>Tabel 4. 10</b> <i>Sinyal Input</i> dari <i>Input Layer</i> ke <i>Hidden Layer</i> .....	33
<b>Tabel 4. 11</b> <i>Sinyal output</i> di <i>hidden layer</i> .....	34
<b>Tabel 4. 12</b> <i>Sinyal output</i> pada <i>hidden layer</i> .....	34
<b>Tabel 4. 13</b> <i>Sinyal</i> pada <i>output layer</i> .....	34
<b>Tabel 4. 14</b> Koreksi Bobot dan bias di <i>Hidden Layer</i> Terhadap <i>Output Layer</i> ....	35
<b>Tabel 4. 15</b> Faktor Kesalahan di Unit tersembunyi .....	36
<b>Tabel 4. 16</b> Aktivasi Kesalahan di Unit tersembunyi .....	36
<b>Tabel 4. 17</b> Koreksi Bias dan bobot <i>Input Layer</i> terhadap <i>Hidden Layer</i> .....	37
<b>Tabel 4. 18</b> Bias dan bobot Akhir pada <i>Input Layer</i> terhadap <i>Hidden Layer</i> .....	38
<b>Tabel 4. 19</b> Bobot dan bias akhir pada <i>Input Layer</i> terhadap <i>Output Layer</i> .....	39
<b>Tabel 4. 20</b> Bias dan bobot Akhir pada <i>Input Layer</i> terhadap.....	40
<b>Tabel 4. 21</b> Bobot dan bias Akhir pada <i>Input Layer</i> terhadap <i>Output Layer</i> .....	41
<b>Tabel 4. 23</b> Data Aktual dan Hasil Peramalan Harga Emas .....	42

**DAFTAR LAMPIRAN**

<b>Lampiran 1</b> Data Harga Emas .....	47
<b>Lampiran 2</b> Estimasi Parameter Model ARIMA .....	51
<b>Lampiran 3</b> Uji <i>Unit Root Test</i> ADF.....	53
<b>Lampiran 4</b> Data Peramalan Hasil Residual ARIMA (1,1,0) .....	54

## BAB 1

### PENDAHULUAN

#### 1.1 Latar Belakang

Emas merupakan logam mulia yang sering dijadikan sebagai alat tukar dalam perdagangan maupun sebagai standar keuangan berbagai negara (Joesoef, 2008). Emas merupakan salah satu instrumen simpanan pokok (investasi) yang relatif stabil dan efektif. Investasi emas telah lama menjadi salah satu alat investasi yang banyak diminati masyarakat Indonesia. Beberapa alasan sebagian besar masyarakat memilih investasi emas adalah karena ketahanannya terhadap inflasi, mudah dicairkan, menguntungkan karena harganya yang relatif stabil dan cenderung meningkat.

Emas memiliki harga yang bersifat fluktuatif atau dapat berubah-ubah setiap waktu. Naik turunnya harga emas mengikuti naik turunnya rupiah terhadap US\$. Oleh karena itu para investor emas harus dapat memperkirakan harga emas serta dapat memutuskan membeli atau menjual emas diwaktu yang tepat sehingga dapat memperoleh keuntungan dalam berinvestasi emas. Pergerakan harga emas yang akan datang dapat dipantau dengan menggunakan peramalan. Salah satu metode peramalan yang dapat digunakan dalam peramalan harga emas adalah analisis deret waktu. Analisis deret waktu dilakukan untuk memperoleh pola data deret waktu dengan menggunakan pengamatan sebelumnya untuk memprediksi suatu nilai pada masa yang akan datang (Maulana, 2018).

Peramalan data deret waktu adalah serangkaian pengamatan terhadap suatu variabel yang diambil dan dicatat secara berurutan dari waktu ke waktu (Wei, 2006). Teknik peramalan deret waktu terbagi menjadi dua bagian. Teknik yang pertama adalah model peramalan yang didasarkan pada model matematika statistik seperti *moving average*, *exponential smoothing*, regresi, dan *autoregressive integrated moving average* (ARIMA). Teknik yang kedua adalah model peramalan yang didasarkan pada kecerdasan buatan seperti *neural network*, algoritma genetika, *simulated annealing*, *genetic programming*, klasifikasi, dan *hybrid*.



Model peramalan yang sering digunakan dalam peramalan data deret waktu adalah model *autoregressive integrated moving average* (ARIMA). Metode ARIMA adalah metode peramalan yang dikembangkan oleh George Box dan Gwilyn Jenkins yang sering juga disebut metode deret waktu Box-Jenkins. Model ARIMA adalah model gabungan dari metode *Autoregressive* (AR) dan *Moving Average* (MA), yang diasumsikan berbentuk linier. Maksudnya, suatu struktur korelasi linier diasumsikan diantara nilai-nilai deret waktu. ARIMA sangat baik ketepatannya untuk peramalan jangka pendek dan untuk data deret waktu non stasioner pada saat linier (Munarsih 2011). Sedangkan untuk data peramalan dalam periode yang cukup panjang ketepatannya kurang baik karena biasanya akan cenderung flat (datar/konstan). Selain itu ARIMA mengalami penurunan keakuratan apabila terdapat komponen nonlinier deret waktu pada data pengamatan. Oleh karena itu, ARIMA tidak mampu memodelkan data deret waktu yang non linear (Zhang, 2003).

Kenyataannya dalam kehidupan tidak hanya data pola linier yang sering ditemui tetapi juga terdapat data pola nonlinier. Salah satu metode peramalan yang dapat digunakan untuk mengatasi pola nonlinier yaitu menggunakan jaringan saraf tiruan. Jaringan saraf tiruan (JST) merupakan sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan saraf biologi (Fausett, 1994: 3). Jaringan saraf tiruan terinspirasi oleh otak manusia dimana neuron saling interkoneksi secara non-linier. *Neuron* merupakan elemen pengolahan jaringan saraf tiruan. Setiap neuron menerima *input*, memproses *input* tersebut kemudian mengirimkan hasilnya berupa sebuah *output*. Jaringan saraf tiruan memiliki beberapa arsitektur jaringan yaitu *single layer*, *multilayer* dan *Competitive layer*.

Pada Jaringan saraf tiruan (JST) terdapat beberapa algoritma yang digunakan untuk peramalan deret waktu, salah satunya yaitu algoritma *backpropagation*. Algoritma *backpropagation* mempunyai cara kerja dengan menyesuaikan bobot yang saling terhubung antara *neuron* untuk mencapai kesalahan minimum antara keluaran hasil prediksi dengan keluaran yang nyata (Lee & Choi, 2013: 2943). Namun terdapat kekurangan pada model jaringan saraf tiruan yang sering ditemui yaitu permasalahan *overfitting*, dimana model yang dibuat hanya menghasilkan output yang baik untuk data yang dilatih saja dan tidak untuk

data yang divalidasi (data yang tidak termasuk proses pelatihan). Tetapi permasalahan tersebut dapat diminimalisir dengan mengkombinasikan jaringan saraf tiruan dengan model lain.

Salah satu model kombinasi (*hybrid*) yang pernah dicobakan untuk peramalan *time series* adalah model jaringan saraf tiruan dengan model ARIMA (Faruk 2010). Ada tiga hal yang menjadi alasan penggunaan pengkombinasian model ARIMA dan neural network (Zhang 2003). Pertama, sering kali terjadi kesulitan untuk menerapkan penggunaan model linier atau model nonlinier pada suatu permasalahan *time series* sehingga model kombinasi ini menjadi alternatif yang lebih mudah. Kedua, dalam kenyataannya *time series* jarang yang linier atau nonlinier dan sering mengandung keduanya sehingga pengkombinasian ini dapat digunakan untuk memodelkan deret waktu yang mengandung linier dan nonlinier. Ketiga, dalam beberapa literatur peramalan menyatakan bahwa tidak ada model tunggal yang terbaik pada setiap situasi.

Beberapa penelitian sebelumnya yang menggunakan metode ARIMA, jaringan saraf tiruan dan *hybrid* ARIMA jaringan saraf tiruan diantaranya adalah hasil penelitian Dwi Prisita Anggriningrum (2013) menunjukkan penggunaan metode ARIMA lebih akurat dibandingkan dengan metode jaringan saraf tiruan. Hasil penelitian Latifah Hanum (2017) menunjukkan peramalan dengan metode jaringan Saraf tiruan *Backpropagation* lebih akurat dibandingkan dengan metode ARIMA. Hasil penelitian Anugerah Fuad Ramadhan (2018) menunjukkan model *Hybrid* ARIMA jaringan saraf tiruan lebih akurat pada data pembukaan saham sedangkan pada data penutupan saham model ARIMA lebih akurat. Perubahan akan struktur data di masa mendatang serta terdapat berbagai macam pola dalam data membuat metode *hybrid* diharapkan menjadi strategi yang baik karena pada dasarnya tujuan utama dari melakukan kombinasi metode adalah untuk menggunakan kelebihan dari masing-masing metode sehingga menjadi cara yang lebih efektif dalam meningkatkan akurasi peramalan.

Pada penelitian ini dilakukan peramalan pada data harga emas dikarenakan harga emas nilainya relatife fluktuatif tetapi menunjukkan adanya tren peningkatan. Data harga emas merupakan data runtun waktu yang bersifat dependen

berhubungan satu sama lain secara statistik. Berdasarkan penelitian sebelumnya pengujian linearitas pada data harga emas menunjukkan adanya pola data linear dan nonlinear sekaligus sehingga dibutuhkan kombinasi antara ARIMA untuk data linear dan jaringan saraf tiruan untuk data nonlinearnya. Berdasarkan Latar belakang tersebut maka dilakukan penelitian untuk meramalkan harga emas menggunakan metode ARIMA dan *hybrid* ARIMA jaringan saraf tiruan dengan judul “*Perbandingan Metode Autoregressive Integrated Moving Average dengan Metode Hybrid Autoregressive Integrated Moving Average Jaringan Saraf Tiruan Pada Peramalan Harga Emas*”.

## **1.2 Rumusan Masalah**

Dari uraian latar belakang tersebut, maka dapat dirumuskan permasalahan yaitu, bagaimana perbandingan hasil peramalan pada data harga emas menggunakan metode *Autoregressive Integrated Moving Average* dan metode *Hybrid Autoregressive Integrated Moving Average* Jaringan Saraf Tiruan?

## **1.3 Batasan Masalah**

Pembahasan dalam tugas akhir ini dibatasi pada data harga emas dari bulan Januari 2016 sampai Desember 2020 yang saling berautokorelasi dan pada model jaringan saraf tiruan digunakan algoritma backpropagation.

## **1.4 Tujuan Penelitian**

Tujuan penelitian ini adalah untuk mengetahui perbandingan hasil peramalan pada data harga emas menggunakan metode *Autoregressive Integrated Moving Average* dan metode *Hybrid Autoregressive Integrated Moving Average* Jaringan Saraf Tiruan.

## **1.5 Manfaat Penelitian**

Penelitian ini dapat menambah wawasan mengenai peramalan menggunakan metode ARIMA dan metode *Hybrid* ARIMA jaringan saraf tiruan dan dapat meramalkan data harga emas menggunakan metode terbaik yang telah didapatkan sehingga bisa dijadikan sebagai bahan referensi untuk penelitian selanjutnya

## BAB 2 TINJAUAN PUSTAKA

### 2.1 Peramalan

Peramalan adalah memperkirakan besarnya atau jumlah sesuatu pada waktu yang akan datang berdasarkan data pada masa lampau yang dianalisis secara alamiah khususnya menggunakan metode statistika (Sudjana, 1989: 254). Peramalan biasanya dilakukan untuk mengurangi ketidakpastian terhadap sesuatu yang akan terjadi di masa yang akan datang. Menurut Makridakis (1999: 8), metode peramalan dibagi ke dalam dua kategori utama, yaitu metode kualitatif dan metode kuantitatif.

- a. Metode *kualitatif* merupakan metode yang menggabungkan beberapa faktor seperti pemikiran intuitif, perkiraan logis dan pengalaman pribadi.
- b. Metode *kuantitatif* merupakan metode yang membutuhkan informasi masa lalu yang dikuantitatifkan dalam bentuk data numerik. Metode ini mendasarkan ramalannya pada metode statistika dan matematika. Terdapat dua jenis model peramalan kuantitatif, yaitu model deret waktu (*time series*) dan model regresi (*regression*).

### 2.2 Analisis Deret Waktu

Analisis deret waktu merupakan salah satu prosedur statistika yang diterapkan untuk meramalkan struktur probabilistik keadaan yang akan terjadi di masa yang akan datang. Interval waktu antar *indeks* waktu  $t$  dapat dinyatakan dalam satuan waktu yang sama (Hendikawati, 2014: 8). Tujuan analisis deret waktu antara lain memahami dan menjelaskan mekanisme tertentu, meramalkan suatu nilai di masa depan, dan mengoptimalkan sistem kendali (Makridakis, Wheelwright, & McGree, 1999). Dasar pemikiran *time series* adalah pengamatan sekarang ( $Y_t$ ) tergantung pada satu atau beberapa pengamatan sebelumnya ( $X_{t-k}$ ). Dengan kata lain, model *time series* dibuat karena secara statistik ada korelasi antar deret pengamatan. Sehingga dapat digunakan untuk meramalkan pada beberapa periode ke depan serta *lead time* ( $l$ ) yang menyatakan periode peramalan di masa mendatang. Fungsi dari  $X_t(l)$  akan menyediakan peramalan pada titik awal  $t$  dengan objek mendapatkan nilai *mean square deviations* sekecil mungkin diantara nilai aktual dengan peramalan untuk setiap *lead time* ( $l$ ).

### 2.3 Metode ARIMA

Model *Autoregressive Integrated Moving Average* (ARIMA) merupakan hasil penggabungan antara model *autoregressive* AR( $p$ ), *moving average* MA( $q$ ) dengan proses *differencing*( $d$ ). Model *autoregressive* adalah suatu bentuk regresi, tetapi tidak menghubungkan variabel tak bebas melainkan menghubungkan nilai-nilai sebelumnya pada *time lag* (selang waktu) yang bermacam-macam. Jadi, suatu model *autoregressive* akan menyatakan suatu ramalan sebagai fungsi nilai-nilai sebelumnya dari deret waktu tertentu (Makridakis, Wheelwright, & Hyndman, 1998). Sedangkan model *moving average* merupakan model yang menggambarkan ketergantungan variabel terikat  $X$  terhadap nilai-nilai residual pada waktu sebelumnya yang berurutan.

Secara umum, bentuk model ARIMA( $p,d,q$ ) sebagai berikut:

$$\phi_p(B)(1 - B)^d X_t = \theta_q(B)e_t \quad (2.1)$$

dengan  $p$  = orde AR

$d$  = orde *differencing*

$q$  = orde MA

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

$$\theta_q(B) = (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q)$$

$\phi_1, \phi_2, \phi_3, \dots, \phi_p$  = koefisien orde dari komponen AR

$\theta_1, \theta_2, \theta_3, \dots, \theta_q$  = koefisien orde dari komponen MA

$(1 - B)^d$  = orde *differencing* non-musiman

$X_t$  = besarnya pengamatan (kejadian) pada waktu ke- $t$

$e_t$  = suatu proses *white noise* atau residual pada waktu ke- $t$  yang diasumsikan mempunyai mean 0 dan variansi konstan (Aswi, 2006).

Tahapan dalam model ARIMA menurut Box-Jenkins ada 4 (Gujarati, 2003), yaitu identifikasi model, penaksiran parameter, pemeriksaan diagnostik, dan peramalan.

## 1. Identifikasi model

Langkah pertama pada tahap identifikasi model adalah dengan memplotkan data secara grafis. Melalui plot data dapat diketahui apakah data mengandung *trend*, musiman, *outlier*, atau variansi tidak *konstan* (Anityaloka, 2013: 1). Tahap selanjutnya untuk identifikasi model sementara sudah *stasioner* atau tidak. Karena model deret waktu umumnya menggunakan asumsi *stasioner*, diperlukan cara atau metode untuk menghilangkan ketidakstasioneran (menstasionerkan yang tidak stasioner) data sebelum melangkah lebih lanjut pada pembentukan model. Hal ini dapat dicapai melalui *differencing*.

Untuk menguji kestasioneran data dapat menggunakan Uji Akar Unit (Unit Root Test) dengan jenis uji ADF (Augmented Dickey Fuller) yaitu dengan melihat nilai Augmented Dickey Fuller (ADF). Jika nilai pada ADF lebih kecil daripada  $t$ -statistik pada nilai kritis berarti data tidak stasioner dan jika nilai ADF lebih besar dari  $t$ -statistik pada nilai kritis berarti data dikatakan stasioner (Kuncoro, 2007).

- Hipotesis yang digunakan dalam uji ini adalah :

$$H_0: \delta = 1 \text{ (Terdapat akar unit atau data tidak stasioner)}$$

$$H_1: \delta \neq 1 \text{ (Tidak terdapat akar unit atau data stasioner)}$$

- Statistik uji :

$$t = \frac{\hat{\delta}}{SE(\hat{\delta})} \quad (2.2)$$

- Taraf signifikan ( $\alpha$ ) = 5%
- Kriteria pengujian :

Jika nilai  $p - value >$  nilai kritis 5% maka  $H_0$  diterima atau  $H_1$  ditolak

Jika nilai  $p - value <$  nilai kritis 5% maka  $H_1$  diterima atau  $H_0$  ditolak

Jika data yang ada telah *stasioner*, maka langkah selanjutnya adalah membuat plot ACF (*Autocorrelation Function*) dan PACF (*Partial Autocorrelation Function*) untuk menentukan model sementara.

## 2. Pendugaan parameter

Model ARIMA yang baik adalah model yang menunjukkan bahwa penaksiran parameternya signifikan berbeda dengan nol (Aswi & Sukarna, 2006). Secara

umum, misalkan  $\theta$  adalah suatu parameter pada model ARIMA Box- Jenkins dan  $\hat{\theta}$  adalah nilai taksiran dari parameter tersebut, serta  $SE(\hat{\theta})$  adalah standar error dari nilai taksiran  $\hat{\theta}$ , maka uji kesignifikan parameter dapat dilakukan dengan hipotesis sebagai berikut :

- Hipotesis:

$$H_0: \theta = 0 \text{ (parameter tidak signifikan)}$$

$$H_1: \theta \neq 0 \text{ (parameter signifikan)}$$

- Statistik uji :

$$t = \frac{\hat{\theta}}{SE(\hat{\theta})} \quad (2.3)$$

- Taraf signifikan ( $\alpha$ ) = 5%

- Kriteria pengujian :

Jika nilai  $p - value >$  nilai kritis 5% maka  $H_0$  diterima atau  $H_1$  ditolak

Jika nilai  $p - value <$  nilai kritis 5% maka  $H_1$  diterima atau  $H_0$  ditolak

### 3. Uji Kesesuaian Model

Pada tahap ini dilakukan verifikasi kesesuaian model dengan sifat-sifat data serta dilakukan pemilihan model terbaik dengan uji Ljung-Box. Uji Ljung-Box digunakan untuk mengetahui apakah residual memenuhi asumsi white noise (residual tidak berkorelasi). Hipotesis uji *white noise* dapat dituliskan sebagai berikut.

- Hipotesis:

$$H_0: \rho_1 = \rho_2 = \dots = \rho_k = 0 \text{ (residual memenuhi syarat white noise)}$$

$$H_1: \text{minimal ada satu } \rho_j \neq 0, j = 0, 1, 2, \dots, K \text{ (residual tidak memenuhi syarat white noise)}$$

- Statistik uji : uji *Ljung-Box* atau *Box-Pierce Modified*:

$$Q = n(n + 2) \sum_{k=1}^K \frac{\hat{\rho}_k^2}{(n-k)} \quad (2.4)$$

- Taraf Signifikan ( $\alpha$ ) = 5%

- Kriteria pengujian :

Jika nilai  $p - value >$  nilai kritis 5% maka  $H_0$  diterima atau  $H_1$  ditolak

Jika nilai  $p - value <$  nilai kritis 5% maka  $H_1$  diterima atau  $H_0$  ditolak

#### 4. Peramalan

Sebelum peramalan terlebih dahulu ditentukan model yang akan digunakan untuk meramalkan data di masa yang akan datang. Dari hasil perhitungan peramalan, keakuratannya harus dilakukan pengukuran kesalahan dengan menggunakan parameter pengukuran dalam peramalan yaitu MAPE dan RMSE.

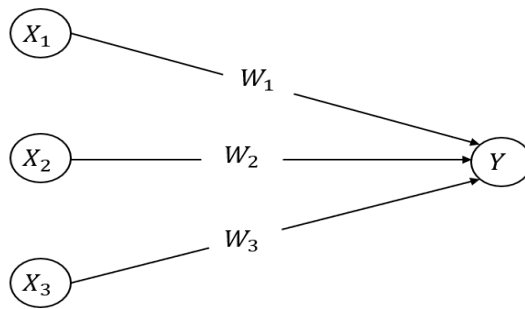
#### 2.4 Metode Jaringan Saraf Tiruan (JST)

Jaringan saraf tiruan merupakan sistem pemrosesan informasi yang memiliki kesamaan cara kerja dengan jaringan saraf biologis (Fausett, 1994). Jaringan saraf tiruan dibentuk sebagai generalisasi model matematika dari jaringan saraf biologis, dengan asumsi bahwa:

1. Pemrosesan informasi terjadi pada banyak elemen sederhana (*neuron*).
2. Sinyal dikirimkan diantara *neuron-neuron* melalui penghubung penghubung.
3. Penghubung antar *neuron* memiliki bobot yang akan memperkuat atau memperlemah sinyal.
4. Untuk menentukan *output*, setiap *neuron* menggunakan fungsi aktivasi (biasanya bukan fungsi linier) yang dikenakan pada jumlahan input yang diterima. Besarnya *output* ini selanjutnya dibandingkan dengan suatu batas ambang (*threshold*).

Menurut Puspitaningrum (2006) model Jaringan saraf tiruan ditentukan oleh 2 hal yaitu arsitektur jaringan dan algoritma pelatihan. Tugas arsitektur yaitu untuk memperjelas arah perjalanan yang ada dalam sebuah jaringan, Sedangkan algoritma pelatihan berfungsi untuk melatih bobot koneksi dan harus diubah jika diperlukan supaya target keluaran tercapai. Terdapat berbagai cara untuk melakukan suatu perubahan nilai bobot, tergantung jenis algoritma pelatihan yang dipakai. Perubahan bobot ini dilakukan supaya kerja jaringan dalam hal mempelajari banyak pola akan menghasilkan target keluaran yang meningkat, sebagai contoh pada gambar 2.1.





**Gambar 2. 1** Input dan Bobot

dengan

$X_1, X_2, X_3$  : nilai input

$W_1, W_2, W_3$  : bobot

$Y$  : output

Gambar 2.1 merupakan contoh gambar *artificial neural network* dengan simpul  $Y$  menerima masukan  $X_1, X_2, X_3$  dengan nilai bobot masing-masing  $W_1, W_2, W_3$ . Lalu ketiga sinyal simpul tersebut dijumlahkan  $net = X_1W_1 + X_2W_2 + X_3W_3$ . Besarnya sinyal yang diterima oleh  $Y$  tergantung fungsi aktivasinya  $y = (net)$ . Untuk melakukan perubahan bobot digunakan nilai fungsi aktivas.

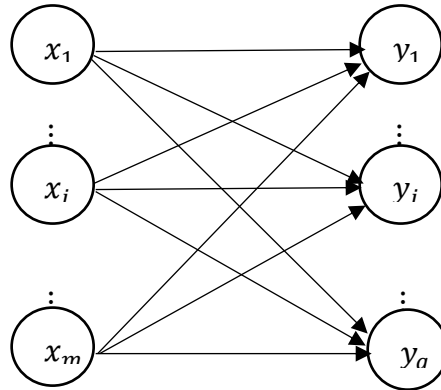
$$y_{net} = X_1W_1 + X_2W_2 + X_3W_3 \quad (2.5)$$

#### 2.4.1 Arsitektur Jaringan Saraf Tiruan

Arsitektur jaringan merupakan susunan beberapa unit didalam lapisan serta pola hubungan antar lapisan. Terdapat tiga jenis lapisan didalam sebuah jaringan. Lapisan *input* merupakan lapisan yang menerima sinyal *input* dari luar, lapisan *output* merupakan lapisan yang mengeluarkan sinyal keluar, sedangkan lapisan tersembunyi merupakan lapisan yang terletak diantara lapisan *input* dan *output*. Jaringan *feedforward* merupakan jaringan yang akan digunakan dimana terdiri dari dua jenis arsitektur (Fausett, 1994):

### 1. *Single Layer Network*

Single layer atau lapisan tunggal hanya memiliki satu lapisan bobot penghubung antara unit *input* dan unit *output*. Jaringan ini hanya menerima sinyal input kemudian secara langsung akan diolah menjadi *output*.



**Gambar 2. 2** Jaringan *Single layer*

Bentuk model matematis dari *single layer network* :

$$y_j = f \sum_{i=1}^q w_{ij} x_i + \varepsilon \quad (2.6)$$

Dimana;

$j$  = banyaknya unit output

$i$  = banyaknya unit input

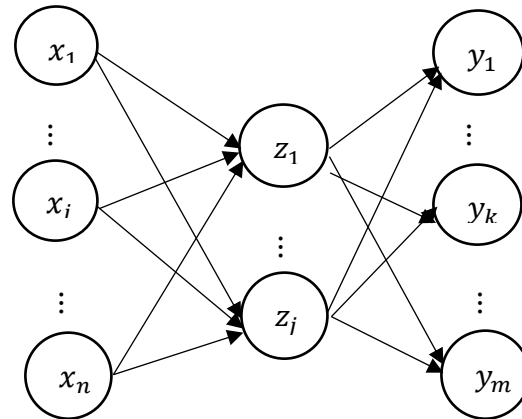
$w_{ij}$  = bobot dari neuron input ke-  $i$  menuju neuron output ke-  $j$

$f$  = fungsi aktivasi

### 2. *Multilayer Network*

Jaringan ini terdiri dari satu atau lebih lapisan tersembunyi (dengan beberapa unit tersembunyi) antara unit *input* dan unit *output*. *Multilayer network* dapat memecahkan permasalahan yang lebih kompleks dibandingkan dengan lapisan tunggal. Namun, membutuhkan proses pelatihan yang lebih sulit.

Bentuk model matematis dari *single layer network* .



**Gambar 2. 3** Jaringan *Multi Layer*

Bentuk model matematis dari *single layer network*

$$y_k(x_1, x_2, \dots, x_m) = f_2(\sum_{j=1}^j f_1(\sum_{i=1}^n x_i v_{ij} + \beta_j) w_{jk} + \gamma_k) + \varepsilon \quad (2.7)$$

Dimana;

$k$  = banyaknya *unit output*

$i$  = banyaknya *unit input*

$v_{ij}$  = bobot dari *neuron input* ke-  $i$  menuju *neuron hidden* ke-  $j$

$w_{jk}$  = bobot dari *neuron hidden* ke-  $j$  menuju *neuron output* ke-  $k$

$f$  = fungsi aktivasi

$\beta_j, \gamma_k$  = bias

#### 2.4.2 Fungsi Aktivasi

Penelitian yang dilakukan oleh Ilyas, N., dkk (2022), Sharma dan Vipul (2015) fungsi aktivasi yang digunakan adalah fungsi aktivasi sigmoid biner. Fungsi aktivasi pada Jaringan saraf tiruan *Backpropagation* digunakan untuk memformulasikan *output* dari setiap neuron. Pada Jaringan saraf tiruan yang sering digunakan adalah fungsi aktivasi sigmoid biner. Hal yang penting dari fungsi ini adalah sebesar apapun nilai  $x$  (positif atau negatif) *output* dari fungsi ini tidak pernah mencapai 0.0 atau 1.0. Hasil dari fungsi ini bernilai antara 1 dan 0 (Siang, 2009).

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.8)$$

Keterangan:

$e$  : bilangan euler 2,71828

$x$  : hasil penjumlahan dari sinyal *input*

$y$  : fungsi untuk mengaktifkan nilai  $x$

$y'$  : turunan dari  $f(x)$

Fungsi sigmoid biner memiliki nilai maksimum  $< 1$ . Maka untuk pola yang targetnya  $> 1$ , pola masukan dan keluaran harus terlebih dahulu ditransformasi atau dinormalisasi sehingga semua polanya memiliki *range* yang sama seperti fungsi sigmoid biner.

Pada proses peramalan menggunakan Jaringan saraf tiruan *Backpropagation*, sebelum dilakukan pelatihan data *input* dan target *output* harus ditransformasi terlebih dahulu. Normalisasi adalah penskalaan terhadap nilai-nilai masuk ke dalam suatu range tertentu. Hal ini dilakukan agar nilai *input* dan target *output* sesuai dengan *range* dari fungsi aktivasi yang digunakan dalam jaringan. Normalisasi data dengan menggunakan interpolasi biasa, sehingga nilainya berada pada interval  $a$  sampai  $b$  (Siang,2009) :

$$\frac{X' - a}{b - a} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2.9)$$

$$X' - a = (b - a) \left[ \frac{x - x_{min}}{x_{max} - x_{min}} \right]$$

$$X'_i = \frac{(b - a)(x - x_{min})}{x_{max} - x_{min}} + a \quad (2.10)$$

Fungsi aktivasi yang digunakan adalah fungsi aktivasi sigmoid biner dimana nilainya berada pada interval (0,1) yaitu tidak pernah tepat menyentuh 0 atau 1, maka data akan dinormalisasi ke interval 0,1 sampai 0,9. Normalisasi yang digunakan adalah normalisasi linear pada selang interval  $[a,b]$  adalah sebagai berikut:

$$\begin{aligned}
 X'_i &= \frac{(b - a)(x - x_{min})}{x_{max} - x_{min}} + a \\
 X'_i &= \frac{(0.9 - 0.1)(x - x_{min})}{x_{max} - x_{min}} + 0.1 \\
 X'_i &= \frac{0.8(x - x_{min})}{x_{max} - x_{min}} + 0.1
 \end{aligned} \tag{2.11}$$

Keterangan :

$X'$  : Data hasil normalisasi; ( $i = 1, 2, \dots, n$ )

$x$  : Data asli

$x_{min}$  : Data minimum

$x_{max}$  : Data maksimum

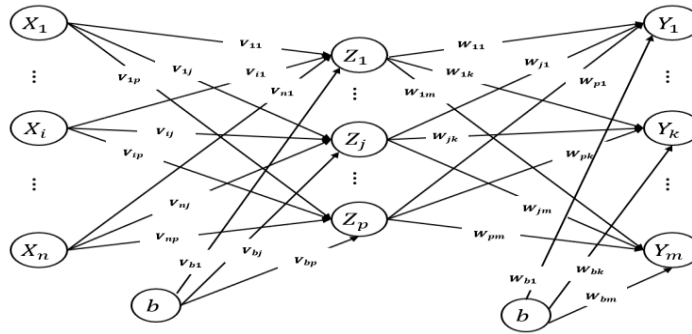
$a$  : Nilai interval terendah (0.1)

$b$  : Nilai interval tertinggi (0.9)

### 2.4.3 Backpropagation

*Backpropagation* merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan *neuron-neuron* yang ada pada lapisan tersembunyinya. Algoritma *backpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward propagation*). Untuk mendapatkan *error* ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju,  $i$  diaktifkan dengan menggunakan fungsi Sigmoid Biner.

*Backpropagation* memiliki beberapa unit neuron yang ada dalam satu atau lebih layer tersembunyi. Arsitektur *backpropagation* dengan  $n$  buah masukan (ditambah sebuah bias), sebuah layer tersembunyi yang terdiri dari  $p$  unit (ditambah sebuah bias), serta  $m$  buah unit keluaran.  $v_{ij}$  merupakan bobot garis dari unit masukan  $x_i$  ke unit *hidden layer*  $z_j$  ( $v_{0j}$  merupakan bobot garis yang menghubungkan bias di unit masukan ke unit layer tersembunyi  $z_j$ ).  $w_{jk}$  merupakan bobot dari unit layer tersembunyi  $z_j$  ke unit keluaran  $y_k$  ( $w_{0k}$  merupakan bias di layer tersembunyi ke unit keluaran  $z_k$ ) (Fauset, L. 1994).



**Gambar 2. 4** Jaringan *Backpropogation*

Algoritma selengkapnya pelatihan jaringan *backpropagation* dengan fungsi *Sigmoid Biner* adalah sebagai berikut:

- a. Langkah 0 : Inisialisasi semua bobot dengan bilangan acak kecil
- b. Langkah 1 : Jika kondisi berhenti belum terpenuhi, kerjakan langkah 2-9.
- c. Langkah 2 : Untuk setiap pasang data pelatihan, kerjakan langkah 3-8.

**Fase I: Propagasi Maju**

- d. Langkah 3 : Tiap unit input ( $x_i, i = 1, 2, \dots, n$ ) menerima sinyal dan diteruskan ke unit-unit tersembunyi.
- e. Langkah 4 : Tiap unit tersembunyi ( $z_j, j = 1, 2, \dots, p$ ) menjumlahkan isyarat masukan terbobot,

$$z_{net_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \tag{2.12}$$

Kemudian menghitung sinyal *output* menggunakan fungsi aktivasi hitung;

$$z_j = f(z_{net_j}) = \frac{1}{1+e^{-z_{net_j}}} \tag{2.13}$$

Sinyal *output* ini kemudian dikirim ke seluruh unit-unit *output*.

- f. Langkah 5 : Tiap unit tersembunyi ( $y_k, k = 1, 2, \dots, m$ ) menjumlahkan isyarat masukan terbobot,

$$y_{net_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \tag{2.14}$$

Kemudian menghitung sinyal *output* menggunakan fungsi aktivasi hitung;

$$y_k = f(y_{net_k}) = \frac{1}{1+e^{-y_{net_k}}} \tag{2.15}$$

**Fase II: Propagasi Mundur**

- g. Langkah 6 : Tiap unit keluaran ( $y_k, k = 1, 2, \dots, m$ ) menerima pola sasaran berkaitan dengan pola pelatihan masukannya. Hitung faktor kesalahannya;

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \quad (2.16)$$

Kemudian hitung nilai perubahan bobot yang nantinya digunakan untuk memperbaiki nilai bobot antara lapisan tersembunyi dan lapisan *output* ( $w_{kj}$ ) dengan laju percepatan  $\alpha$  yaitu:

$$\Delta w_{kj} = \alpha \delta_k z_j \quad ; \quad k = 1, 2, \dots, m \quad ; \quad j = 0, 1, \dots, p \quad (2.17)$$

Hitung juga perubahan bobot bias yang digunakan untuk memperbaiki nilai bias antara lapisan tersembunyi dan lapisan *output* ( $w_{0k}$ ), yaitu:

$$\Delta w_{0k} = \alpha \delta_k \gamma \quad (2.18)$$

- h. Langkah 7: Tiap unit tersembunyi ( $z_i, j = 1, 2, \dots, p$ ) menjumlahkan delta masukannya (dari unit-unit lapisan atasannya) yang sudah berbobot. Hitung faktor kesalahannya

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (2.19)$$

Kemudian hasilnya dikalikan dengan turunan hasil fungsi aktivasi yang digunakan menghitung informasi kesalahan *error*  $\delta_j$  dimana:

$$\delta_j = \delta_{net_j} f'(z_{net_j}) = \delta_{net_j} z_j (1 - z_j) \quad (2.20)$$

Kemudian hitung koreksi bobot dan prasikapnya:

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.21)$$

Menghitung perubahan bias

$$\Delta V_{0j} = \alpha \delta_j \quad (2.22)$$

### Fase III: Perubahan Bobot

- i. Langkah 8 : Tiap unit keluaran ( $y_k, k = 1, 2, \dots, m$ ) memperbarui perubahan bobot-bobotnya ( $j = 0, 1, \dots, p$ )

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad ; \quad k = 1, 2, \dots, m \quad ; \quad j = 0, 1, \dots, p \quad (2.23)$$

Tiap unit tersembunyi ( $z_i, j = 1, 2, \dots, p$ ) memperbarui perubahan bobot-bobotnya ( $i = 0, 1, \dots, n$ )

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad ; \quad (j = 1, 2, \dots, p \quad ; \quad i = 0, 1, \dots, n) \quad (2.24)$$

- j. Langkah 9 : Uji syarat berhenti apabila *error* ditemukan.

## 2.5 Metode *Hybrid* ARIMA-JST

Model hybrid merupakan suatu metode kombinasi dari satu atau lebih dari dua model dalam fungsi suatu sistem. Model ARIMA dan Jaringan saraf tiruan merupakan model untuk mengatasi masalah linier atau nonlinier. Karena pada kenyataannya jarang ditemukan kejadian time series yang murni linier maupun murni nonlinier maka model *hybrid* yang mempunyai kemampuan memodelkan linier dan nonlinier dapat dijadikan strategi yang praktis. Secara umum, kombinasi dari model deret waktu yang memiliki struktur *autokorelasi linier* dan *nonlinier* dapat dituliskan:

$$Y_t = L_t + N_t \quad (2.24)$$

dengan :  $L_t$  = Komponen *linier*

$N_t$  = Komponen *nonlinier*

*Residual* dari model *linier* dapat dituliskan:

$$e_t = Y_t - \hat{L}_t \quad (2.25)$$

$\hat{L}_t$  merupakan nilai ramalan ARIMA pada waktu  $t$ .

*Residual* penting dalam mendiagnosa model-model linier. Dengan memodelkan residual menggunakan jaringan syaraf tiruan, hubungan *nonlinier* dapat ditemukan. Sehingga model jaringan saraf tiruan untuk *residual* dengan *input* akan menjadi:

$$e_t = f(e_{t-1}, e_{t-2}, \dots, e_{t-n}) + \varepsilon_t \quad (2.26)$$

$f$  merupakan fungsi nonlinier yang dijelaskan oleh JST dan  $\varepsilon_t$  adalah *error* yang acak. Persamaan diatas ditulis sebagai  $\hat{N}_t$ . Sehingga, peramalan *hybrid* merupakan kombinasi dari dua komponen yang kemudian ditulis sebagai berikut:

$$\hat{Y}_t = \hat{L}_t + \hat{N}_t \quad (2.27)$$

## 2.6 Ukuran Ketepatan Metode Peramalan

Ukuran ketepatan metode peramalan dilakukan untuk mengukur ketepatan suatu metode peramalan berdasarkan kesalahan dari peramalan tersebut. Dalam praktek ada beberapa alat ukur yang sering digunakan untuk menghitung kesalahan



prediksi, salah satunya adalah *Root Mean Square Error* (RMSE) dan *Mean Absolute Percentage Error* (MAPE) dengan persamaan berikut:

- *Root Mean Square Error* (RMSE)

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^N \left| \frac{Y_t - \hat{Y}_t}{\hat{Y}_t} \right|^2}{N}} \quad (2.28)$$

- *Mean Absolute Percentage Error* (MAPE)

$$\text{MAPE} = \frac{100\%}{N} \sum_{t=1}^N \left| \frac{Y_t - \hat{Y}_t}{\hat{Y}_t} \right| \quad (2.29)$$

(Makridakis, 1999: 59)

Semakin kecil nilai MAPE atau RMSE maka nilai taksiran semakin mendekati dengan nilai data actual atau dengan kata lain metode yang dipilih merupakan metode yang terbaik. Sebuah metode mempunyai kinerja sangat bagus apabila nilai MAPE berada di bawah 10% dan mempunyai kinerja bagus jika nilai berada di antara 10% dan 20% (Pusparinda, 2017).

## 2.7 Emas

Emas merupakan salah satu instrumen simpanan pokok (investasi) yang paling stabil dan efektif. Selain itu, emas juga sering disebut sebagai pengukur modal dan dimensi kekayaan yang paling tua dan efektif. Sejak emas lebih unggul dibanding logam lain dan diperdagangkan lebih sering dalam sistem keuangan, harga dan hubungan dengan berbagai variabel keuangan sering dipantau oleh unit ekonomi (Suharto, 2015). Emas digunakan sebagai standar keuangan di banyak negara dan juga sebagai alat tukar yang relatif abadi, dan diterima di semua negara di dunia. Penggunaan emas dalam bidang moneter dan keuangan berdasarkan nilai moneter absolut dari emas itu sendiri terhadap berbagai mata uang di seluruh dunia, meskipun secara resmi di bursa komoditas dunia, harga emas dicantumkan dalam mata uang dolar Amerika. Bentuk penggunaan emas dalam bidang moneter lazimnya berupa batangan emas dalam berbagai satuan berat gram sampai kilogram (Kurniawan, 2019).

### 2.7.1 Faktor –faktor yang Mempengaruhi Harga Emas

Faktor-faktor yang mempengaruhi harga emas menurut Abdullah (2012) adalah:

1. Inflasi yang meningkat melebihi prediksi.
2. Kericuhan finansial. Krisis moneter pada tahun 1998 dan 2008 termasuk ke dalam kericuhan atau kepanikan finansial.
3. Kenaikan harga minyak yang signifikan.
4. Permintaan emas. Harga emas akan terus naik jika permintaan emas dunia yang terus naik berbanding terbalik dengan pasokan emas yang ada.
5. Kondisi politik di dunia. Ketidakpastian ekonomi adalah akibat dari suhu politik dunia yang tinggi karena ketegangan yang terjadi antar negara-negara di dunia.
6. Perubahan kurs. Melemahnya kurs dollar AS dapat mendorong kenaikan harga emas dunia.

## BAB 3 METODOLOGI PENELITIAN

### 3.1 Sumber Data

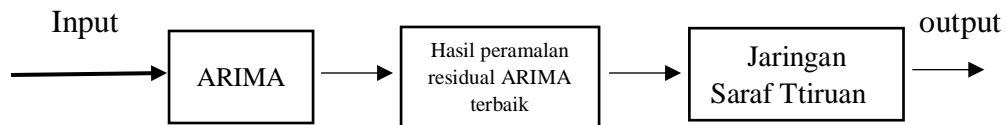
Data yang digunakan dalam penelitian ini adalah data historis sekunder yang diambil dari website harga emas hari ini (<https://harga-emas.org/>). Data tersebut merupakan data harga emas setiap minggunya dari tahun ke tahun yang berupa time series untuk harga emas di Indonesia yang berfluktuasi terhadap dollar terhitung sejak Januari 2016 sampai dengan Desember 2020.

### 3.2 Identifikasi Variabel

Variabel yang digunakan dalam penelitian ini adalah harga emas .

### 3.3 Metode Analisis

Secara garis besar metode analisis pada peramalan data harga emas menggunakan metode ARIMA dan *Hybrid* ARIMA Jaringan saraf tiruan yang akan dibentuk ditunjukkan pada gambar 3.1.



**Gambar 3. 1** Arsitektur Sistem peramalan hybrid ARIMA Jaringan Saraf Tiruan

#### 3.3.1 ARIMA

Langkah-langkah analisis data dalam analisis ARIMA sebagai berikut:

##### 1. Pembagian Data

Data dibagi menjadi dua bagian yaitu data training dan data validation. Data training sebanyak 207 data (Januari 2016 – Desember 2019) digunakan untuk pembentukan model sedangkan 52 data (Januari 2020 – Desember 2020) untuk data validation digunakan untuk validasi model yang diperoleh.

##### 2. Uji kestasioneran

Uji kestasioneran data dapat dilakukan dengan beberapa cara diantaranya dengan melihat plot time series, plot ACF dan PACF serta melakukan uji ADF. Jika data tidak stasioner, maka dilakukan transformasi maupun *differencing*.

3. Identifikasi model

Setelah data stasioner, maka dilakukan pembentukan model ARIMA sementara yang sesuai dengan melihat plot data yang sudah stasioner.

4. Pendugaan Parameter

Model ARIMA sementara yang sudah ditentukan akan dilakukan penaksiran parameternya apakah telah signifikan atau tidak.

5. Pemeriksaan kelayakan model

Pada tahap ini dilakukan verifikasi kesesuaian model dengan sifat-sifat data serta dilakukan pemilihan model terbaik dengan uji Ljung-Box. Uji Ljung-Box digunakan untuk mengetahui apakah residual memenuhi asumsi white noise (residual tidak berkorelasi).

6. Pemilihan model ARIMA

Pemodelan data time series yang sesuai harus memenuhi syarat, yaitu residual memenuhi asumsi white noise. Namun, pemilihan model terbaik dari beberapa model yang telah memenuhi syarat pada penelitian ini akan digunakan kriteria RMSE untuk mengukur kesalahan peramalan, yaitu dengan memilih model yang memiliki nilai RMSE terkecil.

7. Peramalan

### 3.3.2 Jaringan Saraf Tiruan

Langkah-langkah analisis data dalam analisis Jaringan saraf tiruan sebagai berikut:

1. Menetapkan tujuan system

Model jaringan yang dibangun digunakan untuk meramalkan data harga jual emas harian pada bulan Januari 2020 – Desember 2020 berdasarkan data bulan Januari 2016 sampai Desember 2019 dengan bantuan software Matlab.

2. Menentukan fungsi aktivasi

Fungsi *sigmoid biner* (*log sigmoid*) digunakan sebagai fungsi aktivasi.

3. Transformasi Data

Dilakukan transformasi data agar kestabilan taburan data dicapai dan juga untuk menyesuaikan nilai data dengan range fungsi aktivasi yang digunakan dalam jaringan.

#### 4. Pembagian Data

Data *residual* dari pergerakan data harga emas dari bulan Januari 2016 sampai Desember 2019 dibagi menjadi 80% data pelatihan dan 20% data pengujian.

#### 5. Perancangan arsitektur jaringan yang optimum

Perancangan arsitektur jaringan dilakukan dengan menentukan jumlah *neuron input*, *neuron* lapisan tersembunyi, simpul lapisan tersembunyi berikutnya dan simpul keluaran yang akan digunakan dalam jaringan. Penentuan ini dilakukan dengan *trial* dan *error*.

6. Pemilihan arsitektur jaringan yang optimum sehingga siap digunakan pada peramalan.

### 3.3.3 Metode *Hybrid* ARIMA Jaringan Saraf Tiruan

Metode *hybrid* ARIMA Jaringan saraf tiruan merupakan metode penggabungan antara metode ARIMA dengan metode jaringan syaraf tiruaan. Data diproses dengan menggunakan metode ARIMA terlebih dahulu untuk mendapatkan residual dari model ARIMA terbaik. Hal ini dikarenakan data yang digunakan untuk metode *hybrid* ARIMA Jaringan saraf tiruan adalah data hasil peramalan dari residual model ARIMA. Dalam penelitian ini komposisi pembagian data yang digunakan adalah 80% untuk data training dan 20% untuk data test.

## BAB 4

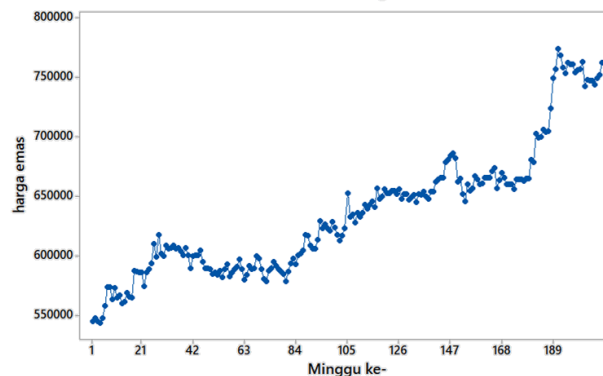
### HASIL DAN PEMBAHASAN

Pada bab ini akan dilakukan analisis deret waktu dengan menggunakan model ARIMA dan *Hybrid* ARIMA Jaringan saraf tiruan. Data yang digunakan adalah data harga emas antam periode Januari 2016 – Desember 2019 yang terdapat pada Lampiran 1 untuk mendapatkan hasil peramalan ARIMA dan *Hybrid* ARIMA Jaringan saraf tiruan. Berikut ini merupakan tahapan untuk mendapatkan model ARIMA dan *Hybrid* ARIMA Jaringan saraf tiruan terbaik serta peramalan berdasarkan model terbaik.

#### 4.1 Peramalan Menggunakan Model ARIMA

##### 4.1.1 Identifikasi Kestasioneran

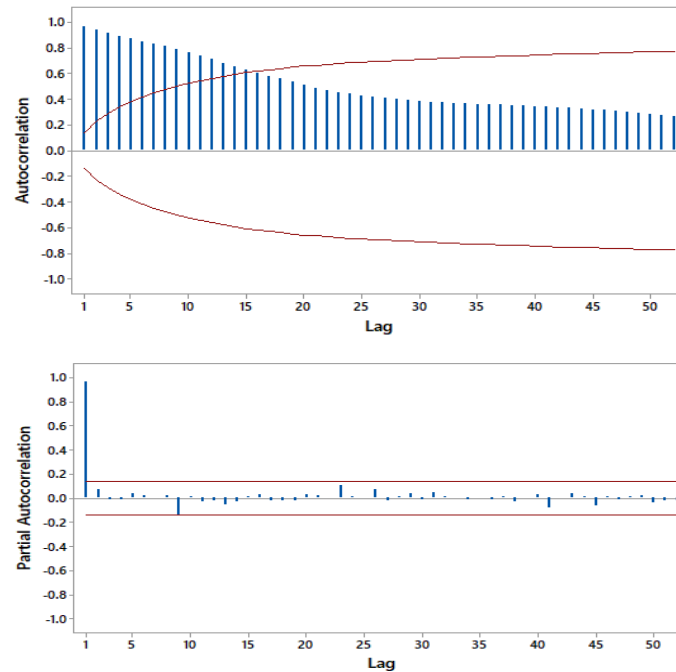
Tahap awal yang dilakukan untuk pemodelan data deret waktu adalah dengan mengidentifikasi kestasionerannya. Identifikasi kestasioneran data dapat dilakukan dengan membuat plot deret waktu, plot ACF plot PACF dan melakukan uji *Augmented Dickey Fuller* (ADF). Langkah pertama yang dilakukan adalah membuat plot data untuk melihat apakah data mempunyai suatu tren dan telah memenuhi stasioneritas. Plot deret waktu dari data kurs pada Lampiran 1 ditunjukkan pada Gambar 4.1 berikut:



**Gambar 4. 1** Plot Deret Waktu Data Harga Emas

Gambar 4.1 menampilkan bahwa data harga emas menunjukkan terjadinya pola tren naik. Unsur tren data dapat dilihat tidak berada diantara garis rata-rata atau konstan sehingga data dapat dikatakan data tidak stasioner dalam rata-rata. Selain melihat

grafik deret Plot data, kestasioneran data juga dapat dilihat dari grafik plot ACF (*Autocorrelation Function*) dan PACF (*Partial Autocorrelation Function*) pada data aktual seperti pada gambar 4.2 dan 4.3.



**Gambar 4. 2** ACF dan PACF Data Harga Emas

Gambar 4.2 menunjukkan bahwa hasil grafik ACF pada *lag* 1 sampai dengan *lag* 16 berada pada luar garis signifikan yang ditunjukkan pada garis yang ada diatas maupun dibawah garis tengah dan menurun secara perlahan menuju nol (hampir linear) sehingga data dikatakan tidak stasioner. Grafik PACF (Partial Autocorrelation Function) menunjukkan data *cut-off* pada *lag* 1. Uji stasioneritas data juga dapat menggunakan Uji Akar Unit (Unit Root Test) dengan jenis uji ADF. Hasil uji ADF pada lampiran 3 data aktual ditunjukkan pada pada tabel 4.1:

**Tabel 4. 1** Pengujian Stasioner Data Harga Emas

Statistik-t Uji ADF	Probabilitas	Nilai Kritis Mac Kinnon			Ket
		1%	5%	10%	
-0.388165	0.9076	-3.461783	-2.875262	-2.574161	Tidak stasioner

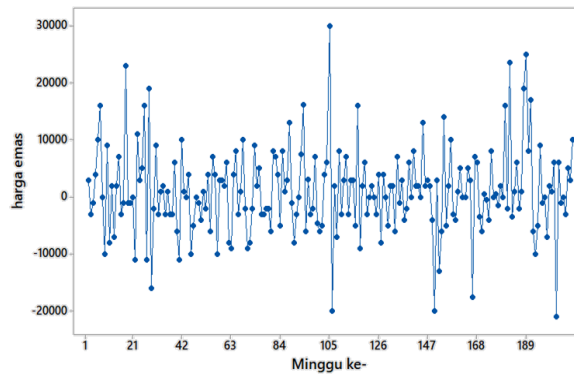
Sumber : Hasil olah data (2023)

Adapun hipotesisnya sebagai berikut:

$$H_0: \delta = 0 \text{ (data tidak stasioner)}$$

$$H_1: \delta < 0 \text{ (data stasioner)}$$

Tabel 4.1 menunjukkan nilai  $t_{hit} < t_{tabel}$  dan  $p - value = 0.9086 > \alpha$ , maka  $H_0$  diterima yang menyatakan bahwa data harga emas tidak stasioner. Hal ini menunjukkan bahwa perlu dilakukan proses *differencing* agar data menjadi stasioner. Setelah dilakukan proses *differencing lag 1*, maka diperoleh *plot* deret waktu berikut:



**Gambar 4. 3** Plot Data Harga Emas Differencing 1

Gambar 4.3 menunjukkan bahwa menunjukkan data telah berfluktuasi di sekitar rata-rata yang konstan atau tidak terjadi perubahan kecenderungan dalam rata-rata sehingga dapat dikatakan bahwa data harga emas hasil *differencing lag 1* telah stasioner. Hasil *differencing lag 1* data harga emas dicek kembali menggunakan uji ADF dengan hasil pada lampiran 3 yang ditampilkan pada tabel 4.2.

**Tabel 4. 2** Pengujian Stasioner Data Harga Emas *Differencing 1*

Statistik-t Uji ADF	Probabilitas	Nilai Kritis Mac Kinnon			Ket
		1%	5%	10%	
-16.58439	0.0000	-3.461783	-2.875262	-2.574161	stasioner

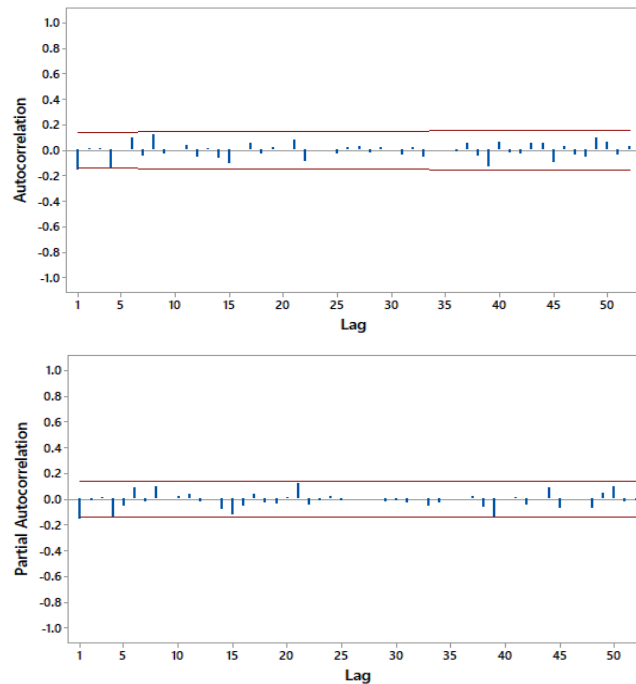
**Sumber :** Hasil olah data (2023)



Tabel 4.2 menunjukkan nilai  $t_{hit} < t_{tabel}$  dan  $p - value = 0.000 < \alpha$ , maka  $H_0$  ditolak yang menyatakan bahwa data harga emas telah stasioner, maka data harga emas dapat digunakan dalam tahap pengidentifikasian model.

#### 4.1.2 Identifikasi Model ARIMA

Setelah kondisi stasioner data telah terpenuhi maka selanjutnya akan dilakukan identifikasi model ARIMA sementara. Identifikasi model ARIMA adalah suatu tahap untuk menentukan orde (p,d,q) dengan pendugaan sementara melihat pola ACF dan PACF hasil differensiasi yang dilakukan sebanyak satu kali (d=1). Pola ACF dan PACF hasil *differencing* data ditunjukkan pada Gambar 4.6



**Gambar 4. 4** Plot ACF dan PACF setelah differencing 1

Dari Gambar 4.4 dapat dilihat bahwa plot ACF dan PACF stasioner. Pada Plot ACF mengalami *cut off* setelah lag 1 begitu pun dengan PACF mengalami *cut off* setelah lag 1. Dengan demikian, model awal yang diperoleh dari kedua *plot* tersebut yaitu : ARIMA (0,1,1), ARIMA (1,1,0), dan ARIMA (1,1,1).

### 4.1.3 Pemeriksaan Diagnostik Model ARIMA

#### 1. Uji Signifikansi Parameter

Uji signifikansi parameter dapat dilakukan dengan melakukan uji t. Hipotesis yang diuji sebagai berikut:

$$H_0: \hat{\theta} = 0 \text{ (parameter tidak signifikan)}$$

$$H_1: \hat{\theta} \neq 0 \text{ (parameter signifikan)}$$

Hasil uji signifikansi parameter masing-masing model ARIMA sementara berdasarkan lampiran 2 sebagai berikut:

**Tabel 4. 3** Uji Signifikansi Parameter Model ARIMA

Model	Parameter	Koefisien	t-hitung	p-value	Ket.
ARIMA (1,1,0)	$\phi_1$	-0.1578	-2.16	0.032	Signifikan
ARIMA (1,1,1)	$\phi_1$	-0.0524	-0.32	0.747	Tidak Signifikan
	$\theta_1$	0.1079	-0.00	0.999	Tidak Signifikan
ARIMA (0,1,1)	$\theta_1$	0.1599	2.12	0.035	Signifikan

**Sumber :** Hasil olah data (2023)

Tabel 4.3 menunjukkan bahwa terdapat dua model yang signifikan yaitu ARIMA (1,1,0) dan ARIMA (0,1,1). Kedua model tersebut memiliki nilai  $p$  –  $value < \alpha = 0.05$  yang artinya tolak  $H_0$ .

#### 2. Uji Kesesuaian Model

##### a. Uji Sisa *White Noise*

Pengujian asumsi *residual white noise* dapat dilakukan dengan menggunakan uji Ljung-Box. Hipotesis yang diuji adalah:

$$H_0: \rho_1 = \rho_2 = \dots = \rho_k = 0 \text{ (sisa memenuhi syarat white noise)}$$

$$H_1: \text{minimal ada satu } \rho_j \neq 0, j = 1, 2, \dots, k \text{ (sisa tidak white noise)}$$

Hasil uji sisa white noise berdasarkan output pada Lampiran 3 menunjukkan bahwa nilai  $p\text{-value} > \alpha = 0,05$  untuk kedua model ARIMA, yang berarti model sudah memenuhi syarat white noise.

b. Uji Asumsi Distribusi Normal

Pengujian asumsi distribusi normal dapat dilakukan dengan menggunakan uji Kolmogorov Smirnov. Hipotesis yang diuji adalah:

$H_0$ : residual data berdistribusi normal

$H_1$ : residual data berdistribusi tidak normal

Dimana  $H_0$  akan ditolak apabila  $D > D_{(\alpha,n)}$

Hasil uji Kolmogorov smirnov menunjukkan bahwa nilai  $D > D_{(\alpha,n)}$

**Tabel 4. 4** Uji Kolmogorov-smirnov

Model	D	$D_{(\alpha,n)}$	Hasil Uji
ARIMA (1,1,0)	0.083	0.094073	Berdistribusi Normal
ARIMA (0,1,1)	0.086	0.094073	Berdistribusi Normal

**Sumber** : Hasil olah data (2023)

Berdasarkan Tabel 4.4 kedua model ARIMA yaitu ARIMA (1,1,0) dan ARIMA (0,1,1) memiliki nilai  $D > D_{(\alpha,n)}$  yang artinya  $H_0$  diterima sehingga kedua model tersebut berdistribusi normal.

**4.1.4 Pemilihan Model Terbaik**

Pemilihan model ARIMA dilakukan karena terdapat lebih dari satu model ARIMA yang lolos uji diagnostik. Model terbaik merupakan model yang mempunyai nilai MAPE dan RMSE paling kecil. Berikut ini nilai MAPE dan RMSE untuk dua model yang lolos uji diagnostik:

**Tabel 4. 5** Nilai MAPE dan RMSE Pada Model ARIMA

Model	MAPE	RMSE
ARIMA (1,1,0)	14.64743702	158524.2342
ARIMA (0,1,1)	14.67157176	158730.3221

**Sumber** : Hasil olah data (2023)

Tabel 4.5 menunjukkan bahwa model ARIMA (1,1,0) merupakan model yang memiliki nilai MAPE dan RMSE terkecil, maka model ARIMA (1,1,0) yang akan digunakan untuk meramalkan harga emas.

## 4.2 Peramalan Menggunakan *Hybrid Jaringan Saraf Tiruan*

### 4.2.1 Normalisasi Data

Normalisasi data dilakukan agar jaringan yang akan menjalani proses pembelajaran tidak mengalami kegagalan (pelatihan atau pengujian). Proses ini sangat dibutuhkan didalam melakukan pembelajaran yang menggunakan fungsi aktifasi sigmoid. Data dinormalisasi dengan interval [0,1] yang merupakan range dari fungsi sigmoid biner. Pada penelitian ini data dinormalisasi ke interval yang lebih kecil, yakni pada interval 0,1 - 0,9. Hasil normalisasi data terlihat pada lampiran 3, menggunakan rumus pada persamaan (2.11):

**Tabel 4. 6** Hasil Normalisasi Harga Emas

No	Tanggal	Harga emas (Rp)
1	4-Jan-16	0.38034
2	11-Jan-16	0.39751
3	18-Jan-16	0.47983
⋮	⋮	⋮
205	2-Dec-19	0.49073
206	9-Dec-19	0.4782
207	16-Dec-19	0.58232

**Sumber:** Hasil olah data (2023)

Tabel 4.6 menunjukan data hasil normalisasi harga emas yang akan diolah dengan metode Jaringan saraf tiruan *Backpropagation* untuk meramalkan data harga emas, dimana data tersebut dibagi dalam dua bagian yaitu data latih dari 4 Januari 2016 sampai 4 Maret 2019 dan data uji dari 11 Maret 2019 sampai 30 Desember 2019.

### 4.2.2 Perancangan Arsitektur Jaringan Saraf Tiruan

Pada jaringan saraf tiruan proses analisis menggunakan model *Backpropagation* pada *input layer* tidak bisa ditentukan secara langsung, tetapi bisa dilihat pada *plot* ACF dan PACF, pada data residual harga emas *plot* ACF dan

PACF terpotong (*cut off*) pada *lag* 4, maka akan di uji *lag* 1 sampai *lag* 4. Kemudian dimensi *input*, *output*, dan matriks data uji menggunakan metode *sliding window* seperti tabel 4.7.

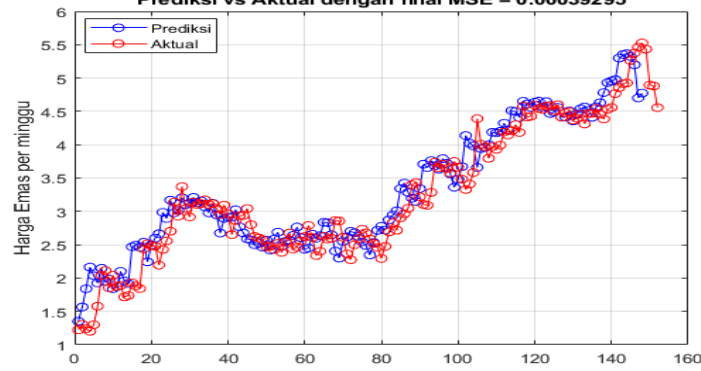
**Tabel 4. 7** Dimensi Matriks Harga Emas

Jumlah Minggu sebagai Input	Dimensi matriks		
	Data input Latih	Data output	Input Data Uji
H1	1x167	1x167	1x491
H1 – H2	2x166	1x166	2x490
H1 – H3	3x165	1x165	3x489
H1 – H4	4x164	1x164	4x487

**Sumber:** Hasil olah data (2023)

Tabel 4.7 menunjukkan bahwa, jika jumlah minggu adalah sebagai *input* (H1) maka *input* pada data latih dan data *output* akan dikurangi 1 dan seterusnya. Dimana hasil pelatihan menunjukkan model terbaik adalah minggu sebelum minggu perdagangan, data input pada tahap pelatihan adalah berjumlah 166 data pengujian berjumlah 42 data.

Proses pelatihan dan pengujian pada tahap ini menggunakan 6 variasi jumlah neuron pada *hidden layer*, yakni 5, 10, 15, 20, 25 dan 30 *neuron*. Model hasil pelatihan tentang variasi dalam jumlah *hidden layer* digunakan untuk menguji data uji. Hasil pengujian menunjukkan bahwa nilai MSE terkecil pada *residual* harga emas dengan banyaknya *input* 4 minggu sebelum minggu perdagangan, dengan 20 neuron pada lapisan tersembunyi yaitu 0.00039295. 4 *input layer* 20 *hidden layer*, dilakukan perhitungan karena memiliki nilai MSE terkecil, hasil pengujian ditunjukkan pada gambar 4.5.



**Gambar 4. 5** Perbandingan nilai prediksi dan aktual 4 *input*, 20 *hidden layer* dan 1 *output*

a. Inisialisasi Bobot

Inisialisasi bobot merupakan tahap awal sebelum melakukan prediksi harga emas, hasil data latih adalah inisialisasi bobot awal. Pada inisialisasi bobot secara random, bobot di inisialisasi secara acak tanpa menggunakan faktor skala (Kholis & Rofii, 2017). Pada penelitian ini peneliti menggunakan dengan teknik random. Inisialisasi bobot dan bias awal dari *input layer* ke *hidden layer*, *hidden layer* ke *output layer*, dimana dalam penelitian ini digunakan *input layer* ke *hidden layer* X1 sampai dengan X4 karena pada *input layer* proses pelatihan jaringan yang memiliki nilai MSE terkecil terdapat pada 4 *input layer*, 20 *hidden layer* adalah sebagai berikut:

**Tabel 4. 8** Bias dan bobot awal pada *Input Layer* Terhadap *Hidden Layer*

$V_j$ ( <i>hidden</i> )	<i>bias</i>	$V_i$			
		X1	X2	X3	X4
1	-3.05401	-7.78358	7.092586	6.205308	20.40012
2	-0.92191	2.768269	-16.7768	-11.2712	12.23531
3	10.30067	-14.3323	2.756734	-15.3136	10.8137
4	11.87247	-17.232	11.22701	-9.61179	-7.14124
5	-10.3603	17.2651	-9.72894	12.80188	-1.88661
6	-3.29142	-9.60869	-2.29116	18.42751	11.3127
7	-1.7841	4.933651	3.598502	10.43335	-20.4523
8	-0.09689	14.03301	-13.3945	6.830987	-11.927
9	15.24945	-15.7512	-0.19841	-13.5585	-11.5459
10	1.969859	-20.9634	9.746213	5.196068	1.676824
11	-4.56851	-5.74147	-10.1836	11.39198	17.31849
12	-11.6491	17.20652	13.35578	-3.95105	8.316324

$V_j$ (hidden)	bias	$V_i$			
		X1	X2	X3	X4
13	-9.67309	-3.79847	15.54228	16.11669	-6.45769
14	4.372817	-8.74067	13.28916	-15.7689	-8.01236
15	-0.32624	-15.7709	0.234306	-8.14034	15.8361
16	-5.13387	-5.07051	-10.9368	0.61014	20.47908
17	-1.38759	-5.52878	10.91655	-19.4202	6.173985
18	4.82073	-15.6075	-11.1684	9.498056	-10.3749
19	-5.79817	-2.48571	17.50523	2.164759	-15.7423
20	-7.11005	-17.7882	5.290768	1.325232	14.80068

Sumber: Hasil olah data (2023)

Tabel 4.8 menunjukkan bahwa X1 sampai X4 adalah bobot awal dari *input layer* ke *hidden layer* dan bias awal *input layer* terhadap *hidden layer*, bobot dan bias tersebut dilakukan perhitungan model *backpropagation* dengan 1 iterasi.

**Tabel 4. 9** Bobot dan bias awal pada *Hidden Layer* Terhadap *Output Layer*

$W_j$	$W_k$	$W_j$	$W_k$
0	0.47844	11	0.137572
1	-0.01024	12	0.095076
2	0.806289	13	0.067134
3	-0.09667	14	0.148682
4	-0.2024	15	-0.25558
5	-0.27137	16	0.120506
6	0.114221	17	-0.45398
7	-0.14599	18	-0.09464
8	-0.16922	19	-0.08281
9	0.05708	20	0.841776
10	-0.08852		

Sumber: Data diolah 2023

Tabel 4.9 Menunjukkan bahwa bobot dan bias awal *hidden layer* terhadap *output layer*, dimana  $W_{0k}$  adalah bias awal dari *hidden layer* ke *output layer* sedangkan  $W_{1-20}$  adalah bobot awal *hidden layer* ke *output layer*, dari nilai-nilai tersebut akan dilakukan perhitungan model *backpropagation* dengan 1 iterasi.

b. Melakukan Pelatihan dengan Backpropagation

Melakukan *Training* dengan data *input* pelatihan dengan algoritma *backpropagation* merupakan algoritma dengan proses pembelajaran terawasi (*supervised learning*). Setelah didapatkannya inisialisasi bobot awal serta biasanya, maka akan dilakukan proses pelatihan (*training*) pada jaringan yang telah dirancang arsitektur dan parameter-parameter yang telah ditentukan menggunakan data pelatihan yang telah ditentukan persentasenya yaitu 80% dari total keseluruhan data. Ada tiga fase proses pelatihan untuk algoritma *backpropagation* yaitu umpan maju (*feedforward*), propagasi balik (*backpropagation*), dan modifikasi bobot.

**Fase 1: Umpan maju (*feedforward*)**

Pada tahap ini akan dicari galat atau *error* keluarannya dengan arah maju (*forward*). Tiap unit masukan  $x_i$  ( $i : 1, \dots, n$ ) akan menerima sinyal masukan  $x_i$  kemudian meneruskannya ke unit tersembunyi. Kemudian akan dihitung semua sinyal input yang sudah terboboti termasuk biasanya disetiap unit tersembunyi  $Z_j$  ( $j : 1, \dots, p$ ) menggunakan rumus pada persamaan 2.12. Hasil perhitungan dapat ditampilkan pada tabel berikut:

**Tabel 4. 10** Sinyal Input dari Input Layer ke Hidden Layer

$Z$	$Z_{net j}$	$Z$	$Z_{net j}$
1	-0.36951	11	-2.79649
2	-2.12972	12	-5.54377
3	8.922595	13	-5.955
4	8.616895	14	1.193384
5	-7.70142	15	-0.76217
6	0.638832	16	-2.64596
7	-2.27423	17	-3.05542
8	-0.33846	18	-1.9791
9	7.973456	19	-4.9239
10	0.774139	20	-6.35121

**Sumber:** Hasil olah data (2023)

Tabel 4.10 hasil dari sinyal *input layer* ke *hidden layer* setelah *hidden layer* menerima *sinyal input*, kemudian menggunakan fungsi aktivasi untuk menghitung



sinyal keluaranya seperti pada persamaan 2.13 dan diteruskan ke *output layer* dengan hasil sebagai berikut:

**Tabel 4. 11** Sinyal output di *hidden layer*

Z	Z <sub>j</sub>	Z	Z <sub>j</sub>
1	0.408658	11	0.057514
2	0.106241	12	0.003897
3	0.999867	13	0.002586
4	0.999819	14	0.767346
5	0.000452	15	0.318176
6	0.654489	16	0.066239
7	0.09328	17	0.044984
8	0.416184	18	0.121415
9	0.999656	19	0.007218
10	0.684416	20	0.001742

**Sumber:** Hasil olah data (2023)

*Sinyal output* di *hidden layer* pada tabel 4.11 akan berperan sebagai sinyal input di *output layer*. Sinyal input tersebut akan diteruskan ke *output layer* dengan bobot-bobot dan bias di *hidden layer* terhadap *output layer* pada tiap unit *output* menggunakan persamaan 2.14, dengan hasil perhitungan sebagai berikut:

**Tabel 4. 12** Sinyal output pada *hidden layer*

Y	Y <sub>net<sub>k</sub></sub>
1	0.266126

**Sumber:** Hasil olah data (2023)

Setelah *output layer* menerima sinyal input tabel 4.12 dari *hidden layer*, maka *sinyal input* tersebut diaktifkan menggunakan fungsi aktivasi seperti pada persamaan 2.14 dengan *output* sebagai berikut:

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}}$$

**Tabel 4. 13** Sinyal pada *output layer*

Y	Y <sub>k</sub>
1	0.566142

**Sumber:** Hasil olah data (2023)

Pada tabel 4.13 menunjukkan bahwa hasil *output* pada proses *feedforward*, dari hasil output tersebut digunakan dalam menghitung proses *backpropagation* untuk mencari nilai dari bobot dan bias di *hidden layer* terhadap *output layer*.

**Fase II: Backpropagation**

Tahap sebelumnya sudah didapatkan *output* jaringan dan dibandingkan dengan *output* dengan target yang di *input*. Dari hal tersebut didapatkan *error* untuk memperbaiki bobot dan bias secara mundur. Menghitung faktor  $\delta$  di unit *output*  $Y$ , hasil dari perhitungan dengan menggunakan Persamaan 2.15 hasil kesalahan  $\delta_k$  sebesar 0.566142.

Faktor kesalahan  $\delta k$  digunakan untuk memperbaiki bobot ( $W_{kj}$ ) dan bias ( $W_{0k}$ ) di lapisan *hidden layer*, perubahan bobot dengan *learning rate* ( $\alpha$ ) = 0.1 menggunakan persamaan 2.17. Hasil perhitungan adalah sebagai berikut:

$$\Delta w_{kj} = \alpha \delta_k z_j$$

**Tabel 4. 14** Koreksi Bobot dan bias di *Hidden Layer* Terhadap *Output Layer*

$\Delta W_j$	$\Delta W_k$	$\Delta W_j$	$\Delta W_k$
0	-0.0111683	11	-0.0006423
1	-0.004564	12	-0.000044
2	-0.0011865	13	-0.000029
3	-0.0111665	14	-0.0085697
4	-0.011166	15	-0.0035534
5	-0.000051	16	-0.0007398
6	-0.0073093	17	-0.0005024
7	-0.0010418	18	-0.001356
8	-0.0046479	19	-0.000081
9	-0.0111642	20	-0.000019
10	-0.0076436		

**Sumber:** Hasil olah data (2023)

Setiap unit tersembunyi  $Z_j$  menerima input delta bobot dan bias dari lapisan di atasnya (*output layer*) seperti pada tabel 4.14 diatas. Input delta bobot dan bias tersebut akan digunakan untuk mencari faktor kesalahan  $\delta_{net j}$  di setiap unit

tersembunyi seperti pada persamaan 2.19. Tabel 4.15 merupakan hasil perhitungan faktor kesalahan disetiap unit tersembunyi.

**Tabel 4. 15** Faktor Kesalahan di Unit tersembunyi

$Z_j$	$\delta_{net j}$	$Z_j$	$\delta_{net j}$
1	0.001144	11	-0.01536
2	-0.09005	12	-0.01062
3	0.010796	13	-0.0075
4	0.022604	14	-0.0166
5	0.030306	15	0.028543
6	-0.01276	16	-0.01346
7	0.016304	17	0.0507
8	0.018898	18	0.010569
9	-0.00637	19	0.009248
10	0.009885	20	-0.09401

**Sumber:** Hasil olah data (2023)

Inputan dari faktor kesalahan di *hidden layer* seperti Tabel 4.15 akan diaktifkan menggunakan fungsi aktivasi seperti persamaan 2.20 dengan hasil sebagai berikut.

**Tabel 4. 16** Aktivasi Kesalahan di Unit tersembunyi

$Z_j$	$\delta_j$	$Z_j$	$\delta_j$
1	0.00027648	11	-0.0008328
2	-0.0085503	12	-0.000041
3	0.0000014	13	-0.000019
4	0.0000041	14	-0.0029644
5	0.000014	15	0.00619218
6	-0.0028846	16	-0.0008324
7	0.00137899	17	0.0021781
8	0.00459175	18	0.00112748
9	-0.0000022	19	0.000066
10	0.00213517	20	-0.0001634

**Sumber:** Data diolah (2023)

Setelah didapatkannya faktor kesalahan  $\delta_j$  yang telah diaktifkan dengan fungsi aktivasi, maka faktor kesalahan tersebut akan digunakan untuk memperbaiki atau mengkoreksi bobot-bobot dan bias yang ada di lapisan bawahnya yaitu *input layer* terhadap *hidden layer* dengan perhitungan seperti pada persamaan 2.21 dan 2.22. Berikut koreksi bobot-bobot dan bias.

**Tabel 4. 17** Koreksi Bias dan bobot *Input Layer* terhadap *Hidden Layer*

$V_j$ (hidden)	bias	$V_i$			
		X1	X2	X3	X4
1	0.000028	0.000028	0.0000031	0.0000029	0.000003
2	-0.000855	-0.000096	-0.0000089	-0.0000085	-0.000123
3	0.00000014	0.00000002	0.00000001	0.00000002	0.00000002
4	0.0000004	0.00000004	0.00000004	0.00000006	0.0000001
5	0.0000014	0.0000002	0.0000002	0.0000003	0.0000002
6	-0.0002885	-0.0000043	-0.0000057	-0.0000059	-0.0000057
7	0.0001379	0.000027	0.000028	0.000024	0.000025
8	0.00045918	0.000094	0.0000581	0.000091	0.00008
9	-0.0000002	-0.00000004	-0.00000008	-0.00000008	-0.0000001
10	0.00021352	0.000043	0.000037	0.000038	0.000035
11	-0.000083	-0.000014	-0.000015	-0.000013	-0.000015
12	-0.0000041	-0.0000007	-0.000007	-0.0000007	-0.0000001
13	-0.0000019	-0.0000003	-0.0000003	-0.0000005	-0.0000003
14	-0.0002964	-0.000047	-0.000055	-0.000053	-0.000072
15	0.00061922	0.0001138	0.00011052	0.0001077	0.0001556
16	-0.000083	-0.000014	-0.0000015	-0.00002	-0.000021
17	0.00021781	0.000038	0.000053	0.000055	0.000054
18	0.00011275	0.000027	0.000028	0.000028	0.000024
19	0.0000066	0.0000017	0.0000012	0.0000016	0.000002
20	-0.000016	-0.0000045	-0.000004	-0.0000035	-0.000004

**Sumber:** Hasil olah data (2023)

Berdasarkan Tabel 4.17 diatas menunjukkan bahwa bias *input* terhadap *hidden layer*, selanjutnya dicari bobot dan bias akhir pada iterasi pertama.

### Fase III: Modifikasi bobot dan bias

Pada tahap ini dilakukan modifikasi bobot dan bias menggunakan koreksi bobot dan bias yang telah dilakukan pada tahap sebelumnya, sehingga akan didapatkan bobot baru untuk menghasilkan target yang sesuai. Pada tahap ini akan menghitung semua perubahan bobot dan bias di setiap lapisan *input* terhadap *hidden layer* ( $V_{ij}$ ,  $V_{oj}$ ) dan *hidden layer* terhadap *output layer* ( $W_{jk}$ ,  $W_{ok}$ ) menggunakan persamaan 2.23 dan 2.24. Proses pelatihan ini akan terus berjalan selama kondisi belum terpenuhi. Pelatihan akan berhenti ketika telah didapatkan error yang optimal, sehingga didapatkan bobot dan bias akhir untuk masing-masing lapisan, dengan satu kali langkah hasil perhitungannya adalah sebagai berikut:

Tabel 4. 18 Bias dan bobot Akhir pada *Input Layer* terhadap *Hidden Layer*

$V_j$ ( <i>hidden</i> )	<i>bias</i>	$V_i$			
		X1	X2	X3	X4
1	-0.0000002	-7.78358	7.092589	6.205311	20.40012
2	-0.0006894	2.768173	-16.7769	-11.2713	12.23519
3	-0.00000001	-14.3323	2.756734	-15.3136	10.8137
4	-0.00000008	-17.232	11.22701	-9.61179	-7.14124
5	-0.00000037	17.2651	-9.72894	12.80188	-1.88661
6	-0.000033	-9.60873	-2.29122	18.42745	11.31265
7	-0.00002	4.933678	3.598531	10.43338	-20.4522
8	-0.000078	14.0331	-13.3945	6.831077	-11.9269
9	-0.00000001	-15.7512	-0.19841	-13.5585	-11.5459
10	0.000019	-20.9634	9.746251	5.196106	1.676859
11	-0.000012	-5.74148	-10.1837	11.39197	17.31847
12	-0.00000039	17.20652	13.35578	-3.95105	8.316323
13	-0.00000013	-3.79847	15.54228	16.11669	-6.4577
14	-0.000044	-8.74072	13.28911	-15.7689	-8.01243
15	-0.0001583	-15.7708	0.234416	-8.14023	15.83626
16	-0.00001	-5.07053	-10.9368	0.61012	20.47906
17	-0.000099	-5.52874	10.9166	-19.4201	6.174039
18	-0.00001	-15.6075	-11.1684	9.498084	-10.3749

$V_j$ (hidden)	bias	$V_i$			
		X1	X2	X3	X4
19	0.00000054	-2.48571	17.50523	2.164761	-15.7423
20	-0.000014	-17.7882	5.290764	1.325229	14.80067

Sumber: Hasil olah data 2023

**Tabel 4. 19** Bobot dan bias akhir pada Input Layer terhadap Output Layer

$W_j$	$W_k$	$W_j$	$W_k$
0	0.467272	11	0.13693
1	-0.01481	12	0.095032
2	0.805103	13	0.067105
3	-0.10784	14	0.140112
4	-0.21357	15	-0.25913
5	-0.27137	16	0.119766
6	0.106912	17	-0.45448
7	-0.14703	18	-0.096
8	-0.17386	19	-0.08289
9	0.045916	20	0.841757
10	-0.09616		

Sumber: Hasil olah data (2023)

Pada Tabel 4.18 menunjukkan hasil dari bobot dan bias pada *input layer* ke *hidden layer* pada iterasi pertama. Tabel 4.19 menunjukkan hasil dari bobot dan bias akhir pada *hidden layer* ke *output layer* pada iterasi pertama.

Fase 1 sampai 3 dilakukan terus berulang sampai mencapai target minimum error atau telah mencapai batas maksimum iterasi yang telah ditetapkan pada saat inisialisasi awal. Selanjutnya karena dimensi data yang cukup besar maka proses pelatihan akan dilakukan dengan menggunakan program komputer (Matlab 2017) untuk membangun model *artificial neural network backpropagation*. Proses pelatihan dilakukan dengan *trial* dan *error* hingga diperoleh hasil 24 kali pelatihan, dimana percobaan pelatihan pada *input layer* sebanyak 4 dan hidden layer 6 (5,10,15,20, 25 dan 30) hingga membentuk jaringan yang optimal. Pada penelitian ini jumlah iterasi yang dibuat melalui program matlab yaitu 50.000 iterasi dengan perubahan bobot, bias dan output terlihat pada tabel 4.24 dan 4.25.

**Tabel 4. 20** Bias dan bobot Akhir pada *Input Layer* terhadap *Hidden Layer* iterasi ke- 50.000

$V_j$ (hidden)	bias	$V_i$			
		X1	X2	X3	X4
1	13.60004	-8.63679	-11.3163	0.510303	4.054026
2	-8.19572	2.135728	-8.85794	11.74327	0.125445
3	-2.672	4.073096	7.614688	-9.2412	-7.69219
4	1.10659	-1.60268	-9.37722	7.767359	8.318396
5	-8.06173	8.150359	8.285948	-8.97917	1.874397
6	0.834982	6.658301	-12.9985	0.702146	-1.67931
7	-0.41016	-4.83828	3.234378	-9.0848	10.52535
8	-5.65325	-4.09756	-0.2096	5.373672	13.38353
9	12.16244	-3.34741	-7.31685	-13.0431	-1.21002
10	-13.8647	8.002317	2.201118	7.623046	9.778069
11	-2.91027	6.31314	-9.77872	9.242472	1.119073
12	-10.2741	-2.44976	5.744059	13.03865	1.566697
13	-10.9143	4.398316	2.564423	13.0265	5.020794
14	5.214949	10.47422	-10.0427	0.596167	-2.20659
15	10.01561	6.721624	-9.78449	-6.13646	-6.44561
16	9.106985	5.20734	-9.02252	-10.5013	1.948414
17	1.503821	-7.53434	-9.98428	0.957516	8.460347
18	2.431772	-9.14386	-5.92212	6.811248	-7.8141
19	0.910393	3.344833	8.069045	6.426087	-9.95849
20	0.698896	-1.38533	3.118513	-14.3375	-2.18646

**Sumber:** Hasil olah data (2023)

**Tabel 4. 21** Bobot dan bias Akhir pada Input Layer terhadap Output Layer iterasi ke-50.000

$W_j$	$W_k$	$W_j$	$W_k$
0	-0.40741	11	-0.19218
1	-0.08048	12	0.74813
2	-0.71615	13	0.381352
3	0.321901	14	-0.81921
4	0.121461	15	0.796088
5	-0.14557	16	-0.02914
6	0.147435	17	0.296736
7	0.324382	18	-0.40872
8	-0.33391	19	0.537384
9	0.330142	20	-1.23886
10	-0.11541		

**Sumber:** Hasil olah data (2023)

Berdasarkan Tabel 4.20 dan Tabel 4.21 diatas menunjukkan bahwa bobot dan bias *input*, *hidden* dan *output* pada iterasi terakhir yang memiliki nilai error MSE terkecil 0.0001575, terdapat pada 4 inputan layer, 20 hidden layer dan 1 output layer. Selanjutnya dilakukan pembentukan model terbaik dari 4 input, 20 hidden dan 1 output.

c. Penentuan Model Jaringan

Berbagai model telah dibangun dengan harga emas, jumlah *input* dan jumlah *hidden neuron* yang bervariasi, maka jumlah model prediksi yang dibangun pada penelitian ini sebanyak 24 model. Model terbaik yang didapatkan pada penelitian ini adalah jumlah *input* 4 minggu sebelum minggu penjualan dengan 20 neuron pada *hidden layer*, dapat ditulis berdasarkan persamaan 2.10 sebagai berikut :

$$y_k = \left( w_{ok} + \sum_{j=1}^{20} w_{jk} \cdot \left( \frac{1}{1 + e^{-[v_{0j} + \sum_{i=1}^4 v_{ij}x_i]}} \right) \right)$$



$$y_k = \left( -0.40741 + \sum_{j=1}^{20} w_{jk} \cdot \left( \frac{1}{1 + e^{-[13.60004 + \sum_{i=1}^4 v_{ij}x_i]}} \right) \right)$$

$$y_k = \left( -0.40741 \right.$$

$$\quad \left. + ((-08048 + (-071615) + (0.321901) + \dots + (0.537384) \right.$$

$$\quad \left. + (-1.23886) \cdot \left( \frac{1}{1 + e^{[2.45848 - \sum_{i=1}^4 v_{ij}x_i]}} \right) \right)$$

$$y_k = (-0.07462) \left( \frac{1}{1 + e^{[2.45848 - \sum_{i=1}^4 v_{ij}x_i]}} \right)$$

### 4.3 Perbandingan Hasil Ramalan

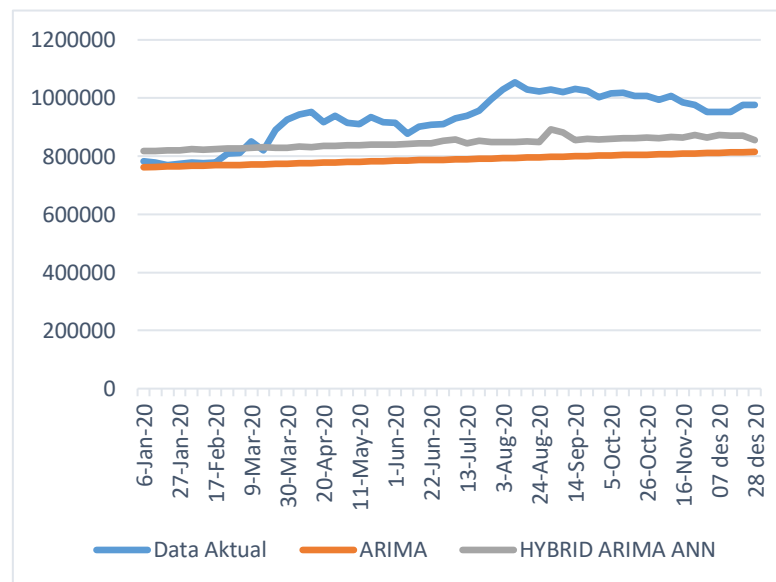
Peramalan data harga emas di Indonesia yang telah dilakukan menggunakan metode ARIMA, dan *hybrid* ARIMA Jaringan saraf tiruan masing-masing memiliki hasil peramalan dengan nilai MAPE berbeda. Hasil perbandingan data aktual harga emas Tahun 2020 dengan hasil peramalan ARIMA dan *hybrid* ARIMA Jaringan saraf tiruan pada Tahun 2020 ditunjukkan pada Tabel 4.22 berikut:

**Tabel 4. 22** Data Aktual dan Hasil Peramalan Harga Emas di Indonesia Periode Januari 2020-Desember 2020 dengan Metode ARIMA dan *hybrid* ARIMA Jaringan saraf tiruan

Harga Emas			
Tanggal	Aktual	Peramalan ARIMA	Peramalan <i>Hybrid</i> ARIMA Jaringan saraf tiruan
6-Jan-20	783000	761670	818700
13-Jan-20	778000	762935	817577
20-Jan-20	769000	763942	819601
27-Jan-20	774000	764983	820511
3-Feb-20	779000	766019	823578
⋮	⋮	⋮	⋮
21 des 20	976000	813696	870834
28 des 20	977000	814732	856002
MAPE		14.7%	10%
Korelasi		0.800665	0.821496

**Sumber:** Hasil olah data (2023)

Berdasarkan Tabel 4.23 menunjukkan perbandingan dari kedua model ARIMA dan *Hybrid* ARIMA Jaringan saraf tiruan dengan melihat akurasi nilai *Mean Absolute Persentasde Error* (MAPE). Model ARIMA dengan nilai MAPE 14.7%, sedangkan model *Hybrid* ARIMA Jaringan saraf tiruan memiliki nilai error MAPE sebesar 10% dan korelasi antara data aktual dengan hasil peramalan *Hybrid* ARIMA Jaringan saraf tiruan lebih besar dibandingkan korelasi antara data aktual dengan hasil peramalan ARIMA. Dengan demikian model *Hybrid* ARIMA Jaringan saraf tiruan adalah model terbaik dalam proses peramalan harga emas dari tanggal Januari 2016 sampai dengan Desember 2019. Grafik perbandingan data aktual harga emas dengan hasil peramalan ARIMA, dan *hybrid* ARIMA Jaringan saraf tiruan pada Tahun 2020 ditunjukkan pada Gambar 4.9 berikut.



**Gambar 4. 6** Grafik Perbandingan Data Aktual dengan Hasil Peramalan ARIMA dan *Hybrid* ARIMA-JST

Grafik perbandingan data aktual dan hasil peramalan menggunakan kedua metode menunjukkan bahwa garis plot hasil peramalan menggunakan metode ARIMA bergerak linear atau konstan, dan garis plot hasil peramalan menggunakan metode *hybrid* ARIMA Jaringan saraf tiruan di titik awal berada diatas garis plot data aktual kemudian turun di bawah garis plot data aktual kemudian naik turun mendekati garis plot data aktual sehingga dapat dikatakan bahwa peramalan menggunakan metode *hybrid* ARIMA Jaringan saraf tiruan lebih baik dibandingkan dengan metode ARIMA.

## BAB V PENUTUP

### 5.1 Kesimpulan

Pemilihan metode yang optimal untuk peramalan didasarkan pada besarnya nilai MAPE (Mean Absolute Percentage Error ) yang diperoleh serta garis plot dari grafik perbandingan data aktual dengan hasil peramalan dari masing-masing metode. Hasil analisis dengan metode Autoregressive Integrated Moving Average (ARIMA), diperoleh model terbaik yaitu ARIMA (1,1,0) dengan nilai MAPE sebesar 14.7%. Pada peramalan dengan metode *Hybrid* ARIMA Jaringan saraf tiruan diperoleh arsitektur jaringan terbaik yaitu 4 input, 20 *hidden layer* dan 1 *output* dengan nilai MAPE sebesar 10%. Berdasarkan hasil peramalan dengan kedua metode tersebut, metode *Hybrid* ARIMA Jaringan saraf tiruan memiliki nilai MAPE lebih kecil dibandingkan model ARIMA. Sehingga metode yang optimal digunakan dalam peramalan harga emas mingguan di Indonesia adalah metode *Hybrid* ARIMA Jaringan saraf tiruan.

### 5.2 Saran

Dalam pengembangan metode ini masih memiliki keterbatasan. Oleh karena itu, perlu dikembangkan beberapa hal diantaranya melakukan uji coba komposisi data dalam proses training dan testing metode *Hybrid* ARIMA Jaringan saraf tiruan. Memperbanyak variasi parameter jaringan seperti hidden layer. Selain itu, bisa melakukan metode *hybrid* dengan metode lain seperti *hybrid* ARIMA-RBF.

**DAFTAR PUSTAKA**

- Anggriningrum, D. P., Hendikawati, P. & Abidin, Z. (2013). *Perbandingan Prediksi Harga Saham dengan Menggunakan Jaringan Syaraf Tiruan Backpropagation dan ARIMA*. *UNNES Journal of Mathematics*, 2(2): 154 – 159.
- Box, G. P., & Jenkins, G. (1976). *Time Series Analysis, Forecasting and Control*, St. San Fransisco: Holden-Day, Inc.500.
- Faruk, D. Ö. (2010). *A Hybrid Neural Network and ARIMA Model for Water Quality Time Series Prediction*, Aydın, Turkey: Eng App Intelligence.
- Halim, Siana, dan Adrian Michael Wibisono. (2000). *Penerapan Jaringan Saraf Tiruan untuk Peramalan*. *Dalam Jurnal Teknik Industri Vol.2 No.2 Hal.* 106-113.
- Khashei, M. & Bijari. (2011). *Which Methodology is Better for Combining Linear and Nonlinear Models for Time Series Forecasting?*. *Journal of Industrial and Systems Engineering*, 4(4): 265 – 285.
- Kusumadewi, S. & Hartati, S. (2010). *Integrasi Sistem Fuzzy dan Jaringan Syaraf*, Yogyakarta: Graha Ilmu.
- Muhammad & Mahater. (2010). *Perbandingan Jaringan Syaraf Tiruan Backpropagation dan Metode Arima Sebagai Metode Peramalan Kurs Rupiah Terhadap Dollar Amerika Serikat*, [Skripsi]. Medan, Indonesia: Universitas Sumatera Utara.
- Siang & Jong Jek. (2005). *Jaringan Syaraf Tiruan dan Pemogramannya Menggunakan Matlab*, Yogyakarta : Andi Offset.
- Wang, X. & Meng, M. (2012). *A Hybrid Neural Network and ARIMA Model for Energy Consumption Forecasting*. *Journal of Computers*, 7(5): 1184 – 1190.
- Wibowo, A.P.S. (2007). *Perbandingan Jaringan Syaraf Tiruan Backpropagation dan Metode Deret Berkala Box-Jenkins (ARIMA) sebagai Metode*

*Peramalan Curah Hujan*, Skripsi. Semarang: FMIPA Universitas Negeri Semarang.

Zhang, G. P. (2003). *Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model*. *Neurocomputing*, 50.159-175, 160.

## LAMPIRAN

**Lampiran 1** Data Harga Emas di Indonesia 4 Januari Tahun 2016 – 30 Desember Tahun 2019

Tanggal	Emas Antam	Tanggal	Emas Antam	Tanggal	Emas Antam
4-Jan-16	545000	31-Oct-16	601000	28-Aug-17	605000
11-Jan-16	548000	7-Nov-16	605000	4-Sep-17	618000
18-Jan-16	545000	14-Nov-16	595000	11-Sep-17	617000
25-Jan-16	544000	21-Nov-16	590000	18-Sep-17	609000
1-Feb-16	548000	28-Nov-16	590000	25-Sep-17	606000
8-Feb-16	558000	5-Dec-16	589000	2-Oct-17	606000
15-Feb-16	574000	12-Dec-16	585000	9-Oct-17	613472
22-Feb-16	574000	19-Dec-16	586000	16-Oct-17	629616
29-Feb-16	564000	26-Dec-16	584000	23-Oct-17	623562
7-Mar-16	573000	2-Jan-17	588000	30-Oct-17	626589
14-Mar-16	565000	9-Jan-17	582000	6-Nov-17	623562
21-Mar-16	567000	16-Jan-17	589000	13-Nov-17	621544
28-Mar-16	560000	23-Jan-17	593000	20-Nov-17	628607
4-Apr-16	562000	30-Jan-17	583000	27-Nov-17	624000
11-Apr-16	569000	6-Feb-17	586000	4-Dec-17	618000
18-Apr-16	566000	13-Feb-17	589000	11-Dec-17	613000
25-Apr-16	565000	20-Feb-17	591000	18-Dec-17	617000
2-May-16	588000	27-Feb-17	597000	25-Dec-17	623000
9-May-16	587000	6-Mar-17	589000	1-Jan-18	653000
16-May-16	586000	13-Mar-17	580000	8-Jan-18	633000
23-May-16	586000	20-Mar-17	584000	15-Jan-18	635000
30-May-16	575000	27-Mar-17	592000	22-Jan-18	628000
6-Jun-16	586000	3-Apr-17	589000	29-Jan-18	636000
13-Jun-16	589000	10-Apr-17	590000	5-Feb-18	633000
20-Jun-16	594000	17-Apr-17	600000	12-Feb-18	636000

27-Jun-16	610000	24-Apr-17	598000	19-Feb-18	643000
4-Jul-16	599000	1-May-17	589000	26-Feb-18	640000
11-Jul-16	618000	8-May-17	581000	5-Mar-18	643000
18-Jul-16	602000	15-May-17	579000	12-Mar-18	646000
25-Jul-16	600000	22-May-17	588000	19-Mar-18	641000
1-Aug-16	609000	29-May-17	590000	26-Mar-18	657000
8-Aug-16	606000	5-Jun-17	595000	2-Apr-18	648000
15-Aug-16	607000	12-Jun-17	592000	9-Apr-18	650000
22-Aug-16	609000	19-Jun-17	589000	16-Apr-18	656000
29-Aug-16	606000	26-Jun-17	587000	23-Apr-18	653000
5-Sep-16	607000	3-Jul-17	585000	30-Apr-18	653000
12-Sep-16	604000	10-Jul-17	579000	7-May-18	655000
19-Sep-16	601000	17-Jul-17	587000	14-May-18	655000
26-Sep-16	607000	24-Jul-17	594000	21-May-18	652000
3-Oct-16	601000	31-Jul-17	598000	28-May-18	656000
10-Oct-16	590000	7-Aug-17	593000	4-Jun-18	648000
17-Oct-16	600000	14-Aug-17	601000	11-Jun-18	652000
24-Oct-16	601000	21-Aug-17	602000	18-Jun-18	652000
	Emas		Emas		
Tanggal	Antam	Tanggal	Antam		
25-Jun-18	647000	6-May-19	664000		
2-Jul-18	649000	13-May-19	664500		
9-Jul-18	651000	20-May-19	663000		
16-Jul-18	645000	27-May-19	665000		
23-Jul-18	652000	3-Jun-19	665000		
30-Jul-18	651000	10-Jun-19	681000		
6-Aug-18	654000	17-Jun-19	679000		
13-Aug-18	650000	24-Jun-19	702500		
20-Aug-18	648000	1-Jul-19	699000		
27-Aug-18	654000	8-Jul-19	700000		
3-Sep-18	654000	15-Jul-19	706000		

10-Sep-18	662000	22-Jul-19	704000		
17-Sep-18	664000	29-Jul-19	705000		
24-Sep-18	666000	5-Aug-19	724000		
1-Oct-18	666000	12-Aug-19	749000		
8-Oct-18	679000	19-Aug-19	757000		
15-Oct-18	681000	26-Aug-19	774000		
22-Oct-18	684000	2-Sep-19	768000		
29-Oct-18	686000	9-Sep-19	758000		
5-Nov-18	682000	16-Sep-19	753000		
12-Nov-18	662000	23-Sep-19	762000		
19-Nov-18	665000	30-Sep-19	761000		
26-Nov-18	652000	7-Oct-19	761000		
3-Dec-18	646000	14-Oct-19	754000		
10-Dec-18	660000	21-Oct-19	756000		
17-Dec-18	655000	28-Oct-19	757000		
24-Dec-18	657000	4-Nov-19	763000		
31-Dec-18	667000	11-Nov-19	742000		
7-Jan-19	664000	18-Nov-19	748000		
14-Jan-19	660000	25-Nov-19	747000		
21-Jan-19	661000	2-Dec-19	747000		
28-Jan-19	666000	9-Dec-19	744000		
4-Feb-19	666000	16-Dec-19	749000		
11-Feb-19	666000	23-Dec-19	752000		
18-Feb-19	671000	30-Dec-19	762000		
25-Feb-19	674000				
4-Mar-19	656500				
11-Mar-19	663500				
18-Mar-19	669500				
25-Mar-19	666000				
1-Apr-19	660000				
8-Apr-19	660500				



15-Apr-19	660000				
22-Apr-19	656000				
29-Apr-19	664000				

## Lampiran 2 Estimasi Parameter Model ARIMA

- ARIMA 1,1,0

### Final Estimates of Parameters

Type	Coef	SE Coef	T	P
AR 1	0.2020	0.0906	2.23	0.028
Constant	31.86	18.48	1.72	0.087

Differencing: 1 regular difference

Number of observations: Original series 120, after differencing 119

Residuals: SS = 4752486 (backforecasts excluded)  
MS = 40620 DF = 117

### Modified Box-Pierce (Ljung-Box) Chi-Square statistic

Lag	12	24	36	48
Chi-Square	6.0	21.1	31.7	36.0
DF	10	22	34	46
P-Value	0.812	0.516	0.579	0.855

- ARIMA 1,1,1

### Final Estimates of Parameters

Type	Coef	SE Coef	T	P
AR 1	-0.6635	0.1611	-4.12	0.000
MA 1	-0.8440	0.1154	-7.31	0.000
Constant	66.61	33.96	1.96	0.052

Differencing: 1 regular difference

Number of observations: Original series 120, after differencing 119

Residuals: SS = 4681538 (backforecasts excluded)  
MS = 40358 DF = 116

### Modified Box-Pierce (Ljung-Box) Chi-Square statistic

Lag	12	24	36	48
Chi-Square	4.7	16.9	30.1	34.4
DF	9	21	33	45
P-Value	0.857	0.715	0.613	0.874

**Lampiran 2** Estimasi Parameter Model ARIMA (Lanjutan)

- ARIMA 0,1,1

## Final Estimates of Parameters

Type		Coef	SE Coef	T	P
MA	1	-0.2326	0.0899	-2.59	0.011
Constant		40.05	22.71	1.76	0.080

Differencing: 1 regular difference

Number of observations: Original series 120, after differencing 119

Residuals: SS = 4725801 (backforecasts excluded)

MS = 40391 DF = 117

## Modified Box-Pierce (Ljung-Box) Chi-Square statistic

Lag	12	24	36	48
Chi-Square	5.2	19.7	30.2	34.9
DF	10	22	34	46
P-Value	0.875	0.602	0.654	0.883

### Lampiran 3 Uji Unit Root Test ADF

- Uji ADF sebelum *differencing*

Null Hypothesis: SERIES03 has a unit root  
 Exogenous: Constant  
 Lag Length: 0 (Automatic - based on SIC, maxlag=14)

	t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic	-0.388165	0.9076
Test critical values:		
1% level	-3.461783	
5% level	-2.875262	
10% level	-2.574161	

\*MacKinnon (1996) one-sided p-values.

- Uji ADF setelah *differencing*

Null Hypothesis: D(SERIES03) has a unit root  
 Exogenous: Constant  
 Lag Length: 0 (Automatic - based on SIC, maxlag=14)

	t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic	-16.58439	0.0000
Test critical values:		
1% level	-3.461938	
5% level	-2.875330	
10% level	-2.574198	

\*MacKinnon (1996) one-sided p-values.

**Lampiran 4** Data Peramalan Hasil Residual ARIMA (1,1,0) untuk Data Input hybrid ARIMA-Jaringan Saraf Tiruan

Tanggal	Residual Arima	Tanggal	Residual Arima
4-Jan-16	0.46833	10-Oct-16	0.54988
11-Jan-16	0.38034	17-Oct-16	0.45871
18-Jan-16	0.39751	24-Oct-16	0.42232
25-Jan-16	0.47983	31-Oct-16	0.48215
1-Feb-16	0.58463	7-Nov-16	0.2739
8-Feb-16	0.69176	14-Nov-16	0.31915
15-Feb-16	0.45707	21-Nov-16	0.40841
22-Feb-16	0.26463	28-Nov-16	0.40446
29-Feb-16	0.53666	5-Dec-16	0.35554
7-Mar-16	0.31656	12-Dec-16	0.42627
14-Mar-16	0.43254	19-Dec-16	0.39124
21-Mar-16	0.31588	26-Dec-16	0.47751
28-Mar-16	0.43485	2-Jan-17	0.33605
4-Apr-16	0.53339	9-Jan-17	0.51485
11-Apr-16	0.38961	16-Jan-17	0.49837
18-Apr-16	0.39751	23-Jan-17	0.2739
25-Apr-16	0.77502	30-Jan-17	0.44344
2-May-16	0.45776	6-Feb-17	0.47356
9-May-16	0.40215	13-Feb-17	0.45802
16-May-16	0.41768	20-Feb-17	0.51785
23-May-16	0.2491	27-Feb-17	0.30961
30-May-16	0.56541	6-Mar-17	0.26163
6-Jun-16	0.4921	13-Mar-17	0.46129
13-Jun-16	0.50463	20-Mar-17	0.55356
20-Jun-16	0.68017	27-Mar-17	0.39193
27-Jun-16	0.28617	3-Apr-17	0.42859
4-Jul-16	0.68971	10-Apr-17	0.57768
11-Jul-16	0.21544	17-Apr-17	0.4121
18-Jul-16	0.35185	24-Apr-17	0.27554
25-Jul-16	0.55519	1-May-17	0.27485
1-Aug-16	0.39424	8-May-17	0.37039
8-Aug-16	0.42859	15-May-17	0.55519
15-Aug-16	0.45339	22-May-17	0.47193
22-Aug-16	0.37802	29-May-17	0.50232
29-Aug-16	0.42859	5-Jun-17	0.38498
5-Sep-16	0.37571	12-Jun-17	0.36644
12-Sep-16	0.36644	19-Jun-17	0.38198
19-Sep-16	0.50627	26-Jun-17	0.38429
26-Sep-16	0.34068	3-Jul-17	0.32215
3-Oct-16	0.23519	10-Jul-17	0.53039

Tanggal	Residual Arima	Tanggal	Residual Arima
17-Jul-17	0.54729	23-Apr-18	0.41305
24-Jul-17	0.49837	30-Apr-18	0.45107
31-Jul-17	0.35159	7-May-18	0.42463
7-Aug-17	0.53271	14-May-18	0.37339
14-Aug-17	0.45407	21-May-18	0.47519
21-Aug-17	0.46893	28-May-18	0.30498
28-Aug-17	0.62893	4-Jun-18	0.46361
4-Sep-17	0.43459	11-Jun-18	0.42927
11-Sep-17	0.29339	18-Jun-18	0.34232
18-Sep-17	0.35485	25-Jun-18	0.43949
25-Sep-17	0.41305	2-Jul-18	0.45571
2-Oct-17	0.53609	9-Jul-18	0.33141
9-Oct-17	0.68814	16-Jul-18	0.51485
16-Oct-17	0.36335	23-Jul-18	0.42068
23-Oct-17	0.453	30-Jul-18	0.46429
30-Oct-17	0.37998	6-Aug-18	0.36481
6-Nov-17	0.38163	13-Aug-18	0.37966
13-Nov-17	0.52506	20-Aug-18	0.50859
20-Nov-17	0.36479	27-Aug-18	0.4339
27-Nov-17	0.31611	3-Sep-18	0.54429
4-Dec-17	0.32841	10-Sep-18	0.46961
11-Dec-17	0.47056	17-Sep-18	0.45571
18-Dec-17	0.52249	24-Sep-18	0.42463
25-Dec-17	0.9	1-Oct-18	0.62198
1-Jan-18	0.17878	8-Oct-18	0.4812
8-Jan-18	0.40473	15-Oct-18	0.47124
15-Jan-18	0.31588	22-Oct-18	0.45802
22-Jan-18	0.52807	29-Oct-18	0.36249
29-Jan-18	0.39193	5-Nov-18	0.1
5-Feb-18	0.45966	12-Nov-18	0.42027
12-Feb-18	0.53571	19-Nov-18	0.22498
19-Feb-18	0.38961	26-Nov-18	0.29666
26-Feb-18	0.45966	3-Dec-18	0.62361
5-Mar-18	0.47356	10-Dec-18	0.37476
12-Mar-18	0.34927	17-Dec-18	0.43949
19-Mar-18	0.657	24-Dec-18	0.58
26-Mar-18	0.31725	31-Dec-18	0.39656
2-Apr-18	0.43022	7-Jan-19	0.3509
9-Apr-18	0.51785	14-Jan-19	0.42627
16-Apr-18	0.38729	21-Jan-19	0.5

Tanggal	Residual Arima	Tanggal	Residual Arima
28-Jan-19	0.43159	15-Jul-19	0.40283
4-Feb-19	0.42	22-Jul-19	0.4309
11-Feb-19	0.49768	29-Jul-19	0.71751
18-Feb-19	0.4782	5-Aug-19	0.85244
25-Feb-19	0.15506	12-Aug-19	0.60222
4-Mar-19	0.48821	19-Aug-19	0.70266
11-Mar-19	0.52944	26-Aug-19	0.36617
18-Mar-19	0.37952	2-Sep-19	0.25073
25-Mar-19	0.31867	9-Sep-19	0.31915
1-Apr-19	0.41387	16-Sep-19	0.54824
8-Apr-19	0.41339	23-Sep-19	0.42532
15-Apr-19	0.3567	30-Sep-19	0.41768
22-Apr-19	0.53502	7-Oct-19	0.31124
29-Apr-19	0.43854	14-Oct-19	0.43485
6-May-19	0.42777	21-Oct-19	0.44017
13-May-19	0.39785	28-Oct-19	0.51554
20-May-19	0.4476	4-Nov-19	0.10763
27-May-19	0.42463	11-Nov-19	0.46456
3-Jun-19	0.66859	18-Nov-19	0.41837
10-Jun-19	0.426	25-Nov-19	0.41768
17-Jun-19	0.78048	2-Dec-19	0.37339
24-Jun-19	0.42007	9-Dec-19	0.49073
1-Jul-19	0.42743	16-Dec-19	0.4782
8-Jul-19	0.51554	23 Dec-19	0.58232
		30-Dec-19	0.45674