



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - IS184853**

**PENERAPAN METODE HYBRID  
AUTOREGRESSIVE INTEGRATED MOVING  
AVERAGE-RADIAL BASIS FUNCTION NEURAL  
NETWORK (ARIMA-RBFNN) UNTUK PERAMALAN  
HARGA ETHEREUM**

***APPLICATION OF HYBRID AUTOREGRESSIVE  
INTEGRATED MOVING AVERAGE-RADIAL BASIS  
FUNCTION NEURAL NETWORK (ARIMA-RBFNN)  
FOR FORECASTING ETHEREUM PRICE***

**MUHAMMAD IKRAM FARISI**  
0521 16 4000 0091

Dosen Pembimbing  
Edwin Riksakomara, S.Kom., M.T.

DEPARTEMEN SISTEM INFORMASI  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020





**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - IS184853**

**PENERAPAN METODE HYBRID  
AUTOREGRESSIVE INTEGRATED MOVING  
AVERAGE-RADIAL BASIS FUNCTION NEURAL  
NETWORK (ARIMA-RBFNN) UNTUK  
PERAMALAN HARGA ETHEREUM**

**MUHAMMAD IKRAM FARISI**  
0521 16 4000 0091

Dosen Pembimbing  
Edwin Riksakomara, S.Kom., M.T.

DEPARTEMEN SISTEM INFORMASI  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020

*“Halaman ini sengaja dikosongkan”*



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**UNDERGRADUATE THESIS - IS184853**

***APPLICATION OF HYBRID  
AUTOREGRESSIVE INTEGRATED MOVING  
AVERAGE-RADIAL BASIS FUNCTION NEURAL  
NETWORK (ARIMA-RBFNN) FOR  
FORECASTING ETHEREUM PRICE***

**MUHAMMAD IKRAM FARISI**  
0521 16 4000 0091

Supervisor  
Edwin Riksakomara, S.Kom., M.T.

DEPARTEMEN SISTEM INFORMASI  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020

*“Halaman ini sengaja dikosongkan”*

**LEMBAR PENGESAHAN****PENERAPAN METODE HYBRID AUTOREGRESSIVE  
INTEGRATED MOVING AVERAGE-RADIAL BASIS  
FUNCTION NEURAL NETWORK (ARIMA-RBFNN)  
UNTUK PERAMALAN HARGA ETHEREUM****TUGAS AKHIR**

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer (S.Kom)

pada

Departemen Sistem Informasi  
Fakultas Teknologi Elektro dan Informatika Cerdas  
(ELECTICS)

Institut Teknologi Sepuluh Nopember

Oleh:

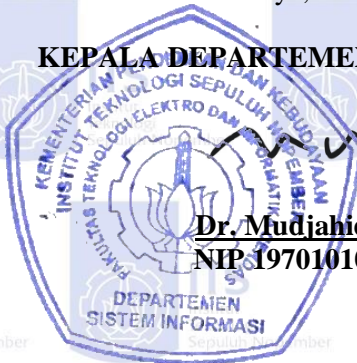
**MUHAMMAD IKRAM FARISI**

NRP. 0521164000091

Surabaya, 24 Agustus 2020

**KEPALA DEPARTEMEN SISTEM INFORMASI**

**Dr. Mudjahidin ST., MT.**  
**NIP.197010102003121001**



*“Halaman ini sengaja dikosongkan”*



## LEMBAR PERSETUJUAN

### **PENERAPAN METODE HYBRID AUTOREGRESSIVE INTEGRATED MOVING AVERAGE-RADIAL BASIS FUNCTION NEURAL NETWORK (ARIMA-RBFNN) UNTUK PERAMALAN HARGA ETHEREUM**

#### **TUGAS AKHIR**

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Departemen Sistem Informasi  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Oleh:

**MUHAMMAD IKRAM FARISI**

NRP. 0521164000091

Disetujui Tim Penguji : Tanggal Ujian : 22 Juli 2020  
Periode Wisuda : September 2020

**Edwin Riksakomara, SKom, MT**

(Pembimbing I)

**Ahmad Mukhlason, SKom, MSc, PhD**

(Penguji I)

**Faizal Mahananto, SKom, MEng, PhD**

(Penguji II)

*“Halaman ini sengaja dikosongkan”*

**PENERAPAN METODE HYBRID AUTOREGRESSIVE  
INTEGRATED MOVING AVERAGE-RADIAL BASIS  
FUNCTION NEURAL NETWORK (ARIMA-RBFNN)  
UNTUK PERAMALAN HARGA ETHEREUM**

**Nama Mahasiswa** : Muhammad Ikram Farisi  
**NRP** : 0521 16 4000 0091  
**Departemen** : Sistem Informasi FTEIC-ITS  
**Pembimbing I** : Edwin Riksakomara, S.Kom, M.T  
**Pembimbing II** : -

**ABSTRAK**

*Ethereum merupakan salah satu mata uang kripto yang pertumbuhan pasarnya sedang meningkat. Perkembangan Ethereum yang pesat ini disebabkan oleh faktor kemudahan dalam melakukan transaksi untuk keperluan investasi, selain itu keuntungan yang didapatkan dari hasil trading atau investasi Ethereum cukup tinggi. Perkembangan Ethereum di Indonesia sendiri cukup cepat, hal ini didasari oleh banyaknya permintaan pasar pada platform penyedia jasa transaksi cryptocurrency. Peningkatan permintaan pasar ini disebabkan sudah mulai banyaknya minat masyarakat Indonesia terhadap investasi cryptocurrency, khususnya Ethereum. Selain hal tersebut regulasi tentang perdagangan cryptocurrency juga sudah mendapatkan perizinan dari pemerintah Indonesia, hal inilah yang juga mendorong minat masyarakat untuk melakukan investasi atau trading cryptocurrency. Selain tingkat keuntungannya yang tergolong tinggi, investasi Ethereum juga memiliki resiko yang sangat tinggi. Resiko ini disebabkan karena sifat harga Ethereum ataupun cryptocurrency lain yang fluktuatif atau tidak stabil. Hal ini disebabkan karena sifat harga Ethereum yang murni dipengaruhi oleh penawaran(supply) dan permintaan(demand)*

yang terjadi di pasar kripto. Selain itu tidak ada Lembaga yang menjadi pengawas terhadap ketidakpastian harga ini. Ketidakpastian harga ini yang menjadi masalah utama bagi masyarakat yang melakukan kegiatan investasi ataupun trading. Untuk menghadapi ketidakpastian harga ini maka diperlukan suatu informasi yang dapat membantu investor dalam pengambilan keputusan investasi ataupun transaksinya. Salah satu cara untuk mendapatkan informasi ini adalah dengan melakukan metode Peramalan.

Penelitian tugas akhir ini menggunakan menggunakan metode ARIMA, RBFNN dan hybrid ARIMA-RBFNN untuk menghasilkan hasil peramalan harga Ethereum. Hasil pengukuran dengan indikator MAPE untuk mengevaluasi model peramalan, menunjukkan bahwa ketiga metode yang digunakan yaitu ARIMA, RBFNN, dan Hybrid ARIMA-RBFNN memiliki tingkat akurasi yang sangat baik, karena memiliki nilai MAPE dibawah 10%. Jika dibandingkan dengan metode ARIMA dan RBFNN, metode Hybrid ARIMA-RBFNN merupakan metode yang memiliki nilai akurasi terbaik berdasarkan pada data pengujian harga Ethereum karena memiliki nilai MAPE yang lebih kecil yaitu 2,80%, sedangkan nilai MAPE pada metode ARIMA adalah 2,82% dan RBFNN adalah 3,25%.

**Kata Kunci :** Peramalan, Ethereum, ARIMA, RBF

**APPLICATION OF HYBRID AUTOREGRESSIVE  
INTEGRATED MOVING AVERAGE-RADIAL BASIS  
FUNCTION NEURAL NETWORK (ARIMA-RBFNN) FOR  
FORECASTING ETHEREUM PRICE**

**Name** : Muhammad Ikram Farisi  
**NRP** : 0521 16 4000 0091  
**Department** : Information System FTEIC-ITS  
**Supervisor I** : Edwin Riksakomara, S.Kom, M.T.  
**Supervisor II** : -

**ABSTRACT**

*Ethereum is one of the cryptocurrencies whose market growth is increasing. The rapid development of Ethereum is caused by the convenience factor in conducting transactions for investment purposes, in addition to the profit gained from the results of trading or investment of Ethereum is quite high. The development of Ethereum in Indonesia is quite fast, it is based on the many demands of the platform market provider of trading investments cryptocurrency. The increasing demand for this market is due to the many interests of Indonesian people to invest in cryptocurrency, especially Ethereum. In addition to the regulation on cryptocurrency trading has also obtained licensing from the Indonesian government, this is what also encourages the interest of people to invest or trade cryptocurrencies. In addition to the high level of profit, Ethereum Investment also has a very high risk. This risk is due to the nature of Ethereum's price or other cryptocurrencies that fluctuate or unstable. This is due to the nature of Ethereum price which is purely influenced by the supply and demand that occurs in the crypto market. Also, no institution is the supervisor against the uncertainty of this price. This price uncertainty is the main problem for people who are investing or trading. To deal with it, information is needed to help investors in making investment or transaction decisions. One way to get this information is to do the forecasting method.*

*This thesis research uses ARIMA, RBFNN, and hybrid ARIMA-RBFNN methods to produce Ethereum price forecasting results. The measurement results with MAPE indicator to evaluate the forecasting model, show that the three methods used, ARIMA, RBFNN, and Hybrid ARIMA-RBFNN have an excellent accuracy rate, because it has a MAPE value below 10%. When compared with ARIMA and RBFNN method, the hybrid method of ARIMA-RBFNN is a method that has the best accuracy value based on Ethereum price test data because it has a smaller MAPE value of 2,80%, while the MAPE value of the ARIMA method is 2,82% and RBFNN method is 3,25%.*

**Keywords:** *Forecasting, Ethereum, RBF, ARIMA*

## SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertandatangan di bawah ini:

Nama : Muhammad Ikram Farisi  
NRP : 05211640000091  
Tempat/Tanggal lahir : Situbondo / 29 Juli 1998  
Fakultas/Departemen : FTEIC / Sistem Informasi  
Nomor Telp/Hp/email : farisikram21@gmail.com

Dengan ini menyatakan dengan sesungguhnya bahwa penelitian/makalah/tugas akhir saya yang berjudul

### **PENERAPAN METODE HYBRID AUTOREGRESSIVE INTEGRATED MOVING AVERAGE-RADIAL BASIS FUNCTION NEURAL NETWORK (ARIMA-RBFNN) UNTUK PERAMALAN HARGA ETHEREUM**

#### **Bebas Dari Plagiarisme Dan Bukan Hasil Karya Orang Lain.**

Apabila dikemudian hari ditemukan seluruh atau sebagian penelitian/makalah/tugas akhir tersebut terdapat indikasi plagiarisme, maka saya bersedia menerima sanksi sesuai peraturan dan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sesungguhnya dan untuk dipergunakan sebagaimana mestinya.

Surabaya, 19 Agustus 2020



MUHAMMAD IKRAM FARISI  
NRP. 05211640000091

*“Halaman ini sengaja dikosongkan”*



## KATA PENGANTAR

Dengan mengucapkan rasa syukur kepada Tuhan Yang Maha Pengasih dan Maha Penyayang atas izin-Nya penulis dapat menyelesaikan buku yang sederhana ini dengan judul

**“PENERAPAN METODE HYBRID AUTOREGRESSIVE  
INTEGRATED MOVING AVERAGE-RADIAL BASIS  
FUNCTION NEURAL NETWORK (ARIMA-RBFNN)  
UNTUK PERAMALAN HARGA ETHEREUM”**

Penyelesaian Tugas Akhir ini, penulis diiringi oleh pihak-pihak yang selalu memberi dukungan, saran, dan doa sehingga penelitian berlangsung dengan lancar. Secara khusus penulis mengucapkan terima kasih dari lubuk hati terdalam kepada:

1. Allah SWT, yang telah memberikan rahmat dan hidayah serta atas izin-Nya penulis mampu mengerjakan tugas akhir ini hingga selesai.
2. Bapak Edwin Riksakomara, S.Kom, M.T selaku dosen pembimbing yang telah mencurahkan segenap tenaga, waktu dan pikiran dalam penelitian ini.
3. Bapak Ahmad Muklason, S.Kom., M.Sc., Ph.D dan Bapak Faizal Mahananto, S.Kom., M.Eng., Ph.D. selaku dosen penguji yang telah memberikan kritik dan saran kepada penulis guna meningkatkan kualitas Tugas Akhir.
4. Segenap dosen dan karyawan Departemen Sistem Informasi.
5. Orang tua serta saudara penulis, yang terus mendoakan dan memberikan dukungan kepada penulis untuk menyelesaikan tugas akhir serta pendidikan sarjana.
6. Rekan-rekan ARTEMIS yang telah memberikan banyak kenangan semasa kuliah
7. Pihak lainnya yang berkontribusi dalam tugas akhir yang belum dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa penyusunan laporan penelitian tugas akhir ini masih jauh dari kata sempurna. Untuk itu, penulis

menerima segala kritik dan saran yang dapat membantu penulis untuk dapat membuat penelitian yang lebih baik dimasa mendatang. Semoga buku tugas akhir ini dapat memberikan manfaat untuk para pembaca.

Surabaya, Juni 2020

Penulis

## DAFTAR ISI

ABSTRAK.....	v
ABSTRACT.....	vii
SURAT PERNYATAAN BEBAS PLAGIARISME.....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI.....	xiii
DAFTAR TABEL.....	xvii
DAFTAR GAMBAR .....	xix
DAFTAR KODE.....	xxi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	5
1.3 Batasan Permasalahan .....	5
1.4 Tujuan .....	5
1.5 Manfaat .....	6
1.6 Relevansi .....	6
BAB II DASAR TEORI DAN TINJAUAN PUSTAKA.....	9
2.1 Penelitian Sebelumnya .....	9
2.2 Landasan Teori .....	11
2.2.1 Ethereum .....	12
2.2.2 Konsep Dasar Peramalan.....	12
2.2.3 Metode <i>Autoregressive Integrated Moving</i> <i>Average (ARIMA)</i> .....	13
2.2.4 <i>Artificial Neural Network (ANN)</i> .....	18
2.2.5 <i>Radial Basis Neural Network (RBFNN)</i> .....	19
2.2.6 <i>Hybrid (ARIMA-RBFNN)</i> .....	20
2.2.7 Uji Performa Model Peramalan .....	21
BAB III METODOLOGI PENELITIAN .....	23
3.1 Metodologi .....	23
3.2 Uraian Metodologi.....	24
3.2.1 Identifikasi Masalah .....	24

3.2.2	Studi Literatur.....	24
3.2.3	Persiapan Data.....	24
3.2.4	Peramalan menggunakan ARIMA.....	25
3.2.5	Peramalan Menggunakan RBFNN.....	26
3.2.6	Analisa Hasil Peramalan dan Penarikan Kesimpulan.....	29
3.2.7	Penyusunan Buku Tugas Akhir.....	29
<b>BAB IV PERANCANGAN .....</b>		<b>31</b>
4.1	Persiapan Data.....	31
4.1.1	Pengumpulan data.....	31
4.1.2	Menentukan Data Pelatihan dan Pengujian.....	31
4.2	Peramalan ARIMA.....	31
4.2.1	Identifikasi Model ARIMA.....	31
4.2.2	Estimasi Parameter Model.....	33
4.2.3	Uji Diagnosa.....	34
4.2.4	Pemilihan Model Terbaik.....	34
4.2.5	Peramalan ARIMA.....	34
4.2.6	Analisis Hasil Peramalan ARIMA.....	34
4.3	Peramalan RBFNN.....	35
4.3.1	Normalisasi Data.....	35
4.3.2	Pembentukan Lag.....	36
4.3.3	Pembagian Data.....	36
4.3.4	Pembentukan Arsitektur RBFNN.....	37
4.3.5	Pelatihan Model RBFNN.....	37
4.3.6	Pengujian Model RBFNN.....	37
4.3.7	Hitung Akurasi Model RBFNN.....	38
4.3.8	Pemilihan model RBFNN.....	38
4.4	Peramalan ARIMA-RBFNN.....	38
4.5	Analisa Hasil Peramalan.....	39
<b>BAB V IMPLEMENTASI .....</b>		<b>41</b>
5.1	Pengumpulan Data.....	41
5.2	Menentukan Data Pelatihan dan Pengujian.....	41
5.3	Pembuatan dan Penerapan Model ARIMA.....	42
5.3.1	Identifikasi Model ARIMA.....	42
5.3.2	Estimasi Paramater Model.....	48
5.3.3	Uji Diagnosa.....	50

5.3.4	Pemilihan Model Terbaik .....	50
5.3.5	Peramalan ARIMA .....	51
5.3.6	Analisa Hasil Peramalan ARIMA .....	51
5.4	Penerapan dan Pembuatan Model RBFNN .....	52
5.4.1	Normalisasi .....	53
5.4.2	Pembentukan Lag .....	53
5.4.3	Pembagian Data .....	54
5.4.4	Pembentukan Model RBFNN .....	54
5.4.5	Pengujian Model .....	55
5.4.6	Hitung Akurasi Peramalan .....	55
5.4.7	Pemilihan Model Terbaik .....	56
5.5	Penerapan dan Pembuatan Model ARIMA-RBFNN .....	57
5.6	Analisa Hasil Peramalan .....	57
<b>BAB VI HASIL DAN PEMBAHASAN .....</b>		<b>59</b>
6.1	Hasil Uji Model ARIMA .....	59
6.1.1	Hasil Uji Signifikansi .....	59
6.1.2	Hasil Uji Diagnostik .....	59
6.1.3	Hasil Validasi Akurasi Model dan Uji Coba Model .....	60
6.1.4	Hasil dan Analisis Peramalan Data Aktual .....	61
6.1.5	Perbandingan MAPE .....	62
6.2	Hasil Permodelan RBFNN .....	63
6.2.1	Pengujian Model RBFNN .....	63
6.2.2	Hasil Peramalan RBFNN .....	65
6.3	Hasil Permodelan ARIMA-RBFNN .....	66
6.3.1	Pengujian Model RBFNN .....	67
6.3.2	Hasil Peramalan ARIMA-RBFNN .....	68
6.3.3	Analisa Hasil Peramalan ARIMA-RBFNN .....	69
6.4	Analisa Hasil Peramalan ARIMA, RBFNN, ARIMA-RBFNN .....	70
<b>BAB VII KESIMPULAN DAN SARAN .....</b>		<b>73</b>
7.1	Kesimpulan .....	73
7.2	Saran .....	74
<b>DAFTAR PUSTAKA .....</b>		<b>75</b>
<b>BIODATA PENULIS .....</b>		<b>79</b>

LAMPIRAN A: HASIL TRANSFORMASI.....	A-1
LAMPIRAN B: UJI SIGNIFIKANSI PARAMETER.....	B-1
LAMPIRAN C: UJI DIAGNOSA .....	C-1
LAMPIRAN D: HASIL PERAMALAN ARIMA .....	D-1
LAMPIRAN E: HASIL PENGUJIAN RBFNN .....	E-1
LAMPIRAN F: HASIL PERMALAN RBFNN .....	F-1
LAMPIRAN G: NILAI RESIDUAL ARIMA (INPUT RBFNN).....	G-1
LAMPIRAN H: HASIL PENGUJIAN RESIDUAL ARIMA (RBFNN).....	H-1
LAMPIRAN I: HASIL PERAMALAN HYBRID ARIMA- RBFNN .....	I-1

## DAFTAR TABEL

Tabel 2.1 Penelitian Sebelumnya .....	9
Tabel 2.2 Formula Transformasi .....	16
Tabel 2.3 Pola Teoritis ACF dan PACF .....	17
Tabel 2.4 Interpretasi Keakuratan MAPE .....	22
Tabel 4.1 Formula Transformasi .....	32
Tabel 4.2 Skenario Pembentukan Lag Data.....	36
Tabel 4.3 Tabel skenario perancangan model RBFNN .....	37
Tabel 5.1 Data Harga Ethereum 10 Periode Awal dan Akhir	42
Tabel 5.2 Formula Transformasi .....	44
Tabel 5.3 Penjelasan Kode 5.1 .....	44
Tabel 5.4 Estimasi Parameter data harga Ethereum .....	49
Tabel 5.5 Tabel Perbandingan Nilai AIC model ARIMA .....	51
Tabel 5.6 Penjelasan Kode 5.2 .....	52
Tabel 5.7 Penjelasan Kode 5.3 dan 5.4.....	54
Tabel 5.8 Penjelasan Kode 5.5 .....	55
Tabel 5.9 Penjelasan Kode 5.6 .....	55
Tabel 5.10 Penjelasan Kode 5.7 .....	56
Tabel 6.1 Hasil uji signifikansi parameter model ARIMA .....	59
Tabel 6.2 Hasil uji diagnostik model ARIMA.....	59
Tabel 6.3 Hasil Perhitungan MAPE model ARIMA(1,1,1)...	62
Tabel 6.4 Hasil Uji Coba Model RBFNN .....	63
Tabel 6.5 Hasil Forecast RBFNN.....	65
Tabel 6.6 Hasil Perhitungan MAPE model RBFNN(2,11,1).	66
Tabel 6.7 Hasil Pengujian model RBFNN pada Residual ARIMA .....	67
Tabel 6.8 Hasil Peramalan ARIMA-RBFNN .....	68
Tabel 6.9 Hasil Perhitungan MAPE model ARIMA-RBFNN70	
Tabel 6.10 Hasil Perbandingan Nilai MAPE pada Data Pelatihan.....	70
Tabel 6.11 Hasil Perbandingan Nilai MAPE pada data Pengujian .....	71
Tabel 6.12 Hasil Perbandingan Nilai MAPE pada Data Aktual(Train & Test).....	71

*“Halaman ini sengaja dikosongkan”*



## DAFTAR GAMBAR

Gambar 1.1 Perkembangan Harga Ethereum tahun 2017-2019 .....	2
Gambar 1.2 Bidang Keilmuan Lab RDIB .....	7
Gambar 2.1 Arsitektur RBFNN.....	20
Gambar 3.1 Alur Pengerjaan .....	23
Gambar 5.1 Uji Stationer Ragam pada data pengujian Harga Ethereum.....	43
Gambar 5.2 Grafik data latih harga Ethereum .....	45
Gambar 5.3 Grafik data latih setelah ditransformasi .....	45
Gambar 5.4 Hasil Uji Stationer .....	46
Gambar 5.5 Hasil Differencing level 1 pada data pelatihan ..	46
Gambar 5.6 Hasil Differencing level 2 pada data pelatihan ..	47
Gambar 5.7 Grafik ACF dan PACF pada data latih, diff = 1.	47
Gambar 5.8 Grafik ACF dan PACF pada data latih, diff = 2.	48
Gambar 5.9 Hasil Uji Signifikansi ARIMA(1,1,1).....	49
Gambar 5.10 Hasil Uji Diagnostik ARIMA(1,1,1).....	50
Gambar 6.1 Grafik uji coba model ARIMA(1,1,1) pada data train .....	60
Gambar 6.2 Grafik uji coba model ARIMA(1,1,1) pada data Test .....	61
Gambar 6.3 Grafik hasil uji coba model ARIMA data harga Ethereum.....	62
Gambar 6.4 Grafik Pengujian model RBFNN.....	64
Gambar 6.5 Running Time model RBFNN(2,11,1) .....	64
Gambar 6.6 Grafik Hasil Peramalan Data Pelatihan RBFNN(2,11,1).....	65
Gambar 6.7 Grafik Hasil Peramalan Data Pengujian RBFNN(2,11,1).....	66
Gambar 6.8 Diagram Perbandingan MAPE Residual dengan RBFNN .....	68
Gambar 6.9 Grafik Hasil Peramalan data harga Ethereum dengan ARIMA-RBFNN .....	69
Gambar 6.10 Grafik hasil peramalan ARIMA-RBFNN, pada 100 periode terakhir .....	69

*“Halaman ini sengaja dikosongkan”*

## DAFTAR KODE

Kode 5.1 Transformasi Box Cox.....	44
Kode 5.2 Inverse Box Cox .....	51
Kode 5.3 Pembentukan Lag .....	53
Kode 5.4 Pembentukan Variabel Input dan Target RBFNN..	53
Kode 5.5 Pembentukan Model RBFNN .....	54
Kode 5.6 Pengujian Model RBFNN.....	55
Kode 5.7 Denormalisasi Hasil Peramalan .....	56

*“Halaman ini sengaja dikosongkan”*

# **BAB I**

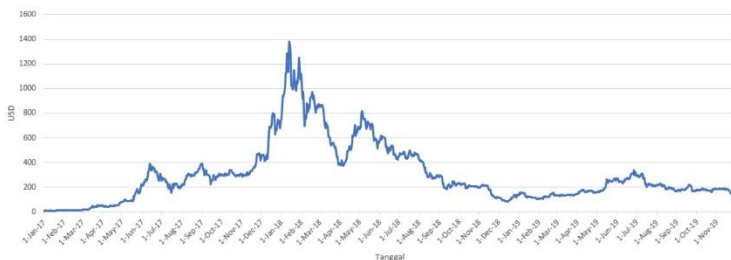
## **PENDAHULUAN**

Pada bab pendahuluan ini akan diuraikan proses identifikasi masalah penelitian yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan tugas akhir, manfaat tugas akhir, dan relevansinya terhadap pengerjaan tugas akhir. Berdasarkan uraian pada bab ini, diharapkan gambaran umum tentang permasalahan dan pemecahan masalah pada tugas akhir dapat dipahami.

### **1.1 Latar Belakang**

Pada era globalisasi saat ini, mata uang kripto atau *cryptocurrency* diciptakan sebagai salah satu instrumen investasi modern dan juga sebagai alat pembayaran alternatif. Banyak mata uang kripto yang telah dibuat semenjak beberapa decade terakhir, misalnya bitcoin, Litecoin, peercoin, ripple, dan Ethereum. Ethereum adalah mata uang virtual paling populer ke-dua setelah bitcoin, dengan kapitalisasi pasar tertinggi setelah bitcoin menurut data dari *CoinMarketCap*. Ethereum dapat menjadi mata uang kripto yang lebih populer dari pada bitcoin karena tren penggunaannya yang meningkat secara signifikan[1], salah satu faktor yang menyebabkan hal ini adalah meningkatnya minat investor untuk mencari mata uang kripto alternatif selain bitcoin[2]. Di Indonesia mata uang virtual atau *cryptocurrency* masih belum di akui sebagai alat pembayaran yang sah oleh Bank Indonesia(BI) dan Otoritas Jasa Keuangan(OJK), tetapi sekarang mata uang virtual ini kini menjadi produk komoditas dan dapat diperdagangkan di bursa berjangka Indonesia, dengan kata lain Ethereum dan mata uang kripto lain telah sah atau legal sebagai alat investasi. Selain itu, Badan Pengawas Perdagangan Berjangka Komoditi Indonesia telah mengeluarkan peraturan No. 5 tahun 2019 tentang ketentuan teknis penyelenggaraan pasar fisik asset kripto (*crypto asset*) di bursa berjangka[3].

Investasi *cryptocurrency* atau mata uang kripto beberapa tahun terakhir mengalami peningkatan jumlah peminat dan permintaan pasar, hal ini terjadi karena beberapa factor salah satunya yaitu kemudahan untuk melakukan transaksi jual beli dan investasi karena seorang trader atau investor tidak memerlukan modal yang besar untuk mulai berinvestasi. Selain itu factor lain yang menyebabkan pesatnya perkembangan investasi *cryptocurrency* saat ini adalah karena keuntungan yang diperoleh dari investasi ini terbilang cukup tinggi jika dibandingkan dengan investasi lain. Saat ini untuk menjadi seorang trader sangatlah mudah, alasan ini yang menjadi salah satu faktor berkembangnya investasi kripto karena untuk memulai berinvestasi atau mulai bertransaksi jual beli (*trading*) tidak perlu dengan modal yang besar. Investasi *cryptocurrency* sekarang telah menyentuh banyak lapisan masyarakat terutama kalangan masyarakat dengan modal kecil seperti mahasiswa. Investasi *cryptocurrency* atau mata uang kripto khususnya Ethereum merupakan salah satu investasi yang berisiko tinggi, namun dapat memberikan keuntungan yang besar apabila dapat dikelola dengan baik kapan waktu yang tepat untuk membeli dan menjual.



**Gambar 1.1 Perkembangan Harga Ethereum tahun 2017-2019**

Masalah utama yang perlu diperhatikan para trader atau investor Ethereum adalah tentang ketidakpastian harga Ethereum itu sendiri yang sangat fluktuatif. Pada gambar 1.1 yang merupakan data historis Ethereum pada tahun 2017-2019 bersumber dari website [investing.com](https://www.investing.com), dapat dilihat bahwa perkembangan harga Ethereum mengalami kenaikan dan

penurunan, *trend* yang menurun(*non-stationary*). Ketidakpastian harga ini lebih ekstrim dari ketidakpastian harga di bidang investasi lain misalnya saham atau mata uang asing(valuta asing) karena sifat alami dari *cryptocurrency* yang tidak diatur oleh suatu Lembaga seperti bank sentral, tetapi dikendalikan oleh pemilik *cryptocurrency* itu sendiri karena harga berubah berdasarkan permintaan(*supply*) dan penawaran(*demand*) yang naik atau turun sesuai pasar. Ketidakpastian harga yang terjadi ini tentu tidak disenangi oleh para trader ataupun investor *cryptocurrency* Ethereum karena dapat menimbulkan resiko yang tidak diinginkan pada investasi mereka, sedangkan ketidakpastian ini merupakan suatu hal yang lazim terjadi dan tidak dapat dihindarkan ketika melakukan investasi. Salah satu cara untuk menghadapi ketidakpastian harga ini adalah dengan meramalkannya(*forecast*), dengan ini kita dapat memperkirakan atau memprediksi nilai kenaikan atau penurunan harga Ethereum dimasa mendatang.

*Forecasting* digunakan untuk memprediksi sesuatu yang kemungkinan besar akan terjadi dengan melibatkan data masa lalu dan menempatkannya dimasa yang akan datang dengan suatu bentuk model matematis agar penggunaanya dapat mengambil keputusan secara tepat berdasarkan hasil dari *forecasting*.

Pada saat ini sudah banyak metode yang dapat digunakan untuk melakukan peramalan. Beberapa metode yang paling umum digunakan dalam *forecasting time series* adalah metode ARIMA dan ANN. Kedua metode ini memiliki kekurangan dan keunggulan masing – masing dalam menyelesaikan masalah. Metode ARIMA sering digunakan untuk menganalisis data *time series*, karena mampu menghasilkan nilai peramalan yang baik. ARIMA sangat baik tingkat akurasi dalam peramalan untuk rentang waktu yang cukup pendek dengan data *time series linear* [4]. Sedangkan untuk peramalan jangka panjang akurasi kurang baik, karena cenderung akan menghasilkan nilai peramalan yang datar. Selain itu, jika terdapat komponen

*time series nonlinear* pada data observasi, maka akurasi dari model ARIMA akan berkurang. Tidak seperti ARIMA, model *neural network* (NN) adalah metode peramalan yang dapat digunakan untuk meramalkan data *time series non-linear*, dan juga mengabaikan stasioneritas data[4].

Kedua komponen yang telah disebutkan memiliki kelebihan masing – masing dan dapat melengkapi kekurangan model lainnya. Selain itu, pada kenyataannya dalam suatu data *time series* dapat memiliki komponen *linear* dan *nonlinear* sekaligus dan satu model mungkin tidak dapat menangkap kedua komponen tersebut, hal ini terjadi karena permasalahan di dunia nyata biasanya merupakan masalah yang lebih kompleks. Bahkan menurut penelitian sebelumnya, hasil peramalan dari kombinasi beberapa model memiliki tingkat akurasi yang lebih baik dari model individu[4]. Oleh karena itu, kombinasi dari metode ARIMA dan ANN dapat memaksimalkan hasil peramalan yang mengandung komponen *linear* dan *non-linear*.

Pada ANN, ada beberapa algoritma pembelajaran yang dikembangkan untuk peramalan data *time series*. Salah satunya adalah *Radial Basis Function Neural Network* (RBFNN), dimana menggunakan fungsi radial basis. Algoritma pembelajaran dari RBFNN cukup handal dan dapat menyelesaikan permasalahan peramalan data *time series* yang mengandung komponen *non-linear* dan *non-stationer*[4].

Tujuan dari penelitian ini adalah untuk membangun model prediksi harga Ethereum dengan metode ARIMA, RBFNN dan penggabungan metode ARIMA dan RBFNN. Metode ARIMA dapat menangkap hasil peramalan yang memiliki komponen linier dan menggabungkannya dengan RBF untuk mencari hubungan linier dan nonlinier, sehingga diharapkan dapat menghasilkan hasil yang terbaik.

Penelitian ini diharapkan dapat membantu masyarakat yang sedang melakukan kegiatan *trading cryptocurrency* Ethereum



maupun yang sedang berinvestasi dalam Ethereum agar mampu mengambil keputusan yang tepat untuk menghasilkan profit yang maksimal atau mengurangi tingkat resiko yang dihasilkan dari kegiatan *trading* maupun investasi berdasarkan pada hasil peramalan.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, permasalahan yang akan diangkat pada penelitian tugas akhir ini adalah sebagai berikut:

1. Bagaimana membangun peramalan harga Ethereum dengan metode ARIMA, RBFNN dan Hybrid ARIMA-RBFNN ?
2. Bagaimana hasil dan tingkat akurasi yang dihasilkan dari peramalan harga Ethereum dengan menggunakan metode ARIMA, RBFNN dan hybrid ARIMA-RBFNN ?

## 1.3 Batasan Permasalahan

Berdasarkan deskripsi permasalahan diatas, Adapun Batasan masalah dari penelitian tugas akhir ini adalah sebagai berikut :

1. Data yang digunakan adalah data harian Harga Ethereum dari tanggal 1 Januari 2017 sampai dengan 31 Desember 2019 yang bersumber dari website investing.com
2. Variabel yang digunakan adalah variable Price
3. Dalam menentukan pembagian data pelatihan dan pengujian untuk penelitian ini menggunakan persentase 70 : 30.
4. Perangkat lunak yang digunakan dalam penelitian yaitu Microsoft Excel, Minitab, dan Rstudio.

## 1.4 Tujuan

Berdasarkan pada rumusan masalah yang telah disebutkan sebelumnya, tujuan yang akan dicapai dalam penulisan tugas akhir ini adalah :

1. Membangun model yang sesuai dengan metode ARIMA, RBFNN dan Hybrid ARIMA-RBFNN dalam meramalkan harga Ethereum
2. Mengetahui hasil dan tingkat akurasi dari peramalan harga Ethereum dengan menggunakan metode ARIMA, RBFNN dan Hybrid ARIMA-RBFNN

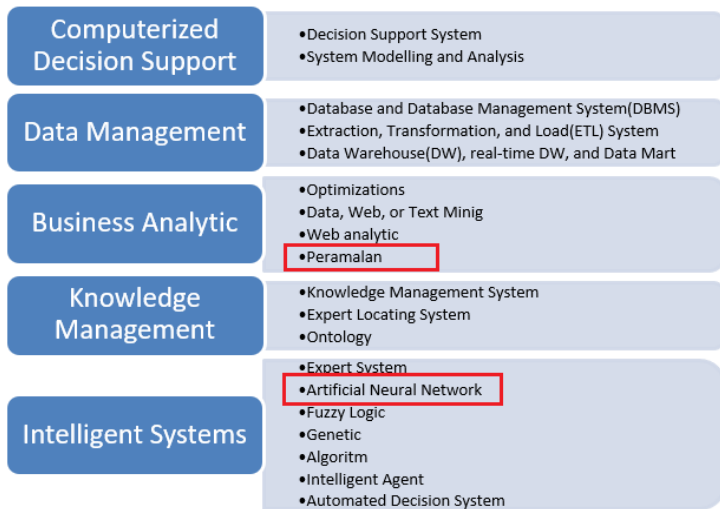
### **1.5 Manfaat**

Penyusunan tugas akhir ini diharapkan dapat membawa manfaat, diantaranya :

1. Memberikan gambaran bagi masyarakat yang melakukan kegiatan investasi dibidang *cryptocurrency* khususnya Ethereum agar dapat mengetahui dan dapat memantau harga Ethereum untuk beberapa waktu kedepan sehingga dapat membantu melakukan pengambilan keputusan investasi
2. Memberikan kontribusi pengetahuan mengenai penerapan metode ARIMA yang dikombinasikan dengan RBFNN
3. Dapat digunakan sebagai referensi dalam penerapan ilmu peramalan dengan metode ARIMA, RBFNN, dan Hybrid ARIMA-RBFNN dalam penelitian berikutnya

### **1.6 Relevansi**

Usulan tugas akhir ini memiliki relevansi dengan mata kuliah Teknik Peramalan, Analitika Bisnis, dan Penggalian Data. Hal ini dikarenakan pada penyusunan tugas akhir ini memanfaatkan bidang keilmuan dari mata kuliah tersebut. Selain itu Tugas akhir ini memiliki tujuan terkait dengan pemanfaatan data yang akan ditransformasikan menjadi informasi yang dapat mendukung pengambilan keputusan.



**Gambar 1.2 Bidang Keilmuan Lab RDIB**

Berdasarkan pada bidang ilmu yang menjadi focus utama dan tujuan penelitian yang sudah dijelaskan pada gambar 1.2, dapat disimpulkan bahwa usulan tugas akhir ini memiliki relevansi dengan ranah penilitan pada Laboratorium Rekayasa Data dan Intelegensi Bisnis (RDIB) yang ada pada Departemen Sistem Informasi ITS

*“Halaman ini sengaja dikosongkan”*

## **BAB II**

### **DASAR TEORI DAN TINJAUAN PUSTAKA**

Pada bab ini akan dijelaskan mengenai penelitian yang telah dilakukan sebelumnya yang berhubungan dengan pengerjaan tugas akhir dan dasar – dasar teori yang digunakan sebagai acuan atau pedoman dalam proses pengerjaan tugas akhir ini.

#### **2.1 Penelitian Sebelumnya**

Pada pengerjaan tugas akhir ini, terdapat beberapa penelitian yang dijadikan acuan atau landasan yang telah dirangkum. Referensi studi sebelumnya ini dipilih atas dasar kesamaan metode dan topik yang digunakan pada tugas akhir. Beberapa penelitian atau studi sebelumnya yang dijadikan acuan dalam pengerjaan tugas akhir ini, disajikan dalam tabel 2.1.

**Tabel 2.1** Penelitian Sebelumnya

<b>Penelitian 1 [5]</b>	
Judul Penelitian	<b>Predicting Price Changes in Ethereum</b>
Penulis	Matthew Chen, Neha Narwal and Mila Schultz
Tahun	2015
Deskripsi Umum Penelitian	Penelitian ini membahas tentang prediksi perubahan harga Ethereum menggunakan 6 metode (Logistic Regression, Naïve Bayes, Support Vector Machine, Random Forest, ARIMA, dan RNN) untuk menentukan metode yang terbaik dalam memprediksi perubahan harga Ethereum. Data yang digunakan merupakan data harga Ethereum dengan interval 1 jam dari tanggal 30 Agustus 2015 sampai 2 Desember 2017. Hasil dari penelitian ini menyimpulkan bahwa metode ARIMA merupakan metode dengan performa terbaik.
Keterkaitan dengan Tugas Akhir	Penelitian ini membahas topik yang sama yaitu memprediksi harga Ethereum. Penelitian ini dapat menjadi refensi terkait penelitian yang pernah dilakukan terkait peramalan dengan menggunakan metode ARIMA serta dalam proses menentukan

	metode yang digunakan berdasarkan pada penelitian ini..
<b>Penelitian 2 [4]</b>	
Judul Penelitian	<b>PERAMALAN DERET WAKTU MENGGUNAKAN MODEL FUNGSI BASIS RADIAL (RBF) DAN AUTO REGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA)</b>
Penulis	Dian Tri Wiyanti & Reza Pulungan
Tahun	2012
Deskripsi Umum Penelitian	Penelitian ini membahas tentang kombinasi metode ARIMA dan RBF dalam meramalkan Data time series, data yang digunakan pada penelitian ini merupakan data Indeks Harga Perdagangan Besar (IHPB). Dari hasil pengujian yang dilakukan pada penelitian ini dapat disimpulkan bahwa model dari kombinasi metode ARIMA-RBF merupakan model terbaik saat digunakan untuk pengujian pada data IHPB karena memiliki nilai RMSE, MAPE, maupun waktu komputasi yang terbilang lebih baik dari pada model individu (ARIMA dan RBF)
Keterkaitan dengan Tugas Akhir	Penelitian ini menggunakan metode yang sama dengan metode yang akan digunakan pada tugas akhir. Penelitian ini dapat dijadikan referensi dalam pengerjaan tugas akhir untuk metodologi pengerjaan serta dasar teori.
<b>Penelitian 3 [1]</b>	
Judul Penelitian	<b>Peramalan Harga Garam Konsumsi Menggunakan Artificial Neural Network Feedforward-Backpropagation (Studi Kasus : PT. Garam Mas, Rembang, Jawa Tengah)</b>
Penulis	Mochammad Yusuf Habibi dan Edwin Riksakomara
Tahun	2017
Deskripsi Umum Penelitian	Penelitian ini membahas penerapan metode <i>Artificial Neural Network</i> (ANN) dalam peramalan harga garam konsumsi dengan studi kasus PT. Garam Mas, menggunakan tools matlab. Hasil yang didapatkan pada penelitian ini disimpulkan bahwa metode ANN dapat dilakukan untuk

	melakukan peramalan harga Garam dengan hasil yang sangat baik karena memiliki nilai MAPE <10% atau lebih tepatnya 4,394%.
Keterkaitan dengan Tugas Akhir	Penelitian ini dapat dijadikan referensi dalam pemahaman penggunaan metode ANN untuk melakukan peramalan
<b>Penelitian 4 [7]</b>	
Judul Penelitian	<b>PERAMALAN DERET WAKTU DENGAN MENGGUNAKAN AUTOREGRESSIVE (AR), JARINGAN SYARAF TIRUAN RADIAL BASIS FUNCTION (RBF) DAN HIBRID AR-RBF PADA INFLASI INDONESIA</b>
Penulis	Al Hikmah, Arief Agoestanto, Riza Arifudin
Tahun	2018
Deskripsi Umum Penelitian	Penelitian ini membahas tentang penggunaan metode Hybrid AR( <i>Autoregressive</i> ) dan Neural Network dalam meramalkan harga Inflasi di Indonesia berdasarkan Indeks Harga Konsumen(IHK) pada bulan Januari 2003 sampai dengan September 2016. Pada penelitian ini menggunakan tools MATLAB, dan didapatkan model hybrid AR-RBF (2-1-1) dengan nilai MAPE untuk peramalan 9,37%. Pada penelitian ini disimpulkan peramalan dengan metode hybrid dapat dilakukan untuk meramalkan data inflasi, tetapi nilai MAPE terkecil didapatkan dari peramalan dengan metode RBF.
Keterkaitan dengan Tugas Akhir	Penelitian ini membahas mengenai metode yang mirip dengan metode yang akan digunakan dalam pengerjaan tugas akhir dan dapat dijadikan referensi pemilihan algoritma yang digunakan dalam ANN serta sebagai referensi untuk metode yang akan dipakai dalam menggunakan MATLAB.

## 2.2 Landasan Teori

Pada sub bab ini akan dibahas teori – teori yang mendukung serta berkaitan dengan tugas akhir yang dikerjakan.

### 2.2.1 Ethereum

Pada tahun 2014, Ethereum diperkenalkan kepada publik oleh Vitalik Buterin, ia merupakan salah satu pengembang Bitcoin dari tahun 2011 hingga 2013. Sama seperti mata uang kripto lainnya, Ethereum juga mengadopsi teknologi Blockchain dan satuan mata uang dari Ethereum adalah Ether(ETH). Blockchain dibuat pada tahun 2009, merupakan sistem terdesentralisasi yang dapat mencatat semua transaksi yang dilakukan.

Etherum diciptakan dengan tujuan untuk membuat platform di mana program *Smart Contract* dapat dibuat dan dijalankan. *Smart Contracts* adalah program yang dapat mengeksekusi sebuah kesepakatan yang ada dalam suatu kontrak secara otomatis. *Smart Contracts* memungkinkan transaksi dilakukan segera tanpa penundaan, hambatan dan pihak ketiga atau perantara, dan dapat dilakukan pada jaringan yang terdesentralisasi.

### 2.2.2 Konsep Dasar Peramalan

Peramalan (*Forecasting*) adalah sebuah kegiatan yang dilakukan untuk memperkirakan suatu kejadian yang akan terjadi berdasarkan pada data – data atau kejadian pada masa lalu[8]. Melakukan peramalan artinya mengestimasi nilai suatu variabel berdasarkan nilai variabel itu sendiri atau variabel terkait. Peramalan juga dapat dipengaruhi oleh kemampuan *judgment*, dan kemampuan ini akan dipengaruhi oleh data historis dan pengalaman [9].

Terdapat banyak macam jenis peramalan yang ada saat ini bergantung pada aspek yang dilihat. Namun yang umum ada 2 jenis peramalan, yaitu peramalan yang didasarkan pada aspek jangka waktunya dan peramalan yang berdasarkan aspek sifatnya.

Untuk peramalan yang berdasarkan aspek jangka waktunya, terdapat 3 kategori [9], yaitu sebagai berikut:



1. Peramalan jangka pendek, periode ramalan kurang dari 3 bulan.
2. Peramalan jangka menengah, dimana periode ramalan antara 3 bulan hingga 18 bulan.
3. Peramalan jangka panjang, periode ramalan lebih dari 3 tahun.

Untuk peramalan yang dilihat berdasarkan aspek sifatnya, terdapat 2 kategori [9], yaitu sebagai berikut:

1. Peramalan Kualitatif, peramalan yang berdasarkan pada data yang bersifat subjektif dan berbentuk kalimat verbal pada masa lalu. Hasil peramalan kualitatif didasari pada pengamatan terhadap peristiwa yang terjadi pada periode sebelumnya, dan dipadukan dengan opini atau pemikiran dari orang yang meramalkannya.
2. Peramalan Kuantitatif, peramalan yang berdasarkan pada data yang bersifat objektif pada periode sebelumnya yang didapatkan dari pengamatan nilai – nilai sebelumnya. Hasil peramalan bergantung pada metode yang digunakan, metode yang berbeda akan mendapatkan hasil peramalan yang berbeda pula. Dalam peramalan kuantitatif, ada dua model [10], yaitu berikut:
  - a. Model deret berkala (*time series*)  
Model tersebut didasarkan pada analisis analisis pola keterkaitan variable yang akan diprediksi dengan variable waktu yang di sebut deret waktu.
  - b. Model kausal,  
Model ini didasarkan pada metode sebab-akibat dengan mengukur beberapa variabel independen dan dampaknya terhadap variabel yang akan diprediksi.

### 2.2.3 Metode *Autoregressive Integrated Moving Average* (ARIMA)

Metode ARIMA (*Autoregressive Integrated Moving Average*), umumnya disebut sebagai metode Box-Jenkins, adalah metode yang dikembangkan oleh George Box dan Gwilym Jenkins pada tahun 1970[11]. Metode ARIMA adalah metode yang baik

digunakan untuk peramalan dengan rentang waktu yang pendek. Model ARIMA adalah model yang sepenuhnya mengabaikan variabel independen saat membuat peramalan. Nilai yang digunakan ARIMA untuk melakukan peramalan adalah nilai masa lalu dan masa kini dari variabel dependen untuk menghasilkan prediksi jangka pendek yang akurat[12]. Model ARIMA dikelompokkan ke dalam empat kelompok, yaitu model AR, MA, ARMA, dan ARIMA.

1. Model Autoregressive (AR) adalah model yang menggambarkan bahwa nilai variabel dependen saat ini dipengaruhi oleh nilai variabel dependen pada masa sebelumnya. Bentuk umum dari model AR dengan ordo  $p$  (AR( $p$ )) atau model ARIMA ( $p,0,0$ ) dapat dilihat pada rumus (2-1).

$$Z_t = \mu' + \phi_1 Z_{t-1} + \dots + \phi_p Z_{t-p} + e_t \quad (2-1)$$

dimana,

- $Z_t$  = Variabel yang diramalkan
- $\mu$  = konstanta
- $Z_{t-p}$  = variabel bebas
- $\phi_p$  = koefisien parameter *Autoregressive* ke- $p$
- $e_t$  = sisaan pada saat ke- $t$

2. Model Moving Average (MA) adalah model yang menggambarkan hubungan antara nilai-nilai dalam rangkaian waktu saat ini dan kesalahan acak yang terjadi pada periode waktu sebelumnya. Bentuk umum dari MA dengan ordo  $q$  (MA( $q$ )) atau model ARIMA ( $0,0,q$ ) dapat dilihat pada rumus (2-2).

$$Z_t = \mu + e_t - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q} \quad (2-2)$$

dimana,

- $Z_t$  = Variabel yang diramalkan
- $\mu$  = konstanta
- $e_{t-q}$  = sisaan pada saat  $t - q$

- $\theta_q$  = koefisien parameter *Moving Average* ke-q

3. Model *Autoregressive Moving Average* (ARMA) merupakan model gabungan dari *Autoregressive* (AR) dan *Moving Average* (MA) tanpa melalui proses *differencing*. Model ARMA mengasumsikan bahwa data periode sekarang dipengaruhi oleh data periode sebelumnya dan nilai sisa periode sebelumnya [11]. Bentuk persamaan model dapat dilihat pada rumus (2-3).

$$Z_t = \mu + \phi_p Z_{t-p} + e_t - \theta_q e_{t-q} \quad (2-3)$$

4. Model *Autoregressive Integrated Moving Average* (ARIMA) merupakan pengembangan dari model ARMA yang memiliki *differencing*. Asumsi yang digunakan agar model ini dapat digunakan adalah data time series harus stationer yang berarti bahwa rata-rata variasi dari data yang dimaksud adalah konstan. Untuk mengatasi ketidakstationeran pada suatu data maka proses *differencing* perlu dilakukan agar data menjadi stationer. Bentuk notasi dari model ini yaitu ARIMA(p,d,q) dengan orde d yang merupakan parameter jumlah proses *differencing*. Pada umumnya model ARIMA dapat dilihat pada rumus (2-4).

$$\phi_p(B)D_d Z_t = \mu + \theta_q(B)e_t \quad (2-4)$$

dimana,

- $\phi_p$  = koefisien parameter *autoregressive* ke-p
- $\theta_q$  = koefisien parameter *moving average* ke-q
- B = operator backshift
- D = differencing
- $\mu$  = konstanta
- $e_t$  = sisaan pada saat ke-t
- p = derajat *Autoregressive*
- d = tingkat proses *differencing*
- q = derajat *Moving Average*

Adapun langkah – langkah yang dilakukan dalam menganalisa metode ARIMA adalah sebagai [13] berikut :

### 1. Identifikasi Model

Pada tahap ini diawali dengan menganalisa data, yaitu dengan membuat plot data deret waktu untuk mengetahui kestasioneran pada data yang digunakan.

Data yang digunakan harus stationer dalam rata-rata maupun varian, jika data yang digunakan tidak stationer maka data perlu distasionerkan terlebih dahulu. Apabila data tidak stationer dalam varian, maka perlu dilakukan transformasi data dengan melihat nilai estimasi lambda ( $\lambda$ ) yang dapat dilihat pada tabel 2.2.

**Tabel 2.2 Formula Transformasi**

$\lambda$	Transformasi
-1	$1/X_t$
-0.5	$1/\sqrt{X_t}$
0	$\text{Ln}(X_t)$
0.5	$\sqrt{X_t}$
1	Tidak ditransformasi

Selanjutnya jika data telah stationer terhadap varian, maka data harus stationer terhadap mean. Apabila data belum stationer terhadap rata-rata (mean), maka dapat dilakukan proses *differencing* dengan rumus (2-5)

$$Z_t = X_t - X_{t-1} \quad (2-5)$$

Dimana,

- $X_t$  = Data time series pada periode t
- $X_{t-1}$  = Data time series pada periode t-1

Pada proses *differencing*, data yang digunakan adalah data hasil transformasi apabila data aktual tidak stationer terhadap varian, jika data telah stationer terhadap varian maka data yang digunakan adalah data aktual[14]. Proses *differencing* dapat dilakukan lebih dari satu kali sampai data menjadi stationer.

Selanjutnya, untuk menentukan orde AR(p) dan MA(q) dapat ditentukan dengan memplot ACF (*Autocorrelation Function*) dan PACF (*Partial Autocorrelation Function*). Identifikasi orde AR(p) dapat dilihat dengan plot PACF dan orde MA(q) dengan plot ACF terhadap data yang sudah stationer. Bentuk model yang diidentifikasi berdasarkan dari plot data yang dilakukan dapat disimpulkan dari tabel 2.3[13].

**Tabel 2.3 Pola Teoritis ACF dan PACF**

<b>Type Model</b>	<b>Pola ACF</b>	<b>Pola PACF</b>
AR(p)	<i>dies down</i> (Menurun secara eksponensial/sinusoidal)	<i>cut off</i> (terputus setelah lag ke p)
MA(q)	<i>cut off</i> setelah lag ke q	<i>dies down</i> (menurun secara eksponensial / sinusoidal)
ARMA(p,q)	<i>dies down</i> setelah lag (q-p) atau (p-q)	<i>dies down</i> setelah lah (p-q) atau (q-p)

## 2. Estimasi Parameter

Pada tahap ini, kita melakukan pemilihan terhadap model yang telah di estimasi dengan melakukan pengujian hipotesis pada parameter model yang telah di estimasi.

## 3. Uji Diagnosis

Setelah mendapatkan estimasi parameter ARIMA, langkah selanjutnya adalah memilih model yang dapat menginterpretasikan data dengan baik. Yang perlu dilakukan adalah melihat apakah nilai residual atau error

tersebut merupakan nilai acak sehingga merupakan residual yang relatif kecil. Untuk menentukan model terbaik dapat ditentukan dengan melihat nilai *standard error estimate*[15] dengan rumus :

$$S = \left[ \frac{SSE}{n-n_p} \right]^{1/2} = \left[ \frac{\sum_{t=1}^n (Y_t - \hat{y}_t)^2}{n-n_p} \right]^{1/2} \quad (2-6)$$

Dimana,

- SSE : standard error
- $Y_t$  : nilai actual waktu ke-t
- $\hat{Y}_t$  : nilai forecast pada waktu ke-t

Model terbaik yang akan dipilih merupakan model dengan standard error estimate (S) yang paling kecil. Pengujian Diagnostik dapat dilakukan dengan Uji Q-Ljung-Box, untuk mengetahui apakah residual bersifat acak atau memenuhi asumsi white noise[7].

#### 4. Forecast

Setelah mendapatkan model yang sesuai, Langkah selanjutnya adalah melakukan peramalan dengan model tersebut.

##### 2.2.4 Artificial Neural Network (ANN)

ANN adalah suatu metode komputasi yang meniru sistem pada jaringan syaraf manusia. Metode ini menggunakan elemen perhitungan nonlinier dasar yang disebut *neuron* yang saling berhubungan membentuk suatu jaringan yang terorganisir, sehingga mirip dengan jaringan syaraf manusia.

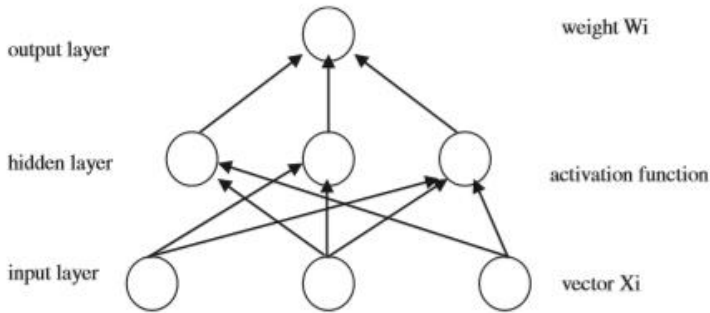
ANN pada dasarnya terdiri dari 3 lapisan(*layer*), yaitu lapisan input(*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan output(*output layer*). Setiap lapisan terdiri dari beberapa *neuron* yang saling terhubung. *Input layer* akan menggunakan bobot kedatangan ( $w$ ) tertentu untuk menerima informasi ( $\alpha$ ). Setelah itu akan dilakukan penjumlahan bobot(*weight*) pada

lapisan tersembunyi. Kemudian hasil dari penjumlahan tersebut akan dibandingkan dengan *threshold* atau nilai ambang. Jika nilai melewati ambang batas, maka akan diteruskan ke output layer, sebaliknya jika nilai tidak melebihi ambang batas maka tidak akan diteruskan ke output layer[6].

### 2.2.5 Radial Basis Neural Network (RBFNN)

Model *Neural Network* RBF merupakan salah satu bentuk MLP atau *multilayer perceptron* yang akan memperbaiki nilai – nilai bobot, nilai tengah, dan jarak antar data agar dapat mengurangi kesalahan yang terjadi pada output[7]. Model RBFNN menggunakan fungsi basis sebagai fungsi aktivasinya untuk setiap *neuron* pada hidden layer. Ada beberapa fungsi basis, namun fungsi basis yang sering digunakan adalah fungsi *gaussian* karena mempunyai sifat lokal, yaitu apabila input dekat dengan rata-rata(pusat) maka fungsi akan menghasilkan nilai satu dan bila input jauh dari rata – rata maka fungsi akan menghasilkan nilai nol.

Model RBFNN terbagi menjadi 3 lapisan, dimana lapisan pertama merupakan lapisan input(*input layer*) yang menerima suatu vector input  $x$ . Kemudian dilanjutkan ke lapisan tersembunyi(*hidden layer*) yang akan memproses input secara nonlinear dengan fungsi aktivasi. Setelah itu dilanjutkan ke lapisan output(*output layer*) yang akan memproses output dari *hidden layer* secara linear[16]. Gambar 2.1 memperlihatkan mengenai struktur dari RBFNN dimana terdapat komponen *vector input*  $X = (x_1, x_2, \dots, x_p)$ , pada lapisan hidden layer terdapat *activation function*(fungsi basis) dan *output* hasil dari kombinasi linear dari fungsi aktivasi dan bobot  $W = (w_1, w_2, \dots, w_m)$ [16].



**Gambar 2.1** Arsitektur RBFNN

Model RBFNN dapat dilihat pada rumus (7).

$$F(x) = \sum_{i=1}^N w_j \varphi(\|x - x_i\|) \quad (2-7)$$

Dimana,  $\varphi(\|x - x_i\|) \{i = 1, 2, 3, \dots, N\}$ , adalah himpunan fungsi nonlinear yang disebut fungsi *radial basis*(RBF), dan  $\| \cdot \|$  yang merupakan norm jarak *Euclid*[7].

### 2.2.6 Hybrid (ARIMA-RBFNN)

Metode *hybrid* atau campuran adalah metode kombinasi dari beberapa metode dalam fungsi suatu sistem menjadi satu. Metode ARIMA dan ANN merupakan metode untuk mengatasi masalah *linear* atau *non-linear*[17]. Karena pada data time series jarang ditemukan kondisi yang murni linier dan murni nonlinier, maka dengan metode kombinasi yang memiliki kemampuan memodelkan komponen linier dan nonlinier sekaligus dapat digunakan sebagai strategi praktis. Umumnya, Anda dapat menulis kombinasi model time series dengan struktur autokorelasi linier dan nonlinier seperti rumus berikut,

$$y_t = Lt + Nt \quad (2-8)$$

Dimana:

- $Lt$ : Komponen linier
- $Nt$ : komponen nonlinier



Adapun langkah pertama digunakan ARIMA untuk linier komponen dan sisaan / residual dari model linier merupakan hubungan nonlinier.  $e_t$  merupakan residual / nilai sisa,  $y_t$  adalah nilai aktual dan  $Z_t$  adalah hasil peramalan ARIMA pada waktu ke  $t$  :

$$e_t = y_t - Z_t \quad (2-9)$$

Kemudian  $e_t$  dimodelkan menggunakan ANN. Sehingga model ANN dari nilai sisa / residual ARIMA menjadi sebagai berikut :

$$e_t = f(e_{t-1}, e_{t-2}, \dots, e_{t-n}) + \varepsilon_t \quad (2-10)$$

Dimana  $f$  merupakan fungsi nonlinear yang ditentukan oleh model ANN dalam kasus ini menggunakan RBFNN dan  $\varepsilon_t$  adalah nilai error. Hasil ramalan dari residual oleh ANN(RBFNN) digambarkan sebagai  $e'_t$ , Sehingga peramalan dengan menggunakan hybrid model dapat ditulis sebagai berikut :

$$Z'_t = Z_t + e'_t \quad (2-11)$$

### 2.2.7 Uji Performa Model Peramalan

Untuk menguji akurasi dari hasil peramalan, beberapa metode pengujian dapat digunakan. Salah satu metode pengukuran akurasi peramalan biasanya dihitung menggunakan MAPE (Mean Absolute Percentage Error). MAPE merupakan perhitungan yang menampilkan nilai absolut rata-rata dari nilai error peramalan atau selisih antara nilai aktual dan nilai peramalan. MAPE merupakan indikator yang paling umum digunakan untuk menentukan akurasi hasil peramalan [18]. Jika nilai MAPE semakin kecil, hasil peramalan akan lebih akurat. Berikut ini adalah rumus MAPE secara umum :

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|y_t - f_t|}{y_t} \times 100\% \quad (2-9)$$

Dimana,

$n$  = banyaknya data

$Y_t$  = nilai actual pada periode ke  $t$

$ft$  = nilai hasil peramalan pada periode ke  $t$

Tingkat akurasi dari hasil peramalan dikatakan sangat baik apabila nilai MAPE  $< 10\%$ , dan buruk bila MAPE  $> 50\%$  [19]. Hal ini dapat dilihat pada tabel 2.4.

**Tabel 2.4 Interpretasi Keakuratan MAPE**

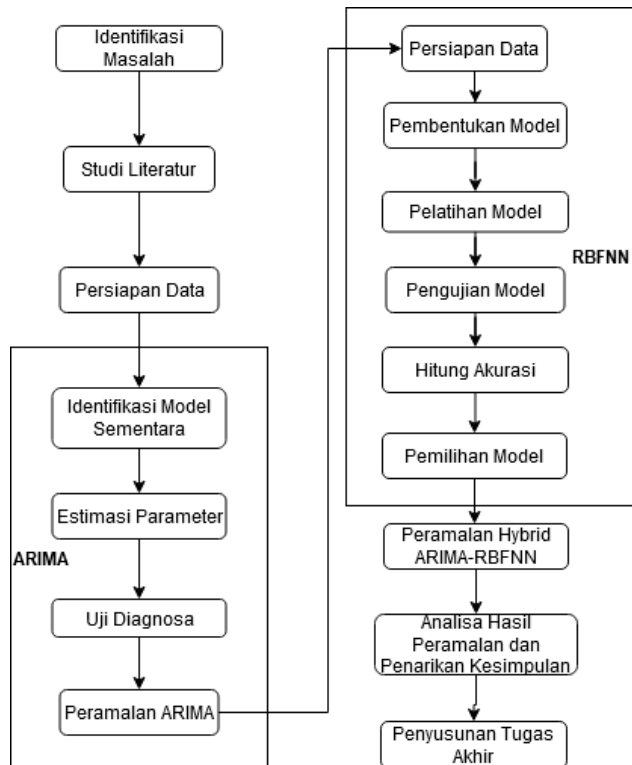
MAPE	INTERPRETASI
$< 10\%$	Sangat Akurat
10% - 20%	Baik
20% - 50%	Cukup
$> 50\%$	Tidak Akurat

## BAB III METODOLOGI PENELITIAN

Pada bab metodologi ini akan dijelaskan mengenai tahapan atau alur pengerjaan yang akan dilaksanakan selama pengerjaan tugas akhir ini. Metodologi digunakan sebagai acuan agar pengerjaan tugas akhir ini berjalan dengan terarah dan sistematis.

### 3.1 Metodologi

Penjelasan mengenai alur penelitian yang akan dihasilkan dapat dilihat pada gambar 3.1.



**Gambar 3.1 Alur Pengerjaan**

## 3.2 Uraian Metodologi

Berdasarkan pada alur metodologi pada Gambar 3.1, sub bab ini akan menjelaskan secara lebih rinci mengenai tahapan yang akan dilakukan dalam penyusunan tugas akhir.

### 3.2.1 Identifikasi Masalah

Pada tahap ini akan dilakukan analisis permasalahan yang akan dijadikan topik dalam pengerjaan tugas akhir

### 3.2.2 Studi Literatur

Pada tahapan ini akan dilakukan proses pengumpulan informasi yang dibutuhkan dalam proses pengerjaan tugas akhir. Informasi ini dapat ditemukan atau didapatkan dari penelitian terdahulu, buku, maupun dokumen lain yang berkaitan dengan topik yang diangkat. Hal ini dilakukan untuk mengetahui permasalahan serta dapat menunjang pengetahuan dalam melakukan penelitian tugas akhir ini. Adapun informasi yang diperlukan dalam penyusunan tugas akhir ini adalah informasi seputar Teknik Peramalan, Cryptocurrency Ethereum, Metode Peramalan ARIMA, RBFNN, dan hybrid ARIMA-RBFNN.

### 3.2.3 Persiapan Data

Pada tahapan ini akan dilakukan pencarian data yang terkait dengan tugas akhir dan akan digunakan dalam pengerjaan tugas akhir. Data yang dibutuhkan berdasarkan pada topik yang diangkat pada penelitian tugas akhir ini adalah data historis harga Ethereum. Dimana data yang diperlukan dalam pengerjaan tugas akhir ini diperoleh dari website [investing.com](https://www.investing.com). Data yang akan digunakan merupakan data *time series* (runtun waktu) dengan tipe data *float* dan tipe atribut *nominal*. Data yang diperoleh tidak memiliki *missing value*. Jumlah keseluruhan data yang diperoleh adalah 1095 buah, dengan rentang waktu harian mulai dari tanggal 1 Januari 2017 hingga 31 Desember 2019. Penggunaan data harian dikarenakan lebih

baik dalam menangkap pola kejadian dibandingkan data mingguan dan bulanan serta baik untuk memprediksi pada rentang waktu yang pendek atau menengah, selain itu pada data Ethereum perubahan pada harga terjadi sesuai pada pola permintaan dan penawaran yang terjadi serta tidak menentu maka penggunaan data harian baik digunakan untuk menyimpulkan pola perubahan harga tersebut. Kecepatan pemrosesan data harian dibandingkan data dengan rentang jam lebih menguntungkan apabila model yang dibuat ingin diaplikasikan pada sebuah platform.

### 3.2.4 Peramalan menggunakan ARIMA

Pada tahapan ini akan dijelaskan mengenai tahapan yang dilakukan dalam peramalan dengan model ARIMA dari data harga Ethereum yang nanti hasil dari pengujian ini yang berupa nilai error atau residual akan dijadikan input untuk peramalan menggunakan metode RBFNN.

#### 3.2.4.1 Identifikasi Model ARIMA

Pada tahap ini diawali dengan membuat data plot *time series* dari data historis terkait harga Ethereum untuk mengetahui pola data yang dimiliki. Setelah melakukan plot data tersebut dilakukan uji stasioner baik dalam rata-rata maupun varian. Jika data yang digunakan belum stasioner dalam varian, maka perlu untuk dilakukan proses transformasi. Apabila data belum stasioner dalam rata-rata (*mean*), maka perlu untuk menjadikan data stasioner dengan proses *differencing*. Selanjutnya dapat dilakukan identifikasi model sementara dengan melihat dari grafik ACF dan PACF, dimana terdapat garis interval yang dijadikan Batasan dalam mengestimasi nilai orde  $p$  dan  $q$ . Nilai Koefisien ACF yang melebihi garis interval akan dijadikan nilai dari orde MA( $q$ ) dan untuk nilai Koefisien PACF yang melebihi batasan garis interval akan dijadikan nilai dari orde AR( $p$ ) mengidentifikasi plot ACF dan PACF dari data yang sudah stasioner untuk menentukan orde  $p$ (AR) dan  $q$ (MA). Selanjutnya nilai orde  $d$  didapatkan dari jumlah *differencing* yang dilakukan.

#### 3.2.4.2 Estimasi Parameter

Setelah menentukan model awal( Sementara), parameter – parameter dari model tersebut diuji untuk menilai performa model dalam melakukan peramalan. Performa model dilihat berdasarkan nilai dari variabel probabilitas pada model. Performa model bisa dikatakan layak apabila memiliki nilai  $p\text{-value} < \alpha$ , dengan level toleransi nilai  $\alpha$  adalah 5% atau 0,05.

#### 3.2.4.3 Uji Diagnosa

Pada tahap ini dilakukan untuk memastikan model yang diestimasi sudah layak digunakan atau belum. Model yang baik dan dapat digunakan untuk peramalan adalah yang residualnya bersifat random atau memiliki residual yang relatif kecil. Pengujian ini dapat menggunakan *standard error estimate* untuk menentukan model yang terbaik. Pengujian diagnostik model juga dapat dilakukan dengan menganalisa hasil Q-Ljung-Box. Uji Q-Ljung-Box digunakan untuk mengetahui apakah nilai residual dari model yang dibentuk memenuhi asumsi white-noise(residual bersifat acak atau tidak berkorelasi). Model dikatakan lolos uji apabila nilai  $P > 0.05$  pada hasil uji Q-Ljung-Box.

Pemilihan model terbaik perlu dilakukan apabila terdapat lebih dari satu model yang layak digunakan atau telah lolos uji diagnosa, pemilihan model dapat dilakukan berdasarkan nilai *Akaike Information Criterion (AIC)* terkecil.

#### 3.2.4.4 Peramalan ARIMA

Setelah mendapatkan model terbaik, maka proses selanjutnya adalah mengimplementasikan model tersebut untuk melakukan peramalan ARIMA dan kemudian hitung nilai error atau residualnya. Pada tahapan ini dibantu dengan menggunakan tools RStudio.

#### 3.2.5 Peramalan Menggunakan RBFNN

Setelah melakukan peramalan dengan metode ARIMA, selanjutnya adalah melakukan peramalan dengan metode

RBFNN. Proses yang dilakukan pada tahap ini menggunakan perangkat lunak RStudio dengan Bahasa R.

#### 3.2.5.1 Persiapan Data

Peramalan dengan menggunakan RBFNN ini diawali dengan mempersiapkan data yang akan digunakan. Pada tahap ini data yang digunakan dilakukan proses normalisasi, pembentukan lag, dan pembagian data. Normalisasi dilakukan untuk mengubah data aktual menjadi lebih kecil dari sebelumnya, tetapi tidak menghilangkan karakteristik dari data tersebut. Selanjutnya adalah tahap pembentukan lag dimana data lag ini akan dijadikan sebagai variable input pada proses pembelajaran mesin (*machine learning*). Setelah itu dilakukan pembagian data menjadi 2 bagian, yaitu data *training* dan data *testing*. Pada tugas akhir ini data yang digunakan dibagi dengan komposisi pembagian 70:30, dimana data residual dari hasil peramalan model ARIMA dibagi 70% untuk dijadikan data *training* dan 30% untuk dijadikan data *testing*.

#### 3.2.5.2 Pembentukan Model RBFNN

Tahap ini dilakukan untuk menentukan model RBFNN. Tahap pertama dalam menentukan model RBFNN adalah dengan menentukan variabel input, dimana input ditentukan dari jumlah lag data yang dibuat. Selanjutnya adalah dengan menentukan jumlah hidden neuron dan nilai maksimum iterasi. Hidden neuron merupakan neuron yang berada pada hidden layer. Fungsi aktivasi yang biasa digunakan dalam pemodelan dengan RBFNN adalah fungsi *Gaussian*[7]. Pembentukan model RBFNN dibangun dengan cara *trial and error* dari beberapa skenario yang direncanakan pada tugas akhir ini.

#### 3.2.5.3 Pelatihan Model

Pada tahap ini semua model yang dibangun akan dilakukan proses pelatihan dengan menggunakan data *training*.

#### 3.2.5.4 Pengujian Model

Pada tahapan ini semua model akan dilakukan proses peramalan atau pengujian model RBFNN yang telah dilatih pada tahap pelatihan model sebelumnya, dengan menggunakan data *testing* yang telah dipilih.

#### 3.2.5.5 Hitung Akurasi

Setelah melakukan pengujian terhadap model RBFNN, maka selanjutnya adalah melakukan perhitungan nilai akurasi dari hasil peramalan tersebut. Pengukuran akurasi model pada metode RBFNN dilakukan dengan menggunakan MAPE (*Mean Absolute Percentage Error*).

#### 3.2.5.6 Pemilihan Model RBFNN

Tahap ini dilakukan untuk menentukan model paling optimal dari hasil peramalan dengan data *testing*, model optimal ditentukan berdasarkan nilai MAPE terkecil.

#### 3.2.5.7 Peramalan ARIMA-RBFNN

Berdasarkan metode yang diusulkan dalam pengerjaan tugas akhir ini yaitu metode kombinasi atau gabungan antara ARIMA dan RBFNN, maka input pada metode RBFNN ini akan menggunakan data nilai error atau residual dari hasil peramalan dengan menggunakan model ARIMA terbaik. Metode hibrid ARIMA-RBFNN merupakan metode penggabungan antara metode ARIMA dengan RBFNN. Pertama data diproses menggunakan metode ARIMA untuk mendapatkan hasil peramalan dan nilai residual dari model ARIMA terbaik. Selanjutnya hasil nilai residual peramalan dengan metode ARIMA akan diproses sebagai nilai input untuk metode RBFNN[7]. Hasil peramalan ARIMA-RBFNN diperoleh dari penjumlahan antara hasil peramalan dengan metode ARIMA dan hasil peramalan residual menggunakan metode *Neural Network*(RBF).



### 3.2.6 Analisa Hasil Peramalan dan Penarikan Kesimpulan

Pada tahapan ini dilakukan analisis terhadap hasil peramalan terhadap model metode ARIMA, RBFNN, dan campuran ARIMA-RBFNN yang telah dibuat, apakah model tersebut sudah tepat untuk melakukan peramalan terhadap harga Ethereum. Analisa ini dilakukan dengan membandingkan hasil peramalan dengan data actual dari harga Ethereum, serta dapat dilakukan analisis tingkat kesalahan(*error*) dari model yang dibuat. Setelah hasil Analisa ditentukan, maka penarikan kesimpulan terhadap penelitian tugas akhir ini dapat dilakukan.

### 3.2.7 Penyusunan Buku Tugas Akhir

Setelah menyelesaikan tahapan dalam penelitian tugas akhir ini hingga mendapatkan hasil, maka selanjutnya yang akan dilakukan dalam penelitian ini adalah membuat dokumentasi dari hasil tersebut yang terdiri dari 7 bab antara lain :

- a. Bab 1 Pendahuluan  
Bab ini berisi hal-hal yang mendorong atau hal – hal yang melatarbelakangi pentingnya tugas akhir ini.
- b. Bab 2 Tinjauan Pustaka  
Bab ini berisi mengenai penelitian yang telah dilakukan sebelumnya yang berhubungan dengan pengerjaan tugas akhir dan dasar – dasar teori yang digunakan sebagai acuan atau pedoman dalam proses pengerjaan tugas akhir ini.
- c. Bab 3 Metodologi  
Bab ini menjelaskan mengenai tahapan atau alur pengerjaan yang akan dilaksanakan selama pengerjaan tugas akhir ini.
- d. Bab 4 Rancangan  
Pada bab ini akan diuraikan tentang perancangan model peramalan yang digunakan dalam tugas akhir.
- e. Bab 5 Implementasi  
Pada bab ini akan dijelaskan mengenai proses pembuatan model peramalan yang digunakan dalam tugas akhir.

- f. Bab 6 Hasil dan Pembahasan  
Bab ini akan membahas tentang hasil dari pembuatan model peramalan yang dilakukan. Hasil yang dijelaskan berupa hasil uji coba dari model yang digunakan dalam peramalan.
- g. Bab 7 Kesimpulan dan Saran  
Bab ini akan membahas tentang kesimpulan dari proses pengerjaan tugas akhir dan saran yang dapat diberikan untuk pengembangan yang lebih baik.

## **BAB IV PERANCANGAN**

Bab ini memperkenalkan rancangan penelitian tugas akhir yang menggunakan metode ARIMA, RBFNN, dan kombinasi metode ARIMA dan RBFNN untuk membuat model peramalan. Bab ini meliputi proses persiapan data dan Analisa model ARIMA, RBFNN, *Hybrid* ARIMA-RBFNN, dan Hasil Peramalan.

### **4.1 Persiapan Data**

Pada tahapan ini akan dilakukan proses pengumpulan data dan pengolahan data awal hingga data tersebut dapat digunakan dalam pengerjaan tugas akhir.

#### 4.1.1 Pengumpulan data

Bahan penelitian yang digunakan dalam tugas akhir ini adalah data historis harga Ethereum dalam kurun waktu 1 Januari 2017 hingga 31 Desember 2019 yang didapat dari website <https://www.investing.com>. Data yang digunakan berupa data harian.

#### 4.1.2 Menentukan Data Pelatihan dan Pengujian

Dalam tahapan ini dilakukan proses rekap data dalam Ms. Excel, kemudian melakukan pembagian data menjadi data pelatihan (*training set*) dan data pengujian (*testing set*) dengan perbandingan 70:30, dimana 70% data digunakan sebagai data latih dan 30 % data digunakan sebagai data uji.

### **4.2 Peramalan ARIMA**

Pada bagian ini akan diuraikan rancangan metode ARIMA dalam melakukan peramalan dengan data harga Ethereum.

#### 4.2.1 Identifikasi Model ARIMA

Tahap pertama dalam permodelan ARIMA adalah melakukan identifikasi model. Tahapan yang dilakukan dalam proses ini

adalah melakukan uji stationer data, transformasi data, differencing data, dan plot ACF dan PACF.

#### 4.2.1.1 Uji Stationer Ragam

Pada tahap ini akan dilakukan uji stationer data terhadap varian dengan menggunakan metode Box Cox transformation pada aplikasi Minitab. Pengujian ini dilihat berdasarkan pada nilai *rounded value*. Jika nilai pada *rounded value* = 1 maka data telah stationer terhadap varian, namun apabila nilai *rounded value*  $\neq$  1 maka data yang digunakan tidak stationer terhadap varian dan perlu ditransformasi agar data menjadi stationer terhadap varian.

#### 4.2.1.2 Transformasi Log

Pada tahap ini dilakukan proses transformasi apabila data yang digunakan belum stationer terhadap ragam. Transformasi pada penelitian ini dapat dilakukan dengan bantuan Box Cox Transformasi. Persamaan transformasi untuk setiap nilai lambda dapat dilihat pada tabel 4.1. Transformasi pada tugas akhir ini dilakukan dengan bantuan *tools* Rstudio.

**Tabel 4.1 Formula Transformasi**

$\lambda$	Transformasi
-1	$1/X_t$
-0.5	$1/\sqrt{X_t}$
0	$\text{Log}(X_t)$
0.5	$\sqrt{X_t}$
1	Tidak ditransformasi

#### 4.2.1.3 Uji Stationer Rataan

Setelah data telah stationer dalam ragam, proses selanjutnya adalah melakukan uji stationer rata-rata pada data yang telah stationer dalam ragam. Pengujian stationer rata-rata dapat dilakukan dengan menggunakan *Augmented Dickey-Fuller Test*, dimana data dikatakan stationer dalam rata-rata apabila nilai

probabilitas  $\leq 0,05$ . Apabila data tidak stationer dalam rata-rata maka perlu dilakukan differencing data hingga stationer dalam rata-rata.

#### 4.2.1.4 Differencing Data

Proses differensiasi / *differencing* perlu dilakukan apabila data tidak stationer terhadap rata-rata. Proses *differencing* pada tugas akhir ini dilakukan dengan bantuan Rstudio. Pada tahap ini, apabila nilai probabilitas atau p-value  $\leq 0,05$  maka data dianggap telah stationer. Jika data masih belum stationer setelah dilakukan differencing pertama, maka perlu dilanjutkan proses differencing hingga data menjadi stationer. Apabila data telah stationer pada differencing level pertama, data masih dapat di *differencing* lagi hingga level kedua dengan tujuan untuk memperbanyak kemungkinan model ARIMA yang dapat digunakan.

#### 4.2.1.5 Plot Grafik ACF dan PACF

Pada tahap ini, untuk mengidentifikasi model ARIMA akan dilakukan plot Grafik Autocorrelation Function (ACF) dan Partial Autocorrelation Function (PACF) pada data yang sudah stationer terhadap ragam maupun rata-rata. Pembentukan plot ACF dan PACF dapat mempermudah identifikasi komponen-komponen model ARIMA untuk menentukan ordo AR dan MA.

#### 4.2.2 Estimasi Parameter Model

Setelah menentukan model awal (sementara), parameter-parameter dari model tersebut diuji untuk menilai performa model dalam melakukan peramalan. Performa model dilihat berdasarkan nilai dari variabel probabilitas pada model. Performa model bisa dikatakan layak apabila memiliki nilai  $p < \alpha$ , dengan level toleransi nilai  $\alpha$  adalah 5% atau 0,05. Pengujian signifikansi parameter dengan menggunakan data pelatihan yang telah ditransformasi agar stationer dibantu dengan menggunakan program RStudio.

#### 4.2.3 Uji Diagnosa

Uji diagnostik parameter dilakukan untuk menguji apakah model ARIMA sudah layak dari residual model. Uji diagnostik dapat dilakukan dengan melakukan uji Q-Ljung Box. Uji Q-Ljung Box digunakan untuk mengetahui apakah nilai residual dapat memenuhi asumsi white noise(residual tidak berkorelasi). Sebuah model dianggap lulus uji keacakan sisaan apabila nilai residualnya tidak berkorelasi(white noise). Pengujian ini dapat dilakukan dengan melihat nilai Probabilitas dari hasil Uji LjungBox yang ada pada RStudio. Model dikatakan lolos uji keacakan apabila probabilitas seluruh variabel  $\geq 0,05$ .

#### 4.2.4 Pemilihan Model Terbaik

Pada tahap ini akan dilakukan proses pemilihan model terbaik, apabila ada lebih dari satu model yang lolos uji pada tahap sebelumnya. Penentuan Model ARIMA terbaik bisa dilakukan dengan membandingkan nilai AIC dari model – model yang lolos uji Diagnosa, model terbaik dipilih berdasarkan nilai AIC terkecil.

#### 4.2.5 Peramalan ARIMA

Pada tahap ini akan dilakukan peramalan dengan menggunakan model ARIMA terbaik yang sebelumnya telah dilakukan proses pengujian dan dipilih sebagai model terbaik. Proses peramalan dilakukan dengan bantuan *tools* RStudio.

#### 4.2.6 Analisis Hasil Peramalan ARIMA

Pada tahap ini akan dilakukan pengukuran tingkat akurasi dari hasil peramalan dengan menggunakan MAPE, dimana dengan menghitung selisih antara data aktual dengan hasil peramalan sehingga akan didapatkan nilai akurasi dari peramalan. Selain melakukan Analisa tingkat keakuratan dapat juga dilakukan uji pengamatan pada pola model peramalan dalam mengikuti pola dari data aktual. Pengujian menggunakan Microsoft Excel.

#### 4.2.6.1 Detransformasi data

Data yang sebelumnya telah ditransformasi perlu di kembalikan nilainya. Hal ini dapat dilakukan dengan menggunakan fungsi eksponensial dalam Microsoft Excel dengan persamaan eksponensial atau fungsi  $\exp(x)$  dimana  $x$  adalah data hasil transformasi. Dapat juga menggunakan fungsi  $InvBoxCox()$  pada RStudio.

#### 4.2.6.2 Validasi Akurasi Model

Tahap ini dilakukan perhitungan nilai MAPE dengan menghitung perbandingan antara data aktual dan hasil peramalan untuk menguji keakuratan peramalan dari model terbaik yang sudah didapatkan dengan menggunakan Microsoft Excel. Pengujian dilakukan menggunakan training data.

#### 4.2.6.3 Uji Coba Model

Model terbaik yang sudah didapatkan juga dilakukan pengujian keakuratan dengan menggunakan testing data untuk melihat apakah model terbaik bisa diterapkan pada data asing atau lain.

### 4.3 Peramalan RBFNN

Tahap selanjutnya adalah melakukan permodelan dengan metode RBFNN yang akan diolah menggunakan *tools* RStudio.

#### 4.3.1 Normalisasi Data

Pada penelitian ini dilakukan normalisasi data dengan tujuan untuk mentransformasi atau mengubah data agar memiliki rentang nilai yang lebih kecil. Normalisasi yang dilakukan dengan tipe 0\_1, dimana hasil normalisasi akan diperoleh data dengan *range* 0 sampai dengan 1. Persamaan normalisasi yang digunakan adalah :

$$X' = \frac{x - \min(X)}{\max(X) - \min(X)} \quad (4-1)$$

#### 4.3.2 Pembentukan Lag

Pada permodelan RBFNN dengan data *univariate time series* harga Ethereum, input data adalah data masa lalu (lag time) dan target atau output data adalah data masa kini. Data *univariate time series* yang telah dinormalisasi perlu dilakukan pembagian lag yang akan digunakan sebagai input pada metode RBFNN. Pada penelitian ini akan dibuat 9 skenario input RBFNN dari lag data yang telah dibuat, pemilihan jumlah lag optimal ditentukan berdasarkan nilai MAPE terkecil dari hasil pengujian. Tabel 4.2 menunjukkan skenario pembentukan lag sebagai input RBFNN.

**Tabel 4.2 Skenario Pembentukan Lag Data**

<b>Skenario</b>	<b>Deskripsi</b>
Lag-2	Penambahan 2 lag sebagai input RBFNN
Lag-3	Penambahan 3 lag sebagai input RBFNN
Lag-4	Penambahan 4 lag sebagai input RBFNN
Lag-5	Penambahan 5 lag sebagai input RBFNN
Lag-6	Penambahan 6 lag sebagai input RBFNN
Lag-7	Penambahan 7 lag sebagai input RBFNN
Lag-8	Penambahan 8 lag sebagai input RBFNN
Lag-9	Penambahan 9 lag sebagai input RBFNN
Lag-10	Penambahan 10 lag sebagai input RBFNN

#### 4.3.3 Pembagian Data

Pada tahap ini data yang digunakan dibagi menjadi 2 bagian, yaitu data pelatihan dan data pengujian. Pada tugas akhir ini data yang digunakan dibagi dengan komposisi pembagian 70:30, dimana data residual dari hasil peramalan model ARIMA dibagi 70% untuk dijadikan data pelatihan dan 30% untuk dijadikan data pengujian. Data pelatihan digunakan untuk pembentukan model dari arsitektur RBFNN yang dirancang, sedangkan data pengujian digunakan untuk mengevaluasi model-model peramalan.



#### 4.3.4 Pembentukan Arsitektur RBFNN

Pada tahap ini akan dilakukan pembentukan arsitektur model RBFNN. Rancangan arsitektur ini terdiri dari penentuan jumlah input layer, neuron pada hidden layer, output layer serta parameter model yang akan digunakan. Pada input layer jumlah variabel input sesuai dengan jumlah pembentukan lag signifikan. Lapisan pada Hidden layer berjumlah satu lapisan. Lapisan ini mengandung neuron dengan fungsi aktivasi berbasis radial, misalnya yang digunakan pada penelitian ini adalah fungsi *Gaussian*. Jumlah neuron pada hidden layer (*hidden neuron*) ditentukan dengan cara *trial & error*. Skenario pembentukan model arsitektur RBFNN dapat dilihat pada tabel 4.3

**Tabel 4.3 Tabel skenario perancangan model RBFNN**

Parameter	Jumlah	Deskripsi
Input Layer	2-10 Lag (Input)	<i>Trial and Error</i>
Hidden Layer	2 – 20 Neuron	<i>Trial and Error</i>
Output Layer	1	Hasil Ramalan
Epoch	[100, 200 , 500, 1000]	<i>Trial and Error</i>

#### 4.3.5 Pelatihan Model RBFNN

Pada tahap ini semua model yang dibangun akan dilakukan proses pelatihan terhadap model RBFNN yang telah dibuat dengan data *training*

#### 4.3.6 Pengujian Model RBFNN

Pada tahapan ini semua model akan dilakukan proses pengujian model RBFNN yang telah dilatih pada tahap pelatihan model sebelumnya, dengan menggunakan data *testing* yang telah dipilih

#### 4.3.7 Hitung Akurasi Model RBFNN

Pada tahap ini akan dilakukan pengukuran akurasi dari model RBFNN. Tahap ini terdiri dari denormalisasi data dan pengukuran akurasi.

##### 4.3.7.1 Denormalisasi Data

Setelah melakukan pengujian terhadap model RBFNN, maka selanjutnya adalah melakukan perhitungan nilai akurasi dari hasil peramalan tersebut. Sebelum pengukuran dilakukan data hasil peramalan perlu dinormalisasi untuk mengembalikan nilai sesuai dengan nilai asli atau sebenarnya.

##### 4.3.7.2 Pengukuran Akurasi

Setelah data didenormalisasi, maka selanjutnya adalah melakukan perhitungan nilai akurasi dari hasil peramalan tersebut. Pengukuran akurasi pada proses peramalan dengan metode RBFNN menggunakan MAPE (*Mean Absolute Percentage Error*).

#### 4.3.8 Pemilihan model RBFNN

Tahap pemilihan model merupakan tahap untuk menentukan model terbaik atau paling optimal yang akan digunakan untuk melakukan peramalan. Pemilihan model RBFNN terbaik pada penelitian ini dengan membandingkan nilai MAPE, dimana model terbaik memiliki nilai MAPE terkecil.

### 4.4 Peramalan ARIMA-RBFNN

Pada tahap peramalan menggunakan *hybrid* ARIMA-RBFNN, Langkah pertama yang dilakukan adalah melakukan peramalan dengan ARIMA, selanjutnya memodelkan nilai residual dari peramalan ARIMA menggunakan metode RBFNN. Hasil dari peramalan ARIMA dan permodelan nilai residual ARIMA oleh RBFNN dengan model terbaik disimpan ke dalam file Excel. Proses selanjutnya adalah menentukan nilai peramalan yang dihasilkan dengan menggunakan metode gabungan ARIMA-

RBFNN dengan menjumlahkan hasil peramalan ARIMA dan peramalan residual ARIMA oleh RBFNN. Persamaan model matematika dari peramalan ARIMA-RBFNN seperti pada rumus (4-2) dan (4-3) :

$$y_t = L_t + N_t \quad (4-2)$$

$$\text{forecast hybrid (ARIMA - RBFNN)} = \text{nilai Peramalan ARIMA} + \text{nilai peramalan residual ARIMA (RBFNN)} \quad (4-3)$$

#### **4.5 Analisa Hasil Peramalan**

Setelah mendapatkan model hybrid ARIMA-RBFNN terbaik maka dilakukan analisis hasil peramalan untuk mengetahui keakuratan atau akurasi model peramalan dengan data harga Ethereum. Perhitungan nilai akurasi model hybrid ARIMA-RBFNN dilakukan menggunakan MAPE dengan cara menghitung selisih antara data aktual dengan hasil peramalan menggunakan program Microsoft Excel.

*“Halaman ini sengaja dikosongkan”*

## **BAB V**

### **IMPLEMENTASI**

Pada bab ini, akan dijelaskan mengenai pelaksanaan penelitian dan pembuatan model peramalan yang digunakan untuk tugas akhir.

#### **5.1 Pengumpulan Data**

Pada tahap ini dilakukan pengumpulan data yang dibutuhkan dalam proses pengerjaan tugas akhir. Proses pengumpulan data dilakukan dengan mengunduh data nilai harga Ethereum dari website [investing.com](https://www.investing.com). Dari proses tersebut didapatkan data harga Ethereum dengan kurs USD dengan rentang waktu harian dari tanggal 01 Januari 2017 hingga 31 Desember 2019 yang berjumlah sebanyak 1095 data. Sebelum melakukan Langkah selanjutnya data yang didapatkan diolah dalam Microsoft Excel untuk menghasilkan data yang akan digunakan pada pengerjaan Tugas akhir. Variabel yang akan digunakan dalam pengujian adalah variable Price. Tabel 5.1 menunjukkan data nilai harga Ethereum untuk 10 periode awal dan akhir.

#### **5.2 Menentukan Data Pelatihan dan Pengujian**

Pada tahap ini, data akan dilakukan pembagian data Harga Ethereum menjadi 2 (dua) bagian atau kelompok data, yaitu data latih (*training*) dan data uji (*testing*). Komposisi pembagian data training dan testing adalah 70 : 30. Total data yang didapatkan berjumlah 1095 data (01 Januari 2017 – 31 Desember 2019). Jumlah data train yang digunakan sebanyak 767 (01 Januari 2017 - 06 Februari 2019) dan data test sebanyak 328 (07 Februari 2019 – 31 Desember 2019) dari total 1095 data harga Ethereum. Data train digunakan untuk menentukan model peramalan yang akan digunakan, sedangkan data test digunakan untuk melakukan validasi dan pengecekan tingkat keakuratan model dengan menggunakan data baru / asing.

**Tabel 5.1 Data Harga Ethereum 10 Periode Awal dan Akhir**

Index No.	Date	Price
1	1/1/2017	8.2
2	1/2/2017	8.37
3	1/3/2017	9.74
4	1/4/2017	11
5	1/5/2017	10.15
6	1/6/2017	10.08
7	1/7/2017	9.84
8	1/8/2017	10.28
9	1/9/2017	10.26
10	1/1/2017	8.2
.....	.....	.....
1086	12/22/2019	132.19
1087	12/23/2019	128.08
1088	12/24/2019	128.05
1089	12/25/2019	125.24
1090	12/26/2019	125.93
1091	12/27/2019	126.41
1092	12/28/2019	128.35
1093	12/29/2019	134.4
1094	12/30/2019	131.83
1095	12/31/2019	129.21

### 5.3 Pembuatan dan Penerapan Model ARIMA

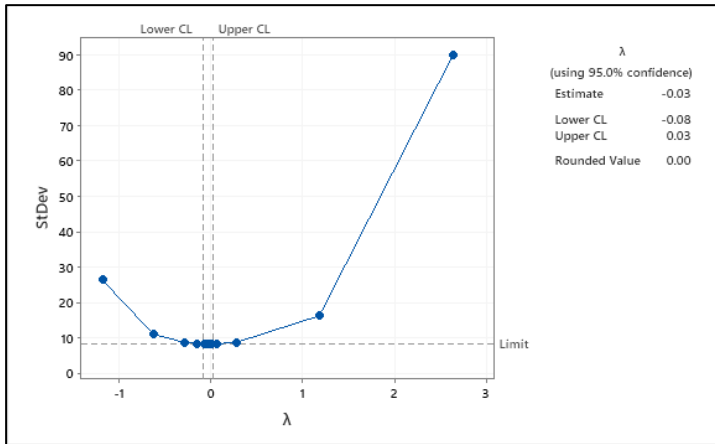
Dalam tahap pembuatan model peramalan akan digunakan pembagian data training dan testing. Data train digunakan untuk membangun dan menentukan model peramalan, sedangkan data testing digunakan untuk menguji performa model yang diperoleh.

#### 5.3.1 Identifikasi Model ARIMA

Pada subbab ini akan diuraikan mengenai proses identifikasi model ARIMA dari tahap pengujian stationer, transformasi, differencing dan melakukan plot ACF dan PACF untuk menentukan parameter model.

### 5.3.1.1 Uji Stationer Ragam

Uji stasioner Ragam dilakukan pada data pelatihan harga Ethereum. Gambar 5.1 menunjukkan hasil uji stasioner dalam ragam terhadap data pelatihan harga Ethereum.



**Gambar 5.1 Uji Stationer Ragam pada data pengujian Harga Ethereum**

Berdasarkan hasil dari nilai pengujian stationer ragam dengan menggunakan metode *box – cox transformation* pada program minitab, dapat disimpulkan bahwa data pelatihan yang digunakan masih belum stationer terhadap ragam karena memiliki nilai *rounded value* = 0.00 atau  $\neq 1$ . Untuk itu data pelatihan perlu ditransformasi agar mencapai nilai *rounded value* = 1 dan stationer terhadap ragam untuk dapat bisa diteruskan ke tahap selanjutnya.

### 5.3.1.2 Transformasi Data

Pada tahap ini akan dilakukan proses transformasi data apabila data sebelumnya, yaitu data pelatihan tidak stationer terhadap ragam. Transformasi data dilakukan berdasarkan tabel 5.2

**Tabel 5.2 Formula Transformasi**

$\lambda$	Transformasi
-1	$1/X_t$
-0.5	$1/\sqrt{X_t}$
0	$\text{Log}(X_t)$
0.5	$\sqrt{X_t}$
1	Tidak ditransformasi

Berdasarkan pada tabel 5.2, data pengujian memiliki lambda ( $\lambda$ ) = 0 , maka perlu dilakukan transformasi. Dimana data aktual yang digunakan merupakan data pengujian. Transformasi data yang digunakan pada penelitian ini menggunakan *BoxCox Transform* dari *library "forecast"* pada *Rstudio* seperti pada Kode 5.1.

```
#transformasi
train_log <- BoxCox(train, lambda=0)
```

**Kode 5.1 Transformasi Box Cox**

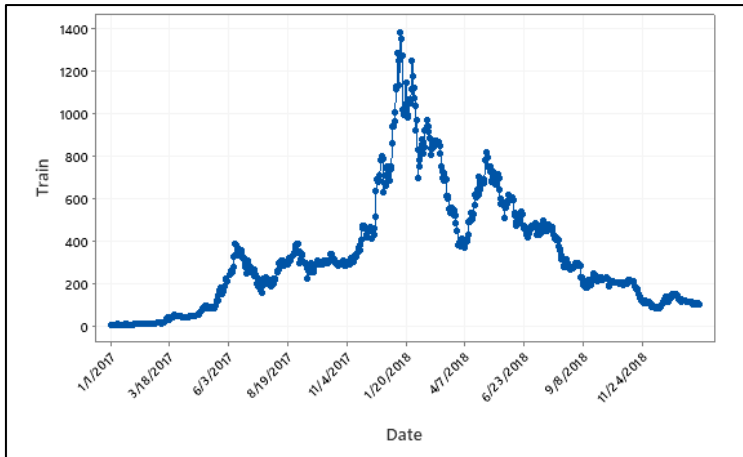
Keterangan Kode 5.1 dijelaskan pada Tabel 5.3

**Tabel 5.3 Penjelasan Kode 5.1**

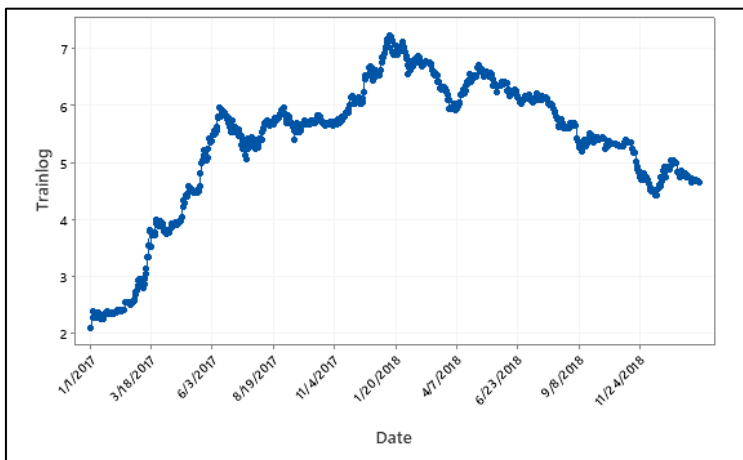
Atribut	Penjelasan
BoxCox()	Fungsi untuk melakukan transformasi BoxCox
lambda	Nilai Lambda yang diperlukan untuk melakukan transformasi

Gambar 5.2 menunjukkan grafik data pengujian sebelum transformasi dan Gambar 5.3 menunjukkan grafik data pengujian setelah transformasi.





**Gambar 5.2** Grafik data latih harga Ethereum



**Gambar 5.3** Grafik data latih setelah ditransformasi

### 5.3.1.3 Uji Stationer Rataan

Uji stasioneritas rataaan dilakukan pada data latih yang telah ditransformasi. Pengujian dilakukan dengan menggunakan fungsi *Augmented Dickey-Fuller(ADF) test* pada *library*

“*tseries*” dari RStudio. Gambar 5.4 menunjukkan hasil uji stasioner rataaan.

```
> adf.test(train_log) #cek stationer

      Augmented Dickey-Fuller Test

data: train_log
Dickey-Fuller = -1.4926, Lag order = 9, p-value = 0.7931
alternative hypothesis: stationary
```

**Gambar 5.4 Hasil Uji Stationer**

Dari hasil uji stationer rataaan dengan ADF test pada gambar 5.4, menunjukkan bahwa nilai Probabilitas  $> 0.05$ . Hal ini menandakan bahwa data latih yang telah ditransformasi tidak stationer dalam rataaan, sehingga tahap selanjutnya adalah dengan melakukan proses *differencing* pada data latih.

#### 5.3.1.4 Differencing Data

*Differencing* dilakukan apabila data yang digunakan pada tahap sebelumnya tidak stationer terhadap *mean* atau rataaan. Pada tahap ini proses *differencing* dilakukan dengan menggunakan fungsi *diff* pada RStudio. Gambar 5.5 menunjukkan hasil *differencing* level 1 pada data latih harga Ethereum yang sudah di transformasi.

```
> diff1 = diff(train_log)
> adf.test(diff1)

      Augmented Dickey-Fuller Test

data: diff1
Dickey-Fuller = -8.4131, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary
```

**Gambar 5.5 Hasil Differencing level 1 pada data pelatihan**

Berdasarkan hasil uji stasioner(ADF) pada data pelatihan seperti yang dapat dilihat pada gambar 5.5, menunjukkan bahwa nilai Probabilitas  $\leq 0.05$ . Sehingga dapat disimpulkan bahwa data latih telah stationer terhadap rataaan pada level

*differencing* pertama. Untuk menambah kemungkinan model ARIMA dilakukan *differencing* kedua dan pengujian stationer seperti pada gambar 5.6.

```
> diff2 <- diff(train_log, differences = 2)
> adf.test(diff2)

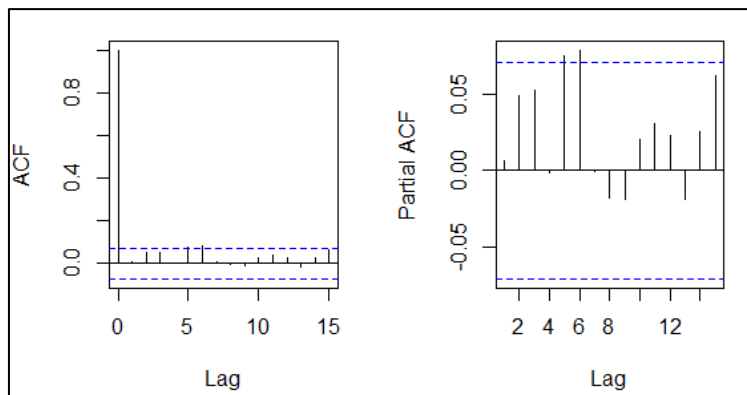
      Augmented Dickey-Fuller Test

data:  diff2
Dickey-Fuller = -14.49, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary
```

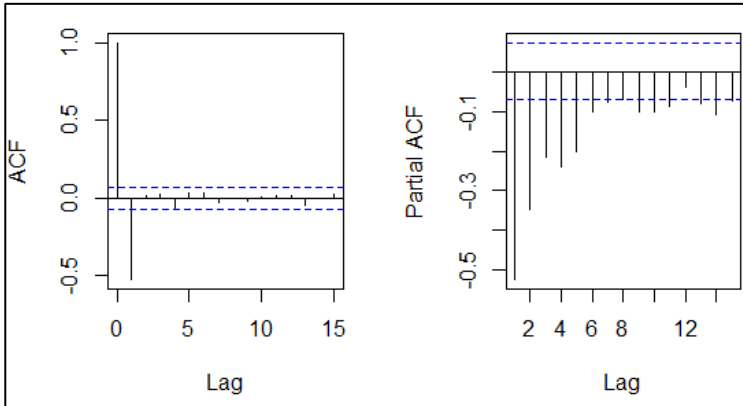
**Gambar 5.6 Hasil Differencing level 2 pada data pelatihan**

### 5.3.1.5 Plot Grafik ACF dan PACF

Identifikasi awal model ARIMA dilakukan pada data pelatihan yang telah stasioner dengan melihat plot grafik ACF PACF. Komponen model Autoregresi (AR) didapatkan dari grafik PACF, sedangkan komponen Moving Average (MA) didapatkan dari grafik ACF. Kemudian komponen *differencing* didapatkan dengan menghitung jumlah *differencing* yang telah dilakukan pada data pelatihan. Gambar 5.7 menunjukkan hasil plot grafik ACF dan PACF pada level *differencing* pertama. Gambar 5.8 menunjukkan hasil plot grafik ACF dan PACF pada level *differencing* kedua.



**Gambar 5.7 Grafik ACF dan PACF pada data latih, diff = 1**



Gambar 5.8 Grafik ACF dan PACF pada data latih, diff = 2

### 5.3.2 Estimasi Paramater Model

Pada tahap ini akan dilakukan estimasi parameter model arima berdasarkan pada hasil plot grafik ACF dan PACF, yang selanjutnya akan dilakukan proses pengujian signifikansi parameter model ARIMA yang telah di estimasi. Tabel 5.4 menunjukkan hasil identifikasi parameter atau komponen model ARIMA menggunakan hasil *differencing* 1.

Setelah melakukan identifikasi parameter selanjutnya adalah melakukan proses uji signifikansi parameter untuk menguji kelayakan model ARIMA yang sudah diestimasi. Kriteria pada pengujian signifikan ini adalah dimana parameter model dikatakan signifikan Jika nilai probabilitas seluruh model  $\leq 0.05$ . Pengujian signifikansi dilakukan dengan melihat hasil fungsi *coefest()* dari *package lmtest* pada model yang ingin diuji signifikansi. Gambar 5.9 menunjukkan contoh model ARIMA yang lolos uji signifikansi

**Tabel 5.4 Estimasi Parameter data harga Ethereum**

Variabel	Level Differencing	Estimasi Parameter
Price	1	ARIMA(0,1,1), ARIMA(0,1,2), ARIMA(0,1,3), ARIMA(0,1,4), ARIMA(0,1,5), ARIMA(0,1,6), ARIMA(1,1,0), ARIMA(1,1,1), ARIMA(1,1,2), ARIMA(1,1,3), ARIMA(1,1,4), ARIMA(1,1,5), ARIMA(1,1,6), ARIMA(2,1,0), ARIMA(2,1,1), ARIMA(2,1,2), ARIMA(2,1,3), ARIMA(2,1,4), ARIMA(2,1,5), ARIMA(2,1,6), ARIMA(3,1,0), ARIMA(3,1,1), ARIMA(3,1,2), ARIMA(3,1,3), ARIMA(3,1,4), ARIMA(3,1,5), ARIMA(3,1,6), ARIMA(4,1,0), ARIMA(4,1,1), ARIMA(4,1,2), ARIMA(4,1,3), ARIMA(4,1,4), ARIMA(4,1,5), ARIMA(4,1,6), ARIMA(5,1,0), ARIMA(5,1,1), ARIMA(5,1,2), ARIMA(5,1,3), ARIMA(5,1,4), ARIMA(5,1,5), ARIMA(5,1,6), ARIMA(6,1,0), ARIMA(6,1,1), ARIMA(6,1,2), ARIMA(6,1,3), ARIMA(6,1,4), ARIMA(6,1,5), ARIMA(6,1,6)
	2	ARIMA(0,2,1)

```
> coefstest(arima(train_log, order=c(1,1,1)))
z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.995205   0.011050  90.068 < 2.2e-16 ***
ma1 -0.983292   0.020745 -47.400 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

**Gambar 5.9 Hasil Uji Signifikansi ARIMA(1,1,1)**

Berdasarkan pada gambar 5.9 nilai probabilitas model  $\leq 0.05$ , Sehingga dapat disimpulkan bahwa model ARIMA(1,1,1) telah

lolos uji signifikansi karena telah memenuhi kriteria pengujian dan dapat dilanjutkan ke proses selanjutnya.

### 5.3.3 Uji Diagnosa

Dari seluruh model yang lulus uji signifikan selanjutnya maka dilakukan uji diagnostik. Uji ini dilakukan untuk menguji kelayakan model ARIMA yang telah didapatkan sebelumnya apakah memenuhi asumsi white noise(residual tidak berkorelasi atau tidak mempunyai pola tertentu). Proses pengujian dengan menggunakan fungsi *checkresiduals()* dari *package forecast*. Model dikatakan lolos uji apabila hasil uji *L-Jung-Box* memenuhi asumsi white noise dengan kriteria *p-value*  $> 0,05$ . Gambar 5.10 menunjukkan contoh model yang lolos uji diagnosa.

```

Ljung-Box test

data: Residuals from ARIMA(1,1,1)
Q* = 8.9825, df = 8, p-value = 0.3438

```

Gambar 5.10 Hasil Uji Diagnostik ARIMA(1,1,1)

Hasil pengujian pada gambar 5.8 memperlihatkan bahwa nilai *p-value*  $> 0.05$ , maka model ARIMA(1,1,1) lolos uji diagnostic karena residualnya bersifat acak dan memenuhi asumsi white noise.

### 5.3.4 Pemilihan Model Terbaik

Pemilihan model ARIMA dilakukan karena terdapat lebih dari satu model ARIMA yang lulus uji diagnostic dengan menggunakan data pelatihan. Pemilihan model dilakukan dengan memilih nilai terkecil *Akaike Information Criterion* (AIC) dari masing – masing model yang lulus uji diagnostic. Tabel 5.5 menunjukkan perbandingan nilai AIC dari semua model yang lulus pada tahapan uji diagnostic.

**Tabel 5.5 Tabel Perbandingan Nilai AIC model ARIMA**

Model	AIC
ARIMA(1,1,1)	<b>-2045,175</b>
ARIMA(0,2,1)	-2040,057

Berdasarkan tabel 5.5, dapat disimpulkan bahwa model ARIMA(1,1,1) adalah model yang terbaik karena memiliki nilai AIC terkecil jika dibandingkan dengan model lainnya, dengan nilai AIC = -2045,175.

### 5.3.5 Peramalan ARIMA

Setelah mendapatkan model terbaik, tahap selanjutnya adalah melakukan peramalan dengan model tersebut untuk mengetahui tingkat akurasi dari hasil peramalannya, pada tahap ini masih menggunakan data pelatihan.

### 5.3.6 Analisa Hasil Peramalan ARIMA

Pada subbab ini akan dilakukan perhitungan akurasi model ARIMA menggunakan MAPE terhadap masing – masing data *training*, *testing*, dan keseluruhan.

#### 5.3.6.1 Detransformasi Data

Setelah melakukan peramalan model dengan data training, maka akan didapatkan data hasil peramalan. Sebelum melakukan Analisa hasil peramalan data peramalan harus di normalkan terlebih dahulu atau dikembalikan pada nilai yang sebenarnya karena sebelumnya sudah ditransformasi. Pengembalian nilai prediksi dapat menggunakan fungsi *InvBoxCox()* pada *package forecast* seperti pada Kode 5.2.

```
#detransform
fitARIMA <- data.matrix(InvBoxCox(fitted(model), 0))
```

**Kode 5.2 Inverse Box Cox**

Keterangan Kode 5.2. dijelaskan pada Tabel 5.6

**Tabel 5.6 Penjelasan Kode 5.2**

<b>Atribut</b>	<b>Penjelasan</b>
InvBoxCox()	Fungsi untuk mengembalikan bentuk data transformasi
Lambda	Nilai lambda transformasi awal (BoxCox, lambda = 0)

### 5.3.6.2 Validasi Akurasi Model

Model terbaik yang sudah didapatkan pada tahap sebelumnya dihitung tingkat keakuratan hasil peramalannya dengan menggunakan nilai MAPE. Pengujian dilakukan menggunakan data pelatihan dan menggunakan bantuan *package MLmetrics*.

Dari hasil pengujian dengan data latih menunjukkan bahwa model ARIMA(1,1,1) memiliki nilai MAPE 4,444 % yang menunjukkan bahwa model ARIMA(1,1,1) memiliki tingkat keakuratan yang sangat baik karena memiliki nilai MAPE < 10%.

### 5.3.6.3 Hasil Uji Coba Model

Model terbaik yang sudah didapatkan pada proses sebelumnya juga diuji dengan menggunakan data testing untuk melihat apakah model dapat diterapkan pada data yang lain. Selanjutnya melakukan perhitungan nilai MAPE untuk mengetahui tingkat keakuratan peramalan dari model tersebut.

Dari hasil pengujian dengan data latih menunjukkan bahwa model ARIMA(1,1,1) memiliki nilai MAPE 2,822% yang menunjukkan bahwa model ARIMA(1,1,1) memiliki tingkat keakuratan yang sangat baik karena memiliki nilai MAPE < 10% serta dapat diterapkan pada data yang lain.

## 5.4 Penerapan dan Pembuatan Model RBFNN

Pada tahap ini akan diuraikan tahapan implementasi peramalan dengan metode RBFNN. Setelah melakukan peramalan dengan metode ARIMA, selanjutnya akan dilakukan peramalan metode RBFNN. Pada tahap ini pembentukan model RBFNN akan menggunakan *tools Rstudio* dengan bahasa pemrograman R



### 5.4.1 Normalisasi

Normalisasi data yang digunakan pada penelitian ini menggunakan fungsi `normalizeData()` dari *library RSNNS* dengan tipe normalisasi `0_1`.

### 5.4.2 Pembentukan Lag

Data yang telah dinormalisasi selanjutnya akan dibuat menjadi beberapa lag data yang akan digunakan sebagai input dari metode RBFNN. Pembentukan lag pada penelitian ini dibuat menjadi 10 lag dengan skenario penggunaan lag sebagai input seperti pada Tabel 4.3. Pembentukan lag menggunakan fungsi `lag()` pada *library dplyr* pada RStudio.

```
#lag
lag1 <- lag(dt_norm, 1)
lag2 <- lag(dt_norm, 2)
lag3 <- lag(dt_norm, 3)
lag4 <- lag(dt_norm, 4)
lag5 <- lag(dt_norm, 5)
lag6 <- lag(dt_norm, 6)
lag7 <- lag(dt_norm, 7)
lag8 <- lag(dt_norm, 8)
lag9 <- lag(dt_norm, 9)
lag10 <- lag(dt_norm, 10)
target <- data.frame(dt_norm)
```

**Kode 5.3 Pembentukan Lag**

Setelah indentifikasi variabel lag dilakukan selanjutnya adalah proses pembentukan variabel input dan target(output) sesuai dengan jumlah lag data yang digunakan sebagai variabel input. Kode 5.4 Menunjukkan pembentukan variabel input dan target dengan lag 2.

```
#####lag2
frame <- data.frame(lag2, lag1, target)
df <- na.omit(frame)
colnames(df) <- c("lag-2", "lag-1", "target")
```

**Kode 5.4 Pembentukan Variabel Input dan Target RBFNN**

Keterangan Kode 5.3. dan 5.4 akan dijelaskan pada Tabel 5.7

**Tabel 5.7 Penjelasan Kode 5.3 dan 5.4**

<b>Atribut</b>	<b>Penjelasan</b>
Lag()	Fungsi untuk membuat lag data, sesuai jumlah lag
Na.omit()	Menghilangkan baris yang mempunyai nilai Na
normalizeData()	Fungsi untuk menormalisasi data
type	Tipe normalisasi
0_1	Nilai normalisasi berapa pada rentang 0 – 1. Menggunakan fungsi MinMax, nilai minimum menjadi 0 dan nilai maksimum menjadi 1

### 5.4.3 Pembagian Data

Pada tahap ini data yang digunakan dibagi menjadi 2 bagian, yaitu data *training* dan data *testing*. Pada tugas akhir ini data yang digunakan dibagi dengan komposisi pembagian 70:30, dimana data aktual dibagi 70% untuk dijadikan data *training* dan 30% untuk dijadikan data *testing*. Proses pembagian data dilakukan secara manual di *RStudio*.

### 5.4.4 Pembentukan Model RBFNN

Pembentukan model RBFNN pada penelitian ini menggunakan library “*RSNNS*” (*Stuttgart Neural Network Simulator*). Model RBFNN dapat dibuat dengan fungsi *rbf()* pada library “*RSNNS*” dengan skenario pembentukan model seperti pada tabel 4.3., dengan nilai *default* pada parameter lain untuk fungsi *rbf()*. Kode 5.4 menunjukkan pembentukan model RBFNN.

```
model <- rbf(x_train, y_train,
            size=n, maxit = maxit[i])
```

**Kode 5.5 Pembentukan Model RBFNN**

Keterangan kode 5.5 akan dijelaskan pada Tabel 5.8.

Tabel 5.8 Penjelasan Kode 5.5

Atribut	Penjelasan
rbf()	Fungsi untuk melakukan <i>modelling</i> dengan metode RBFNN
X_train	Variable input atau independen
y_train	Variable output atau dependen
size	Jumlah <i>neuron</i> pada <i>hidden layer</i> / jumlah <i>hidden neuron</i>
maxit	Jumlah maksimum iterasi dalam proses pembentukan model

#### 5.4.5 Pengujian Model

Pada tahapan ini semua model akan dilakukan proses pengujian model RBFNN yang telah dilatih sebelumnya, dengan menggunakan data *testing*.

```
y_pred <- predict(model, x_test)
```

Kode 5.6 Pengujian Model RBFNN

Keterangan kode 5.6 akan dijelaskan pada Tabel 5.9.

Tabel 5.9 Penjelasan Kode 5.6

Atribut	Penjelasan
Predict()	Fungsi yang digunakan untuk menghasilkan prediksi
model	Model RBFNN
X_test	Variabel input dari data testing

#### 5.4.6 Hitung Akurasi Peramalan

Tahap Perhitungan nilai akurasi terdiri dari denormalisasi data pada hasil peramalan dan pengukuran akurasi model RBFNN.

### 5.4.6.1 Denormalisasi Data

Denormalisasi data dilakukan terhadap hasil peramalan / variabel prediksi dan variabel output(target) agar hasil pengukuran menunjukkan nilai asli atau sebenarnya. Proses denormalisasi menggunakan fungsi *denormalizeData()* dari *library RSNNS*.

```
#denorm
y_pred_denorm <- denormalizeData(y_pred, getNormParameters(dt_norm))
y_test_denorm <- denormalizeData(y_test, getNormParameters(dt_norm))
```

#### Kode 5.7 Denormalisasi Hasil Peramalan

Keterangan kode 5.7 akan dijelaskan pada Tabel 5.10.

**Tabel 5.10 Penjelasan Kode 5.7**

Atribut	Penjelasan
denormalizeData()	Fungsi yang digunakan untuk mengembalikan nilai normalisasi
getNormParameters()	Fungsi untuk mengambil nilai parameter pada saat normalisasi (nilai MixMax).

### 5.4.6.2 Pengukuran Akurasi

Setelah hasil peramalan dengan data *testing* didenormalisasi, selanjutnya akan dilakukan perhitungan nilai akurasi peramalan model dengan MAPE. Perhitungan MAPE setiap model menggunakan fungsi *MAPE()* dari *library "MLmetrics"*.

### 5.4.7 Pemilihan Model Terbaik

Setelah dilakukan pengukuran nilai akurasi model. Selanjutnya menentukan model optimal berdasarkan pada nilai MAPE terkecil dari setiap model. Pada penelitian ini semua nilai MAPE dari setiap skenario model akan ditampung kedalam sebuah list yang nanti akan disimpan kedalam file *xlsx*. Setelah semua skenario model telah disimpan maka selanjutnya akan ditentukan model terbaik berdasarkan nilai MAPE terkecil. Pemilihan model ini untuk menentukan model terbaik untuk kasus peramalan data harga Ethereum berdasarkan dari hasil peramalan data pengujian pada penelitian ini.

## 5.5 Penerapan dan Pembuatan Model ARIMA-RBFNN

Pada tahap ini akan diuraikan tahapan implementasi peramalan dengan metode hybrid ARIMA-RBFNN. Setelah melakukan peramalan dengan metode ARIMA, selanjutnya akan dilakukan peramalan residual ARIMA dengan metode RBFNN. Pada tahap ini pembentukan model RBFNN akan menggunakan *tools Rstudio* dengan bahasa pemrograman *R*.

Langkah yang dilakukan pada peramalan residual ARIMA dengan metode RBFNN sama dengan langkah pada subbab 5.4. Setelah mendapatkan model terbaik RBFNN. Hasil peramalan residual ARIMA dengan model RBFNN terbaik akan dijumlahkan dengan nilai peramalan ARIMA. Proses ini menggunakan bantuan Microsoft Excel, dengan menyimpan nilai peramalan ARIMA dan peramalan residual ARIMA dengan metode RBFNN kedalam file Excel. Selanjutnya akan dilakukan proses perhitungan untuk menentukan nilai forecast yang dihasilkan dari metode hybrid ARIMA-RBFNN. Proses penyimpanan hasil peramalan dengan metode ARIMA dan RBFNN menggunakan fungsi *write.xlsx()* pada *library "openxlsx"*.

## 5.6 Analisa Hasil Peramalan

Analisa Hasil Peramalan ARIMA-RBFNN dilakukan dengan menghitung nilai MAPE dengan menggunakan *tools* Microsoft Excel. Selain itu, dilakukan plot data untuk mengetahui perbandingan pola antara data aktual dan prediksi untuk menganalisa pola hasil peramalan. Selain itu juga melakukan perbandingan nilai akurasi peramalan pada setiap metode yang dilakukan pada penelitian ini.

*“Halaman ini sengaja dikosongkan”*

## **BAB VI**

### **HASIL DAN PEMBAHASAN**

Pada bab ini akan berisikan hasil dan pembahasan setelah melakukan implementasi. Hasil yang akan dijelaskan adalah hasil uji model ARIMA, RBFNN, Hybrid ARIMA-RBFNN, serta hasil peramalannya.

#### **6.1 Hasil Uji Model ARIMA**

Pada bagian ini akan diuraikan hasil dari proses permodelan ARIMA dengan data harga Ethereum.

##### **6.1.1 Hasil Uji Signifikansi**

Pada bagian ini akan ditampilkan hasil uji signifikansi pada data pelatihan harga Ethereum. Uji signifikansi ini dilakukan berdasarkan pada jumlah model yang didapatkan pada estimasi parameter.

**Tabel 6.1 Hasil uji signifikansi parameter model ARIMA**

Model ARIMA	Uji Signifikansi
ARIMA(1,1,1)	Lolos
ARIMA(0,2,1)	Lolos

Tabel 6.1 Menampilkan model ARIMA yang lolos hasil uji signifikansi untuk data training harga Ethereum. Berdasarkan pada tabel 6.1 model yang lolos uji berjumlah 2 model dari total keseluruhan sebanyak 49 estimasi model.

##### **6.1.2 Hasil Uji Diagnostik**

Pada bagian ini akan ditampilkan hasil uji diagnostik pada model yang lolos uji signifikansi untuk data pelatihan harga Ethereum.

**Tabel 6.2 Hasil uji diagnostik model ARIMA**

Model ARIMA	Uji Diagnostik
-------------	----------------

ARIMA(1,1,1)	Lolos
ARIMA(0,2,1)	Lolos

Tabel 6.2 Menampilkan hasil pengujian diagnostik untuk data training harga Ethereum. Berdasarkan pada tabel 6.2 seluruh model lulus uji Diagnostik

### 6.1.3 Hasil Validasi Akurasi Model dan Uji Coba Model

Pada subbab ini akan diuraikan mengenai hasil validasi akurasi model terbaik ARIMA dan pengujian model dengan data test.

#### 6.1.3.1 Validasi Akurasi Model

Pada tahap ini, uji coba model dilakukan dengan menggunakan data pelatihan (*training*). Pengujian dilakukan untuk mengetahui tingkat keakuratan hasil peramalan dari model ARIMA terbaik. Gambar 6.1 merupakan hasil uji coba peramalan dengan menggunakan model ARIMA (1,1,1) yang terpilih sebagai model terbaik.



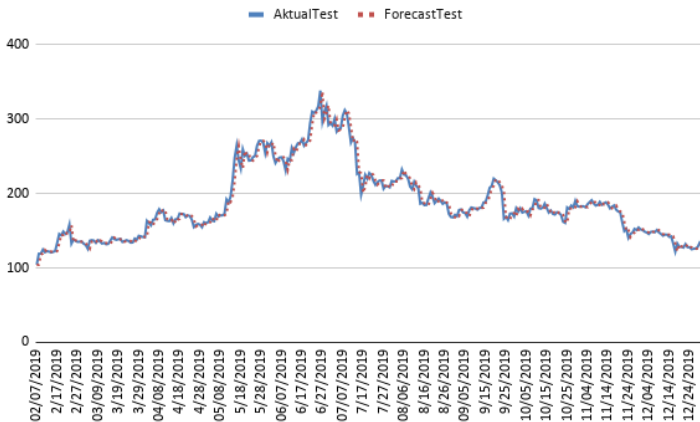
Gambar 6.1 Grafik uji coba model ARIMA(1,1,1) pada data train

Gambar 6.1 menampilkan hasil pengujian model ARIMA(1,1,1) pada data training harga Ethereum. Berdasarkan pada grafik, dapat dilihat bahwa hasil peramalan mampu mengikuti pola data actual.



### 6.1.3.2 Uji Coba Model

Pada tahap ini dilakukan pengujian model dengan menggunakan data uji (*testing*). Pengujian dilakukan untuk mengetahui tingkat keakuratan hasil peramalan dari model yang dipilih dengan menggunakan data baru. Gambar 6.2 merupakan hasil uji validasi model dengan menggunakan model ARIMA(1,1,1) yang terpilih sebagai model terbaik.

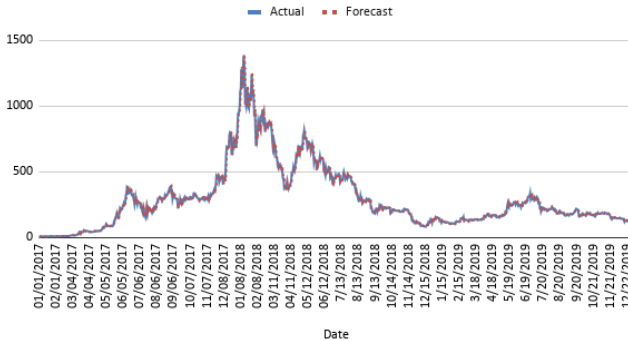


**Gambar 6.2** Grafik uji coba model ARIMA(1,1,1) pada data Test

Gambar 6.2 menampilkan hasil peramalan menggunakan model ARIMA(1,1,1) pada data testing harga Ethereum. Berdasarkan pada gambar grafik, dapat dilihat bahwa hasil peramalan mampu mengikuti pola data aktual.

### 6.1.4 Hasil dan Analisis Peramalan Data Aktual

Pada tahap ini, uji coba model dilakukan dengan keseluruhan data *train* dan *testing*, pengujian dilakukan untuk mengetahui tingkat keakuratan peramalan model ARIMA(1,1,1) dengan keseluruhan data aktual. Gambar 6.3 merupakan hasil dari peramalan dengan menggunakan model ARIMA(1,1,1) dengan keseluruhan data aktual.



**Gambar 6.3** Grafik hasil uji coba model ARIMA data harga Ethereum

Berdasarkan pada gambar 6.3 pola yang dihasilkan dari peramalan data aktual dengan ARIMA(1,1,1) dapat mengikuti pola data aktual. Dapat disimpulkan bahwa model ARIMA(1,1,1) dapat digunakan untuk melakukan peramalan harga Ethereum.

### 6.1.5 Perbandingan MAPE

Bagian ini akan diperlihatkan hasil perhitungan MAPE dari hasil peramalan dengan menggunakan data *train*, data *test*, dan keseluruhan data, untuk mengetahui tingkat keakuratan model pada setiap data dan juga melakukan pengujian apakah model ARIMA yang dipilih mampu melakukan peramalan dengan hasil yang akurat apabila menggunakan data baru. Hasil perhitungan MAPE model ARIMA(1,1,1) dapat dilihat pada tabel 6.3.

**Tabel 6.3** Hasil Perhitungan MAPE model ARIMA(1,1,1)

Data	MAPE (%)
Train	4,444129
Test	2,822823
Aktual( Train & Test)	3,95848

Berdasarkan pada hasil pada tabel 6.3, diketahui bahwa MAPE model yang didapatkan pada data *training*, data *testing*, dan keseluruhan data < 10% yang menandakan bahwa data prediksi

atau peramalan yang dihasilkan oleh model ARIMA(1,1,1) memiliki hasil yang akurat. Model ARIMA(1,1,1) dapat digunakan untuk melakukan peramalan data harga Ethereum.

## 6.2 Hasil Permodelan RBFNN

Pada subbab ini akan diuraikan mengenai hasil permodelan RBFNN.

### 6.2.1 Pengujian Model RBFNN

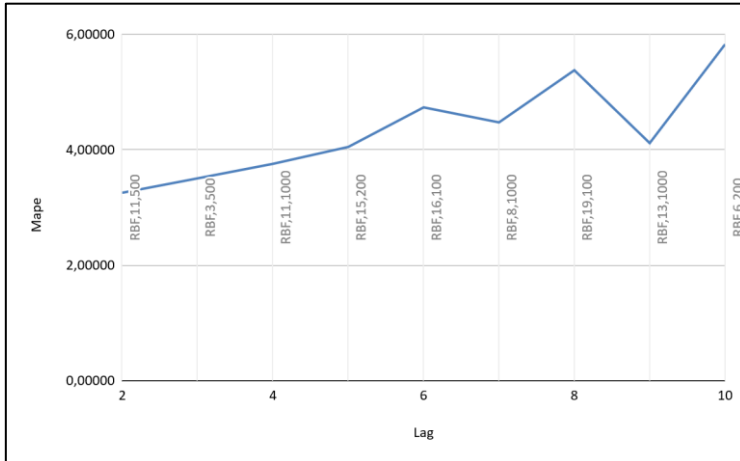
Pengujian dilakukan terhadap model RBFNN dengan menggunakan data *testing*. Tujuan pengujian untuk mendapatkan model RBFNN yang paling optimal, berdasarkan pada nilai akurasi terbaik. Model optimal dipilih berdasarkan nilai MAPE terkecil.

Hasil pengujian dapat dilihat pada tabel 6.4, Untuk nilai Pengujian seluruh model skenario RBFNN dapat dilihat pada **LAMPIRAN E**.

**Tabel 6.4 Hasil Uji Coba Model RBFNN**

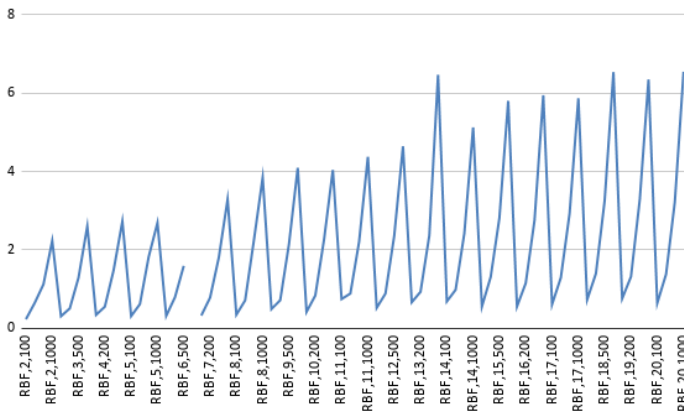
Input	Hidden Neuron	Iterasi	MAPE	Running Time
<b>2</b>	<b>11</b>	<b>500</b>	<b>3,25565</b>	<b>2,2151</b>
3	3	500	3,50518	0,9584
4	11	1000	3,75683	4,0266
5	15	200	4,04977	1,1208
6	16	100	4,73511	0,5623
7	8	1000	4,47486	3,4859
8	19	100	5,37863	0,6699
9	13	1000	4,11630	4,7194
10	6	200	5,82671	0,6011

Berdasarkan pada tabel 6.4, model RBFNN yang memiliki nilai MAPE terkecil adalah dengan 2 variabel input, 11 *hidden neuron* dan maksimum iterasi sebanyak 500. Gambar 6.4 menunjukkan grafik perbandingan nilai MAPE paling kecil pada setiap periode(variabel input sesuai jumlah lag).



**Gambar 6.4** Grafik Pengujian model RBFNN

Gambar 6.5 menampilkan running time pada RBFNN(2,11,1), dimana berdasarkan pada gambar tersebut dapat dikatakan waktu yang dibutuhkan untuk memodelkan RBFNN terikat pada jumlah neuron dan banyak iterasi, semakin banyak jumlah neuron dan iterasi maka waktu yang diperlukan untuk memodelkan lebih lama.



**Gambar 6.5** Running Time model RBFNN(2,11,1)

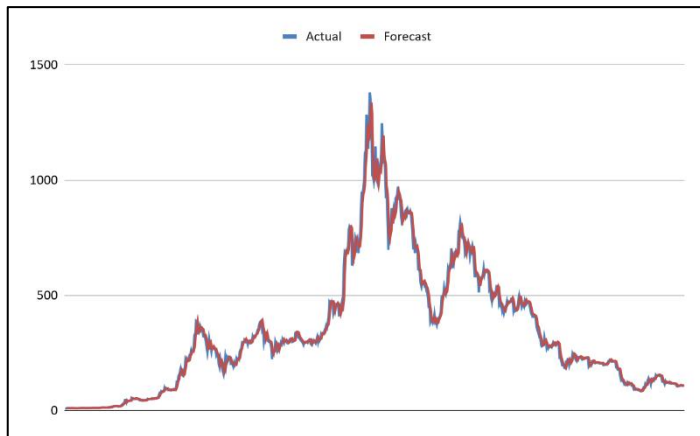
### 6.2.2 Hasil Peramalan RBFNN

Pada tahap ini setelah mendapatkan nilai model RBFNN yang paling optimal, selanjutnya melakukan analisis hasil peramalan RBFNN(2,11,1).

**Tabel 6.5 Hasil Forecast RBFNN**

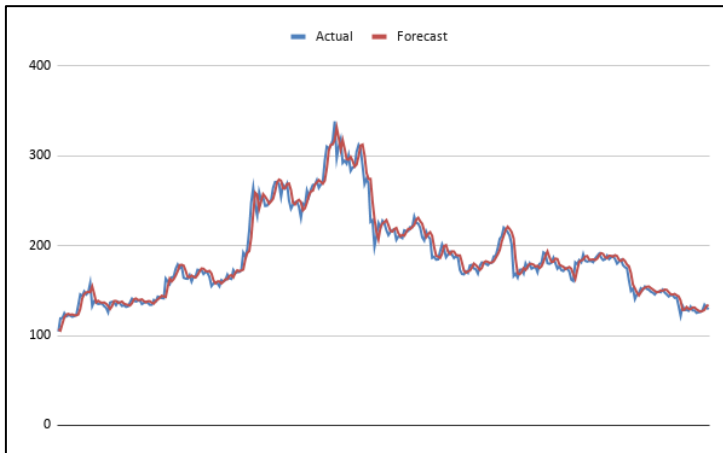
Date	Aktual	Forecast RBFNN	Residu
1/10/2017	8,2		
1/11/2017	8,37		
1/12/2017	9,74	8,27882	1,46118
1/13/2017	11	9,08013	1,91987
1/14/2017	10,15	10,40146	-0,25146
.....		.....	
12/27/2019	126,41	126,79192	-0,38192
12/28/2019	128,35	127,37802	0,97198
12/29/2019	134,4	128,63328	5,76672
12/30/2019	131,83	132,73923	-0,90923
12/31/2019	129,21	134,34772	-5,13772

Tabel 6.6 menunjukkan hasil peramalan dengan model RBFNN pada 5 periode awal dan 5 periode akhir.



**Gambar 6.6 Grafik Hasil Peramalan Data Pelatihan RBFNN(2,11,1)**

Gambar 6.6 merupakan grafik perbandingan data aktual dan data peramalan pada data pelatihan Harga Ethereum dengan model RBFNN paling optimal yaitu dengan 2 input, 11 neuron, dan 1 output. Berdasarkan pada gambar 6.5, dapat dilihat bahwa hasil peramalan mampu mengikuti pola data aktual.



**Gambar 6.7 Grafik Hasil Peramalan Data Pengujian RBFNN(2,11,1)**

Gambar 6.7 merupakan grafik perbandingan data aktual dan data peramalan pada data pengujian Harga Ethereum dengan model RBFNN(2,11,1). Berdasarkan pada gambar 6.5, dapat dilihat bahwa hasil peramalan mampu mengikuti pola data aktual. Berdasarkan pada hasil peramalan dengan model RBFNN(2,11,1) didapatkan nilai MAPE seperti pada tabel 6.6.

**Tabel 6.6 Hasil Perhitungan MAPE model RBFNN(2,11,1)**

Data	MAPE (%)
Train	4,93348
Test	3,25565
Aktual( Train & Test)	4,42998

### 6.3 Hasil Permodelan ARIMA-RBFNN

Pada subbab ini akan diuraikan mengenai hasil permodelan ARIMA-RBFNN. Dimana Permodelan ARIMA-RBFNN pada

penelitian ini menggunakan ARIMA(1,1,1) dan model peramalan RBFNN optimal untuk residual ARIMA(1,1,1).

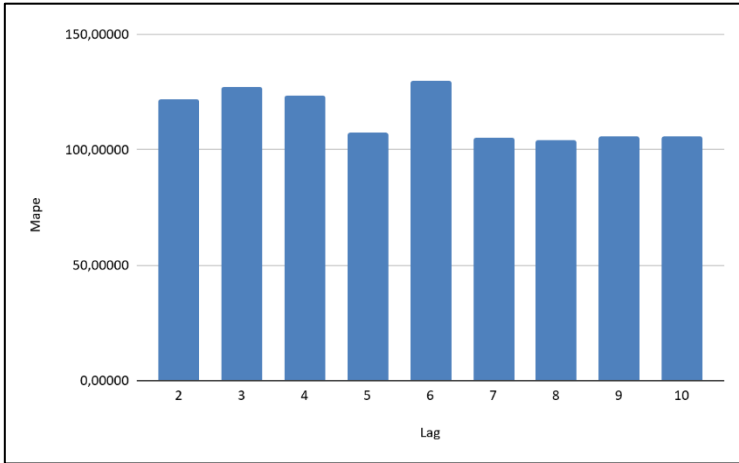
### 6.3.1 Pengujian Model RBFNN

Pada subbab ini akan diuraikan hasil pengujian model peramalan residual ARIMA dengan RBFNN. Tujuan pengujian untuk mendapatkan model RBFNN yang paling optimal untuk meramalkan residual ARIMA, berdasarkan pada nilai MAPE terkecil. Hasil pengujian dapat dilihat pada tabel 6.7, Untuk nilai Pengujian seluruh model skenario RBFNN dapat dilihat pada **LAMPIRAN H**.

**Tabel 6.7 Hasil Pengujian model RBFNN pada Residual ARIMA**

Input	Hidden Neuron	Iterasi	MAPE	MAPE Hybrid	Running Time
<b>2</b>	10	100	121,68709	2,82699	0,275315
3	13	1000	127,05211	2,82764	2,955681
4	13	200	123,49040	2,83523	0,616485
5	10	200	107,38624	2,81895	0,517058
6	20	500	130,13826	2,87028	6,157705
7	5	500	105,03878	2,78805	0,388001
<b>8</b>	<b>14</b>	<b>1000</b>	<b>104,30438</b>	<b>2,78189</b>	<b>3,279257</b>
9	8	200	106,06165	2,80939	0,458855
<b>10</b>	10	100	105,55607	2,82699	0,275315

Berdasarkan pada tabel 6.7, model RBFNN yang memiliki nilai MAPE terkecil adalah dengan 8 variabel input, 14 *hidden neuron* dan maksimum iterasi sebanyak 1000. Gambar 6.8 menunjukkan grafik perbandingan nilai MAPE paling kecil pada setiap periode(variabel input sesuai jumlah lag). MAPE Hybrid merupakan nilai MAPE pengujian dari hasil peramalan ARIMA yang telah ditambahkan dengan peramalan Residual dari RBFNN.



Gambar 6.8 Diagram Perbandingan MAPE Residual dengan RBFNN

### 6.3.2 Hasil Peramalan ARIMA-RBFNN

Pada tahap ini nilai peramalan ARIMA-RBFNN didapatkan dengan menambahkan nilai peramalan metode ARIMA(1,1,1) dan nilai peramalan Residu ARIMA oleh RBFNN(8,14,11). Hasil penjumlahan ini lah yang merupakan hasil peramalan metode *hybrid* ARIMA-RBFNN.

Tabel 6.8 Hasil Peramalan ARIMA-RBFNN

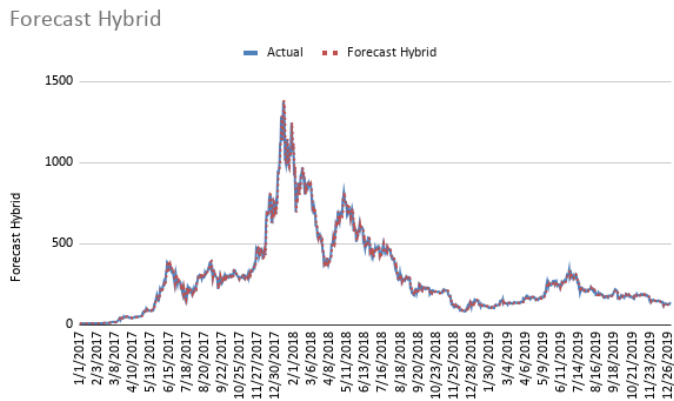
Date	Forecast ARIMA	Forecast Residu (RBFNN)	Forecast ARIMA-RBFNN
1/9/2017	10,30542	-0,093966	10,237326
1/10/2017	10,65877	-0,070150	10,241077
1/11/2017	9,82082	-0,017053	10,646361
1/12/2017	9,86026	-0,050532	9,774510
1/13/2017	9,69638	-0,048092	9,816126
.....		.....	.....
12/27/2019	125,52618	-0,000949	125,801107
12/28/2019	126,01985	-0,131446	126,287042
12/29/2019	127,98846	0,071474	128,280824
12/30/2019	134,11201	-0,209237	133,803318
12/31/2019	131,51978	-0,159458	131,976033



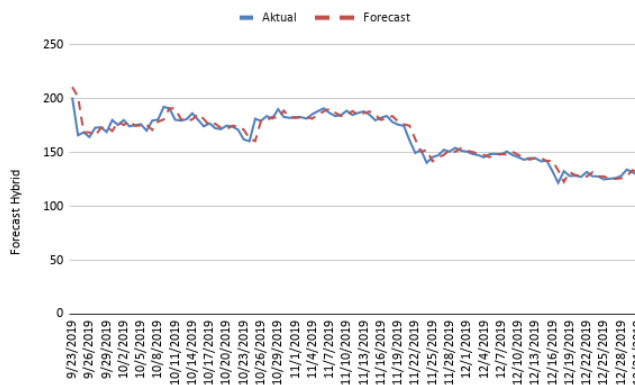
Tabel 6.8 menunjukkan hasil peramalan data harga Ethereum dengan metode *hybrid* ARIMA RBFNN.

### 6.3.3 Analisa Hasil Peramalan ARIMA-RBFNN

Dalam sub bab ini akan dilakukan Analisa hasil peramalan data aktual harga Ethereum dengan menggunakan metode *hybrid* ARIMA-RBFNN. Gambar 6.9 dan 6.10 merupakan grafik hasil peramalan data aktual dengan metode ARIMA-RBFNN



**Gambar 6.9** Grafik Hasil Peramalan data harga Ethereum dengan ARIMA-RBFNN



**Gambar 6.10** Grafik hasil peramalan ARIMA-RBFNN, pada 100 periode terakhir

Berdasarkan pada gambar 6.4 dan 6.5, pola peramalan data aktual harga Ethereum dengan metode ARIMA-RBFNN dapat mengikuti pola data aktual, maka dapat disimpulkan bahwa peramalan data harga Ethereum dapat dilakukan dengan metode *hybrid* ARIMA-RBFNN. Berdasarkan pada hasil peramalan dengan model RBFNN(2,11,1) didapatkan nilai MAPE seperti pada tabel 6.9.

**Tabel 6.9 Hasil Perhitungan MAPE model ARIMA-RBFNN**

Data	MAPE (%)
Train	4,44231
Test	2,80416
Aktual( Train & Test)	3,95161

#### **6.4 Analisa Hasil Peramalan ARIMA, RBFNN, ARIMA-RBFNN**

Pada sub bab ini akan dilakukan perbandingan hasil peramalan menggunakan metode ARIMA dengan metode *hybrid* ARIMA-RBFNN. Perbandingan hasil peramalan dilakukan dengan tujuan untuk menemukan model yang mampu memberikan hasil terbaik. Perbandingan metode ini didasarkan pada perbandingan nilai akurasi setiap metode, dimana pada penelitian ini perhitungan nilai akurasi peramalan menggunakan MAPE. Metode dengan nilai MAPE terkecil merupakan metode yang memiliki tingkat keakuratan peramalan yang lebih baik. Dalam melakukan perbandingan ini, hasil peramalan yang dibandingkan adalah peramalan dengan menggunakan data pelatihan, pengujian dan seluruh data aktual harga Ethereum.

**Tabel 6.10 Hasil Perbandingan Nilai MAPE pada Data Pelatihan**

Metode	Model	MAPE (%)
ARIMA	ARIMA(1,1,1)	4,44413
RBFNN	RBFNN(2,11,1)	4,93348
ARIMA-RBFNN	ARIMA(1,1,1) RBFNN(8,14,1)	4,44231

Pada tabel 6.10 dilakukan perbandingan MAPE pada data pelatihan antara model ARIMA, RBFNN, dan *Hybrid* ARIMA-RBFNN. Hasil perbandingan MAPE didapatkan bahwa, menggunakan metode *Hybrid* ARIMA-RBFNN memiliki nilai MAPE yang lebih kecil jika dibandingkan dengan metode individu, yaitu sebesar 4.442%.

**Tabel 6.11 Hasil Perbandingan Nilai MAPE pada data Pengujian**

Metode	Model	MAPE (%)
ARIMA	ARIMA(1,1,1)	2,82282
RBFNN	RBFNN(2,11,1)	3,25565
ARIMA-RBFNN	ARIMA(1,1,1) RBFNN(8,14,1)	2,80416

Pada tabel 6.11 dilakukan perbandingan MAPE pada data pengujian antara model ARIMA, RBFNN, dan *Hybrid* ARIMA-RBFNN. Hasil perbandingan MAPE didapatkan bahwa, menggunakan metode *Hybrid* ARIMA-RBFNN memiliki nilai MAPE yang lebih kecil jika dibandingkan dengan metode individu, yaitu sebesar 2,804%.

**Tabel 6.12 Hasil Perbandingan Nilai MAPE pada Data Aktual(Train & Test)**

Metode	Model	MAPE (%)
ARIMA	ARIMA(1,1,1)	3,95848
RBFNN	RBFNN(2,11,1)	4,42998
ARIMA-RBFNN	ARIMA(1,1,1) RBFNN(8,14,1)	3,95161

Pada tabel 6.12 dilakukan perbandingan MAPE pada data keseluruhan(*train* dan *testing*) antara model ARIMA, RBFNN, dan *Hybrid* ARIMA-RBFNN. Hasil perbandingan MAPE didapatkan bahwa, menggunakan metode *Hybrid* ARIMA-RBFNN memiliki nilai MAPE yang lebih kecil jika dibandingkan dengan metode individu, yaitu sebesar 3,951%.

Dapat disimpulkan bahwa ketiga metode memiliki tingkat akurasi yang baik karena memiliki nilai MAPE dibawah 10%.. Berdasarkan hasil perbandingan dari data pelatihan, pengujian, dan keseluruhan data didapatkan bahwa hasil peramalan dengan model Hybrid ARIMA-RBFNN merupakan metode peramalan yang terbaik bila dibandingkan dengan model individu untuk kasus peramalan harga Ethereum dengan nilai MAPE 2,80% pada data pengujian.

## BAB VII KESIMPULAN DAN SARAN

Pada bab ini akan dibahas mengenai kesimpulan dari semua proses penelitian yang telah dilakukan dan saran yang diusulkan untuk pengembangan yang lebih baik dimasa mendatang.

### 7.1 Kesimpulan

Berdasarkan hasil penelitian pada tugas akhir ini, maka kesimpulan yang dapat diambil ialah sebagai berikut:

1. Didapatkan model terbaik ARIMA(1,1,1) untuk peramalan harga Ethereum. Hasil pemodelan ARIMA dapat dikatakan sangat baik karena memiliki nilai MAPE yang kecil yaitu 4,44% pada data training, 2,82% pada data testing, dan 3,9584% untuk keseluruhan data aktual.
2. Didapatkan model terbaik RBFNN dengan 2 input variabel, 11 *hidden neuron* dan 1 output untuk peramalan harga Ethereum. Hasil pemodelan ARIMA dapat dikatakan sangat baik karena memiliki nilai MAPE yang kecil yaitu 4,44% pada data training, 2,82% pada data testing, dan 3,9584% untuk keseluruhan data aktual.
3. Model terbaik *hybrid* ARIMA-RBFNN yang digunakan untuk meramal harga Ethereum adalah dengan model ARIMA(1,1,1) dan model RBFNN dengan jumlah input sebanyak 8, *hidden neuron* 14, dan output 1. Pembentukan iterasi model RBFNN baik dilakukan pada tingkat maksimum 1000. Nilai MAPE dari model terbaik peramalan ARIMA-RBFNN dengan menggunakan data pelatihan ialah 4.442%, dengan data pengujian ialah 2,804%, dan aktual harga Ethereum ialah 3,951%. Hasil ini terbilang sangat baik karena nilai MAPE < 10%.
4. Peramalan dengan menggunakan metode *hybrid* ARIMA-RBFNN pada studi kasus ini, berdasarkan pada nilai MAPE yang dihasilkan memiliki tingkat akurasi

yang lebih baik dari pada metode individu ARIMA dan RBFNN.

## 7.2 Saran

Berdasarkan hasil penelitian pada tugas akhir ini, terdapat beberapa saran yang diharapkan dapat bermanfaat bagi penelitian selanjutnya, saran yang diusulkan adalah sebagai berikut :

1. Dalam Tugas Akhir ini data yang digunakan merupakan data harga Ethereum dengan rentang waktu harian. Untuk penelitian selanjutnya dapat menggunakan varian rentang waktu jam, mingguan, atau bulanan.
2. Pembagian data *training* dan data *test* pada penelitian ini hanya menggunakan perbandingan 70 : 30, untuk penelitian selanjutnya dapat menggunakan variasi pembagian data yang berbeda.
3. Memperbanyak variasi parameter model RBFNN untuk peramalan harga Ethereum seperti menambah jumlah parameter neuron dan menambah skenario pembentukan lag.
4. Dalam pengerjaan tugas akhir dengan menggunakan data yang sama dapat dilakukan dengan mencoba metode hybrid lain, seperti ARIMA-ANFIS, ARIMA-ANN, dan lainnya.

## DAFTAR PUSTAKA

- [1] "Selamat Datang Era Mata Uang Virtual, Cryptocurrency," Kumparan, 2017. [Online]. Available: <https://kumparan.com/kumparantech/selamat-datang-era-mata-uang-virtual-cryptocurrency>. [Accessed 3 Nopember 2019].
- [2] "Ethereum Terus Catatkan Rally Rekor, Bitcoin Anjlok," Investing, 2018. [Online]. Available: <https://id.investing.com/news/cryptocurrency-news/ethereum-terus-catatkan-rally-rekor-bitcoin-anjlok-373375>. [Accessed 3 Nopember 2019].
- [3] "Investasi Bitcoin cs, Siapa Takut?," detikcom, 2019. [Online]. Available: [https://finance.detik.com/advertorial-news-block/d-4779955/investasi-bitcoin-cs-siapa-takut?\\_ga=2.63693676.25892907.1573613765-313553053.1566822508](https://finance.detik.com/advertorial-news-block/d-4779955/investasi-bitcoin-cs-siapa-takut?_ga=2.63693676.25892907.1573613765-313553053.1566822508). [Accessed 3 Nopember 2019].
- [4] D. T. Wiyanti dan R. Pulungan, "Peramalan Deret Waktu Menggunakan Model Fungsi Basis Radial (RBF) dan Auto Regressive Integrated Moving Average (ARIMA)," *Jurnal MIPA*, vol. 35, no. 2, pp. 175-182, 2012.
- [5] M. Chen and N. Narwal, "Predicting Price Changes in Ethereum," 2017. [Online]. Available: <https://www.semanticscholar.org/paper/Predicting-Price-Changes-in-Ethereum-Chen-Narwal/ceff65e02b2b9b6b181cfc956350351b8e284a01#citing-papers>. [Accessed 3 Nopember 2019].
- [6] Habibi, M. Y., Riksakomara, Edwin, "Peramalan Harga Garam Konsumsi Menggunakan Artificial Neural Network Feedforward-Backpropagation (Studi Kasus : PT. Garam Mas, Rembang, Jawa Tengah)", *Jurnal Teknik ITS*, Vol. 6, No. 2, 2017
- [7] A. Hikmah, A. Agoestanto and R. Arifudin, "Peramalan Deret Waktu Dengan Menggunakan Autoregressive (Ar), Jaringan Syaraf Tiruan Radial Basis Function (Rbf)

- Dan Hibrid Ar-Rbf Pada Inflasi Indonesia," *Unnes Journal of Mathematics*, vol. 7, no. 2, pp. 228-241, 2018.
- [8] M. Riadi, "Pengertian, Fungsi, dan Jenis-jenis peramalan (forecasting)," *Kajian Pustaka*, 2017. [Online]. Available:<https://www.kajianpustaka.com/2017/11/pengertian-fungsi-dan-jenis-peramalan-forecasting.html>. [Accessed 3 Nopember 2019].
- [9] Makridakis, "International Journal of Forecasting," p. 519, 1988.
- [10] Makridakis, Spyros, Wheelright, Steven C., dan Hyndman, Rob J. *Forecasting: Methods and Application* 3rd edition. New York, John Wiley & Sons, 1998.
- [11] Salwa , Nany, et al., "Peramalan Harga Bitcoin Menggunakan Metode Arima (Autoregressive Integrated Moving Average)", *Journal of Data Analysis*, Vol.1, No.1, p. 21-31, 2018
- [12] Razak. Abd. Fadhilah, "Load Forecasting Using Time Series Models". *Jurnal Kejuruteraan*, 21: 53-62, 2009
- [13] Rekza Nila Anityaloka and Atika Nuraini Ambarwati, "Peramalan Saham Jakarta Islamic Index Menggunakan Metode Arima Bulan Mei-Juli 2010", *Jurnal Statistika Universitas Muhammadiyah Semarang*, vol. 1, pp. 1-5, Mei 2013
- [14] R. Riza, S. Boko, P. Diyah, "Perbandingan Keakuratan Metode Autoregressive Integrated Moving Average (Arima) Dan Exponential Smoothing Pada Peramalan Penjualan Semen Di Pt. Sinar Abadi", *Jurnal Rekursif*, vol. 3, no. 1, p. 23-36, 2015.
- [15] Bangun, Rita Herawaty Br, "Penerapan Autoregressive Integrated Moving Average (Arima) Pada Peramalan Produksi Kedelai Di Sumatera Utara," *Jurnal Agribisnis Sumatera Utara*, vol. 9, no. 2, p. 90-100, 2017.
- [16] Wei, Shen., Xiaopen, Guo., Chao, Wu., Desheng, Wu. "Forecasting Stock Indices Using Radial Basis Function Neural Networks Optimized By Artificial Fish Swarm Algorithm", *Knowledge Based System*. 24, 378-385, 2011



- [17] G. P. Zhang, "Time Series Forecasting Using A Hybrid Arima And Neural Network Model", *Journal of Neurocomputing*, 50, pp.159-175, 2003
- [18] J. S. Armstrong and F. Collopy, "Error Measures For Generalizing About Forecasting Methods: Empirical Comparisons", *Int. J. Forecast*, vol. 8, no. 1, pp. 69–80, 1992.
- [19] Juan José Montaña Moreno, et all, "Using the R-MAPE Index as A Resistant Measure of Forecast Accuracy", *Psicothema*, Vol. 25 No. 4 , pp. 500-506,2013.

*“Halaman ini sengaja dikosongkan”*

## BIODATA PENULIS



Penulis lahir di Situbondo pada tanggal 29 Juli 1998 dan merupakan anak kedua dari dua bersaudara pasangan Bapak Muhammad Hasan dan Ibu Tutik Hartini. Penulis menempuh Pendidikan formal di SD Muhammadiyah Panji, SMP Negeri 1 Situbondo dan SMA Negeri 1 Situbondo.

Pada tahun 2016 penulis melanjutkan Pendidikan jenjang sarjana dengan jalur SBMPTN di Departemen Sistem Informasi, Fakultas Teknologi Elektro dan Informatika Cerdas (F-ELECTICS), Institut Teknologi Sepuluh Nopember(ITS), Surabaya dan terdaftar sebagai mahasiswa dengan NRP 05211640000091. Selama masa perkuliahan penulis aktif mengikuti organisasi mahasiswa, seperti Himpunan Mahasiswa Sistem Informasi(HMSI) selama dua periode yaitu 2017 – 2018 dan 2018 – 2019 dan Kajian Islam Sistem Informasi (KISI) 2017 – 2018.

Pada tahun keempat, penulis mengambil bidang minat Pengolahan Data. Oleh karena itu, penulis terdaftar sebagai mahasiswa tugas akhir di Laboratorium Rekayasa Data dan Intelegensi Bisnis, Departemen Sistem Informasi, ITS. Penulis dapat dihubungi melalui email [farisikram21@gmail.com](mailto:farisikram21@gmail.com)

*“Halaman ini sengaja dikosongkan”*

**LAMPIRAN A:  
HASIL TRANSFORMASI**

Periode	Data Transformasi
01/01/2017	2,10413
01/02/2017	2,12465
01/03/2017	2,27624
01/04/2017	2,39790
01/05/2017	2,31747
01/06/2017	2,31055
01/07/2017	2,28646
01/08/2017	2,33020
01/09/2017	2,32825
01/01/2017	2,36180
.....	.....
12/22/2019	4,88424
12/23/2019	4,85266
12/24/2019	4,85242
12/25/2019	4,83023
12/26/2019	4,83573
12/27/2019	4,83953
12/28/2019	4,85476
12/29/2019	4,90082
12/30/2019	4,88151
12/31/2019	4,86144

*“Halaman ini sengaja dikosongkan”*

## LAMPIRAN B: UJI SIGNIFIKANSI PARAMETER

```

> coefstest(arima(train_log, order=c(0,1,1)))
z test of coefficients:
      Estimate Std. Error z value Pr(>|z|)
ma1  0.0084825  0.0344236  0.2464  0.8054
> #ARIMA(0,1,2)
> coefstest(arima(train_log, order=c(0,1,2)))
z test of coefficients:
      Estimate Std. Error z value Pr(>|z|)
ma1  0.0039514  0.0361750  0.1092  0.913
ma2  0.0520613  0.0367002  1.4186  0.156
> #ARIMA(0,1,3)
> coefstest(arima(train_log, order=c(0,1,3)))
z test of coefficients:
      Estimate Std. Error z value Pr(>|z|)
ma1  0.0070562  0.0362153  0.1948  0.8455
ma2  0.0456644  0.0369091  1.2372  0.2160
ma3  0.0419932  0.0338364  1.2411  0.2146

```

**Gambar B. 1 Uji Signifikansi ARIMA(0,1,1), ARIMA(0,1,2), ARIMA(0,1,3)**

```

> coefstest(arima(train_log, order=c(0,1,4)))
z test of coefficients:
      Estimate Std. Error z value Pr(>|z|)
ma1  0.0077879  0.0364663  0.2136  0.8309
ma2  0.0464859  0.0370429  1.2549  0.2095
ma3  0.0419312  0.0338009  1.2405  0.2148
ma4 -0.0060753  0.0374780 -0.1621  0.8712
> #ARIMA(0,1,5)
> coefstest(arima(train_log, order=c(0,1,5)))
z test of coefficients:
      Estimate Std. Error z value Pr(>|z|)
ma1 -0.0059763  0.0364401 -0.1640  0.86973
ma2  0.0480539  0.0363855  1.3207  0.18661
ma3  0.0456761  0.0336345  1.3580  0.17446
ma4 -0.0017182  0.0366815 -0.0468  0.96264
ma5  0.0805627  0.0360583  2.2342  0.02547 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> #ARIMA(0,1,6)
> coefstest(arima(train_log, order=c(0,1,6)))
z test of coefficients:
      Estimate Std. Error z value Pr(>|z|)
ma1 -0.00089983  0.03598078 -0.0250  0.98005
ma2  0.04806489  0.03596166  1.3366  0.18137
ma3  0.05261550  0.03634173  1.4478  0.14767
ma4 -0.00068652  0.03619634 -0.0190  0.98487
ma5  0.07481896  0.03579558  2.0902  0.03660 *
ma6  0.07885303  0.03573519  2.2066  0.02734 *

```

**Gambar B. 2 Uji Signifikansi ARIMA(0,1,4), ARIMA(0,1,5), ARIMA(0,1,6)**

```

> #ARIMA(1,1,2)
> coeftest(arima(train_log, order=c(1,1,2)))

z test of coefficients:

      Estimate Std. Error  z value Pr(>|z|)
ar1  0.952387   0.045975  20.7152 <2e-16 ***
ma1 -0.955502   0.058193 -16.4196 <2e-16 ***
ma2  0.038603   0.038313   1.0076  0.3137
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> #ARIMA(1,1,3)
> coeftest(arima(train_log, order=c(1,1,3)))

z test of coefficients:

      Estimate Std. Error  z value Pr(>|z|)
ar1  0.957557   0.0515799  18.5645 <2e-16 ***
ma1 -0.9610963  0.0635688 -15.1190 <2e-16 ***
ma2  0.0453133  0.0516718   0.8769  0.3805
ma3 -0.0085515  0.0425262  -0.2011  0.8406
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> #ARIMA(1,1,4)
> coeftest(arima(train_log, order=c(1,1,4)))

z test of coefficients:

      Estimate Std. Error  z value Pr(>|z|)
ar1  0.99505120  0.00952581 104.4584 <2e-16 ***
ma1 -0.99795588  0.03759979 -26.5415 <2e-16 ***
ma2  0.04152225  0.05199187   0.7986  0.4245
ma3 -0.00070278  0.05364057  -0.0131  0.9895
ma4 -0.02649560  0.03458896  -0.7660  0.4437
---

```

Gambar B. 3 Uji Signifikansi ARIMA(1,1,2), ARIMA(1,1,3), ARIMA(1,1,4)

```

> #ARIMA(1,1,5)
> coeftest(arima(train_log, order=c(1,1,5)))

z test of coefficients:

      Estimate Std. Error  z value Pr(>|z|)
ar1  0.516464   0.219912   2.3485  0.01885 *
ma1 -0.518614   0.218589  -2.3725  0.01767 *
ma2  0.052870   0.040798   1.2959  0.19501
ma3  0.023328   0.040850   0.5711  0.56796
ma4 -0.026332   0.042038  -0.6264  0.53107
ma5  0.089583   0.037450   2.3921  0.01675 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> #ARIMA(1,1,6)
> coeftest(arima(train_log, order=c(1,1,6)))

z test of coefficients:

      Estimate Std. Error  z value Pr(>|z|)
ar1  0.1216590  0.3875567   0.3139  0.75359
ma1 -0.1216851  0.3864355  -0.3149  0.75284
ma2  0.0493771  0.0365025   1.3527  0.17615
ma3  0.0465278  0.0417720   1.1139  0.26534
ma4 -0.0072642  0.0420912  -0.1726  0.86298
ma5  0.0751407  0.0358979   2.0932  0.03633 *
ma6  0.0708326  0.0470997   1.5039  0.13261
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Gambar B. 4 Uji Signifikansi ARIMA(1,1,5), ARIMA(1,1,6)



```

> #ARIMA(2,1,2)
> coefptest(arima(train_log, order=c(2,1,2)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.818000   0.601741  1.3594  0.1740
ar2  0.131847   0.585960  0.2250  0.8220
ma1  -0.821157   0.605255 -1.3567  0.1749
ma2  -0.089901   0.574738 -0.1564  0.8757

> #ARIMA(2,1,3)
> coefptest(arima(train_log, order=c(2,1,3)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.394094   1.227021  0.3212  0.7481
ar2  0.532060   1.159914  0.4587  0.6464
ma1  -0.396331   1.230756 -0.3220  0.7474
ma2  -0.504712   1.185620 -0.4257  0.6703
ma3  0.030872   0.079491  0.3884  0.6977

> #ARIMA(2,1,4)
> coefptest(arima(train_log, order=c(2,1,4)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  1.357642   0.745769  1.8205  0.06869 .
ar2  -0.588291   0.673940 -0.8729  0.38271 .
ma1  -1.359680   0.742630 -1.8309  0.06712 .
ma2  0.637989   0.675165  0.9449  0.34469 .
ma3  -0.036021   0.077318 -0.4659  0.64130
ma4  0.039629   0.045716  0.8669  0.38602
---

```

**Gambar B. 5 Uji Signifikansi ARIMA(2,1,2), ARIMA(2,1,3), ARIMA(2,1,4)**

```

> #ARIMA(2,1,5)
> coefptest(arima(train_log, order=c(2,1,5)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.675962   0.207455  3.2584  0.0011206 **
ar2  -0.388522   0.237128 -2.4819  0.0130692 *
ma1  -0.678141   0.206064 -3.2909  0.0009986 ***
ma2  0.640194   0.239792  2.6698  0.0075898 **
ma3  0.014839   0.049575  0.2993  0.7646939
ma4  -0.021627   0.044980 -0.4808  0.6306509
ma5  0.116911   0.037074  3.1534  0.0016137 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> #ARIMA(2,1,6)
> coefptest(arima(train_log, order=c(2,1,6)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.482096   0.091746  5.2547  1.483e-07 ***
ar2  -0.841315   0.080566 -10.4425 < 2.2e-16 ***
ma1  -0.484119   0.096619  -5.0106  5.426e-07 ***
ma2  0.893541   0.090544  9.8685 < 2.2e-16 ***
ma3  0.026078   0.052518  0.4966  0.61950
ma4  0.011168   0.051091  0.2186  0.82697
ma5  0.079266   0.041965  1.8888  0.05891 .
ma6  0.044329   0.041155  1.0771  0.28142
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

**Gambar B. 6 Uji Signifikansi ARIMA(2,1,5), ARIMA(2,1,6)**

```

> #ARIMA(3,1,1)
> coefest(arima(train_log, order=c(3,1,1)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1 -0.347570   0.475574 -0.7308  0.46488
ar2  0.055060   0.038448  1.4321  0.15212
ar3  0.072219   0.038916  1.8558  0.06349
ma1  0.354639   0.476179  0.7448  0.45642
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> #ARIMA(3,1,2)
> coefest(arima(train_log, order=c(3,1,2)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1 -0.601194   0.363461 -1.6541  0.09811
ar2 -0.423451   0.372227 -1.1376  0.25528
ar3  0.071475   0.040213  1.7774  0.07550
ma1  0.610700   0.363654  1.6793  0.09309
ma2  0.479799   0.364749  1.3154  0.18837
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

**Gambar B. 7 Uji Signifikansi ARIMA(3,1,1), ARIMA(3,1,2)**

```

> #ARIMA(3,1,3)
> coefest(arima(train_log, order=c(3,1,3)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1 -0.16908    0.20844  -0.8112  0.417279
ar2  0.22252    0.13563  1.6406  0.100877
ar3  0.83461    0.27842  2.9976  0.002721 **
ma1  0.17011    0.23327  0.7230  0.469654
ma2 -0.18483    0.14675  -1.2595  0.207857
ma3 -0.77985    0.30132  -2.5881  0.009649 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> #ARIMA(3,1,4)
> coefest(arima(train_log, order=c(3,1,4)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1 -0.1942170  0.1047822  -1.8535  0.06381
ar2  0.2051422  0.1214048  1.6897  0.09108
ar3  0.8671311  0.1701364  5.0967  3.457e-07 ***
ma1  0.1921014  0.1110469  1.7299  0.08365
ma2 -0.1678201  0.1320387  -1.2710  0.20373
ma3 -0.8123530  0.1936252  -4.1955  2.723e-05 ***
ma4  0.0092571  0.0416918  0.2220  0.82429
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

**Gambar B. 8 Uji Signifikansi ARIMA(3,1,3), ARIMA(3,1,4)**

```

> #ARIMA(3,1,5)
> coeftest(arima(train_log, order=c(3,1,5)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  1.303155  0.054178  24.0532 <2e-16 ***
ar2 -1.294973  0.028867 -44.8593 <2e-16 ***
ar3  0.901364  0.052367  17.2124 <2e-16 ***
ma1 -1.309279  0.066353 -19.7322 <2e-16 ***
ma2  1.369513  0.070778  19.3494 <2e-16 ***
ma3 -0.954962  0.103114  -9.2613 <2e-16 ***
ma4  0.029091  0.065268  0.4457  0.6558
ma5  0.020989  0.041479  0.5060  0.6128
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> #ARIMA(3,1,6)
> coeftest(arima(train_log, order=c(3,1,6)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1 -0.1742074  0.3792240 -0.4594  0.64596
ar2 -0.0471645  0.4709143 -0.1002  0.92022
ar3 -0.4577860  0.3377604 -1.3554  0.17530
ma1  0.1729805  0.3781294  0.4575  0.64734
ma2  0.0957171  0.4689743  0.2041  0.83828
ma3  0.5177236  0.3509385  1.4753  0.14014
ma4 -0.0012755  0.0376641 -0.0339  0.97299
ma5  0.0878109  0.0362608  2.4216  0.01545 *
ma6  0.1116793  0.0457081  2.4433  0.01455 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Gambar B. 9 Uji Signifikansi ARIMA(3,1,5), ARIMA(3,1,6)

```

> #dengan AR(4)
> #ARIMA(4,1,1)
> coeftest(arima(train_log, order=c(4,1,1)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.9821561  0.0380697  25.7989 <2e-16 ***
ar2  0.0451801  0.0506646  0.8917  0.3725
ar3  0.0051361  0.0506284  0.1014  0.9192
ar4 -0.0361366  0.0367958 -0.9821  0.3261
ma1 -0.9863969  0.0123423 -79.9202 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> #ARIMA(4,1,2)
> coeftest(arima(train_log, order=c(4,1,2)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.399204  0.591476  0.6749  0.4997
ar2  0.551220  0.407783  1.3517  0.1765
ar3  0.038988  0.056762  0.6869  0.4922
ar4 -0.036316  0.076584 -0.4742  0.6354
ma1 -0.402476  0.590736 -0.6813  0.4957
ma2 -0.508662  0.408766 -1.2444  0.2134

```

Gambar B. 10 Uji Signifikansi ARIMA(4,1,1), ARIMA(4,1,2)

```

> #ARIMA(4,1,3)
> coefest(arima(train_log, order=c(4,1,3)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.285826   0.379024  0.7541  0.4508
ar2  0.222541   0.400356  0.5559  0.5783
ar3  0.536444   0.334642  1.6030  0.1089
ar4 -0.050537   0.040018 -1.2629  0.2066
ma1 -0.287611   0.379062 -0.7587  0.4480
ma2 -0.176942   0.411632 -0.4299  0.6673
ma3 -0.508257   0.338808 -1.5001  0.1336

> #ARIMA(4,1,4)
> coefest(arima(train_log, order=c(4,1,4)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.150144   0.867500  0.1731  0.86259
ar2  0.152256   0.395056  0.3854  0.69994
ar3  0.594753   0.290779  2.0454  0.04082 *
ar4  0.091016   0.793221  0.1147  0.90865
ma1 -0.151778   0.867036 -0.1751  0.86104
ma2 -0.107787   0.405257 -0.2660  0.79026
ma3 -0.564754   0.299654 -1.8847  0.05947 .
ma4 -0.133884   0.778824 -0.1719  0.86351
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Gambar B. 11 Uji Signifikansi ARIMA(4,1,3), ARIMA(4,1,4)

```

> #ARIMA(4,1,5)
> coefest(arima(train_log, order=c(4,1,5)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.554607   0.518525  1.0696  0.2848
ar2 -0.317333   0.688497 -0.4609  0.6449
ar3 -0.070967   0.670215 -0.1059  0.9157
ar4  0.679194   0.483845  1.4037  0.1604
ma1 -0.561125   0.517138 -1.0851  0.2779
ma2  0.388617   0.664958  0.5844  0.5589
ma3  0.083388   0.677173  0.1231  0.9020
ma4 -0.706817   0.488785 -1.4461  0.1482
ma5  0.060820   0.039778  1.5290  0.1263

> #ARIMA(4,1,6)
> coefest(arima(train_log, order=c(4,1,6)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.320706   0.068556  4.6780 2.896e-06 ***
ar2 -0.010204   0.047056 -0.2169  0.8283
ar3 -0.374099   0.049227 -7.5994 2.974e-14 ***
ar4  0.891093   0.063692 13.9906 < 2.2e-16 ***
ma1 -0.325902   0.078071 -4.1745 2.987e-05 ***
ma2  0.077574   0.060862  1.2746  0.2025
ma3  0.398276   0.067076  5.9377 2.890e-09 ***
ma4 -0.922177   0.079869 -11.5462 < 2.2e-16 ***
ma5  0.056307   0.042099  1.3375  0.1811
ma6  0.014452   0.042595  0.3393  0.7344
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Gambar B. 12 Uji Signifikansi ARIMA(4,1,5), ARIMA(4,1,6)

```

> #ARIMA(5,1,1)
> coefptest(arima(train_log, order=c(5,1,1)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.501742  0.257764  1.9465  0.05159 .
ar2  0.043798  0.040368  1.0850  0.27793
ar3  0.024998  0.042230  0.5919  0.55389
ar4 -0.027243  0.042824 -0.6362  0.52468
ar5  0.086691  0.039106  2.2168  0.02664 *
ma1 -0.500791  0.257961 -1.9413  0.05222 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> #ARIMA(5,1,2)
> coefptest(arima(train_log, order=c(5,1,2)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.652728  0.254728  2.5625  0.010394 *
ar2 -0.430205  0.212709 -2.0225  0.043124 *
ar3  0.019728  0.047323  0.4169  0.676765
ar4 -0.019017  0.043526 -0.4369  0.662175
ar5  0.118667  0.038523  3.0804  0.002067 **
ma1 -0.651609  0.256540 -2.5400  0.011086 *
ma2  0.479389  0.211406  2.2676  0.023352 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

**Gambar B. 13 Uji Signifikansi ARIMA(5,1,1), ARIMA(5,1,2)**

```

> coefptest(arima(train_log, order=c(5,1,3)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  1.239162  0.081895  15.1312 <2e-16 ***
ar2 -1.236146  0.063394 -19.4994 <2e-16 ***
ar3  0.852471  0.084269  10.1160 <2e-16 ***
ar4  0.025659  0.058430  0.4392  0.6606
ar5  0.024647  0.040906  0.6025  0.5468
ma1 -1.247055  0.074444 -16.7515 <2e-16 ***
ma2  1.311767  0.028857  45.4575 <2e-16 ***
ma3 -0.901025  0.074552 -12.0859 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

**Gambar B. 14 Uji Signifikansi ARIMA(5,1,3)**

```

> #ARIMA(5,1,4)
> coeftest(arima(train_log, order=c(5,1,4)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.616230  0.436763  1.4109  0.1583
ar2 -0.447840  0.570809 -0.7846  0.4327
ar3  0.088840  0.553855  0.1604  0.8726
ar4  0.552458  0.398667  1.3858  0.1658
ar5  0.059549  0.040463  1.4717  0.1411
ma1 -0.623593  0.436384 -1.4290  0.1530
ma2  0.515173  0.561817  0.9170  0.3592
ma3 -0.076213  0.574391 -0.1327  0.8944
ma4 -0.580847  0.416338 -1.3951  0.1630

> #ARIMA(5,1,5)
> coeftest(arima(train_log, order=c(5,1,5)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.2892260  0.6646966  0.4351  0.663472
ar2 -0.6595849  0.2486036 -2.6532  0.007974 **
ar3  0.5466079  0.4481740  1.2196  0.222604
ar4  0.0073071  0.5402928  0.0135  0.989209
ar5  0.6833305  0.8665751  0.7885  0.430380
ma1 -0.2898167  0.7050874 -0.4110  0.681046
ma2  0.7345862  0.2667611  2.7537  0.005892 **
ma3 -0.5120945  0.4366871 -1.1727  0.240924
ma4 -0.0193778  0.5466384 -0.0354  0.971722
ma5 -0.6537544  0.9496516 -0.6884  0.491192
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Gambar B. 15 Uji Signifikansi ARIMA(5,1,4), ARIMA(5,1,5)

```

> #ARIMA(5,1,6)
> coeftest(arima(train_log, order=c(5,1,6)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1 -0.304173      NA      NA      NA
ar2  0.129478      NA      NA      NA
ar3 -0.288756      NA      NA      NA
ar4  0.572638  0.056318 10.1680 < 2.2e-16 ***
ar5  0.625174  0.094524  6.6139 3.743e-11 ***
ma1  0.298241      NA      NA      NA
ma2 -0.072825      NA      NA      NA
ma3  0.363202      NA      NA      NA
ma4 -0.586471  0.084970 -6.9021 5.123e-12 ***
ma5 -0.597464  0.091072 -6.5603 5.369e-11 ***
ma6  0.063911  0.030532  2.0933  0.03633 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Gambar B. 16 Uji Signifikansi ARIMA(5,1,6)

```

> coeftest(arima(train_log, order=c(6,1,1)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1 -0.0014478  0.3744882 -0.0039  0.99692
ar2  0.0472013  0.0360686  1.3087  0.19065
ar3  0.0469247  0.0401829  1.1678  0.24290
ar4 -0.0037546  0.0408866 -0.0918  0.92683
ar5  0.0768437  0.0360669  2.1306  0.03312 *
ar6  0.0808725  0.0458321  1.7645  0.07764 .
ma1  0.0011630  0.3752133  0.0031  0.99753
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> #ARIMA(6,1,2)
> coeftest(arima(train_log, order=c(6,1,2)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.449011  0.119022  3.7725 0.0001616 ***
ar2 -0.780673  0.079435 -9.8278 < 2.2e-16 ***
ar3  0.032556  0.048804  0.6671 0.5047155
ar4  0.016247  0.048811  0.3328 0.7392528
ar5  0.092373  0.043230  2.1368 0.0326166 *
ar6  0.041713  0.043537  0.9581 0.3380057
ma1 -0.448192  0.114221 -3.9239 8.712e-05 ***
ma2  0.841368  0.070476 11.9383 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

**Gambar B. 17 Uji Signifikansi ARIMA(6,1,1), ARIMA(6,1,2)**

```

> #ARIMA(6,1,3)
> coeftest(arima(train_log, order=c(6,1,3)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1 -0.1334930  0.8796190 -0.1518  0.87937
ar2 -0.0070526  0.7954687 -0.0089  0.99293
ar3 -0.2598675  0.4942909 -0.5257  0.59907
ar4  0.0049130  0.0391540  0.1255  0.90014
ar5  0.0944211  0.0383605  2.4614  0.01384 *
ar6  0.1069448  0.0896151  1.1934  0.23272
ma1  0.1340953  0.8815551  0.1521  0.87910
ma2  0.0581701  0.7996219  0.0727  0.94201
ma3  0.3178565  0.5340886  0.5951  0.55175
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> #ARIMA(6,1,4)
> coeftest(arima(train_log, order=c(6,1,4)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.808700  0.746743  1.0830  0.2788
ar2 -0.672206  0.903858 -0.7437  0.4571
ar3  0.325849  0.912689  0.3570  0.7211
ar4  0.388598  0.628643  0.6182  0.5365
ar5  0.064334  0.047545  1.3531  0.1760
ar6 -0.017907  0.052970 -0.3381  0.7353
ma1 -0.815664  0.746580 -1.0925  0.2746
ma2  0.744525  0.908529  0.8195  0.4125
ma3 -0.324216  0.964252 -0.3362  0.7367
ma4 -0.418026  0.653137 -0.6400  0.5222

```

**Gambar B. 18 Uji Signifikansi ARIMA(6,1,3), ARIMA(6,1,4)**

```

> #ARIMA(6,1,5)
> coeftest(arima(train_log, order=c(6,1,5)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.3319196         NA      NA      NA
ar2 -0.7113500         NA      NA      NA
ar3  0.5819814  0.2940774  1.9790 0.04782 *
ar4 -0.0153112  0.2671045 -0.0573 0.95429
ar5  0.6642427         NA      NA      NA
ar6  0.0060027         NA      NA      NA
ma1 -0.3373519         NA      NA      NA
ma2  0.7898596         NA      NA      NA
ma3 -0.5552771  0.2927264 -1.8969 0.05784 .
ma4  0.0136060  0.2931589  0.0464 0.96298
ma5 -0.6358274         NA      NA      NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Gambar B. 19 Uji Signifikansi ARIMA(6,1,5)

```

> #ARIMA(6,1,6)
> coeftest(arima(train_log, order=c(6,1,6)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1 -0.403537  0.797210 -0.5062 0.61273
ar2 -0.135540  0.238699 -0.5678 0.57015
ar3 -0.090624         NA      NA      NA
ar4  0.409316         NA      NA      NA
ar5  0.737270  0.432376  1.7052 0.08816 .
ar6  0.181842  0.213539  0.8516 0.39446
ma1  0.397323  0.803182  0.4947 0.62082
ma2  0.200430  0.253882  0.7895 0.42984
ma3  0.178976         NA      NA      NA
ma4 -0.407817         NA      NA      NA
ma5 -0.708675  0.465690 -1.5218 0.12807
ma6 -0.126969  0.202471 -0.6271 0.53060
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Gambar B. 20 Uji Signifikansi ARIMA(6,1,6)

```

> #tes signifikansi model d2
> coeftest(arima(train_log, order=c(0,2,1)))

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ma1 -0.9889269  0.0055915 -176.86 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Gambar B. 21 Uji Signifikansi ARIMA(0,2,1)



## LAMPIRAN C: UJI DIAGNOSA

```
> checkresiduals(model1, plot = FALSE)

      Ljung-Box test

data: Residuals from ARIMA(0,2,1)
Q* = 9.2268, df = 9, p-value = 0.4166

Model df: 1.    Total lags used: 10

> checkresiduals(model2, plot = FALSE)

      Ljung-Box test

data: Residuals from ARIMA(1,1,1)
Q* = 8.9825, df = 8, p-value = 0.3438

Model df: 2.    Total lags used: 10
```

Gambar C. 1 Uji Diagnosa ARIMA(0,2,1) dan ARIMA(1,1,1)

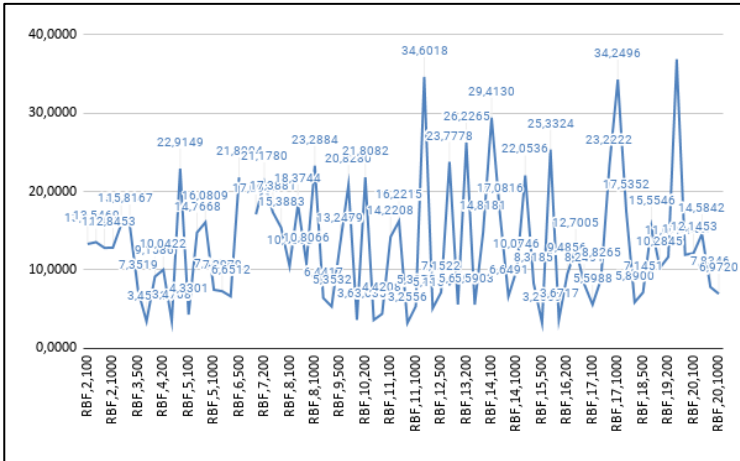
*“Halaman ini sengaja dikosongkan”*

**LAMPIRAN D:  
HASIL PERAMALAN ARIMA**

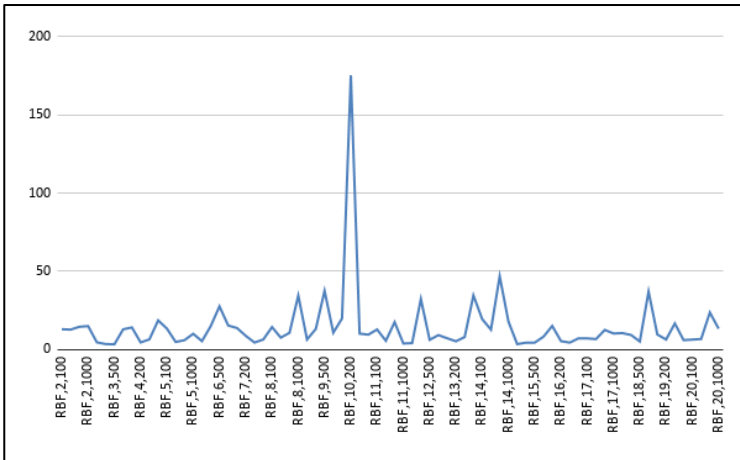
Periode	Aktual	Forecast	Residu
01/01/2017	8,20	8,18276	0,01724
01/02/2017	8,37	8,20089	0,16911
01/03/2017	9,74	8,38036	1,35964
01/04/2017	11,00	9,78413	1,21587
01/05/2017	10,15	11,06837	-0,91837
01/06/2017	10,08	10,19640	-0,11640
01/07/2017	9,84	10,12249	-0,28249
01/08/2017	10,28	9,87781	0,40219
01/09/2017	10,26	10,32597	-0,06597
01/10/2017	10,61	10,30542	0,30458
.....	.....	.....	.....
12/22/2019	132,19	127,17926	5,01074
12/23/2019	128,08	131,81547	-3,73547
12/24/2019	128,05	127,67121	0,37879
12/25/2019	125,24	127,64978	-2,40978
12/26/2019	125,93	124,82013	1,10987
12/27/2019	126,41	125,52618	0,88382
12/28/2019	128,35	126,01985	2,33015
12/29/2019	134,40	127,98846	6,41154
12/30/2019	131,83	134,11201	-2,28201
12/31/2019	129,21	131,51978	-2,30978

*“Halaman ini sengaja dikosongkan”*

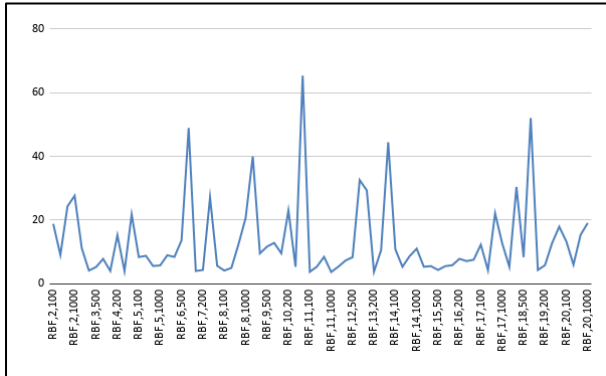
## LAMPIRAN E: HASIL PENGUJIAN RBFNN



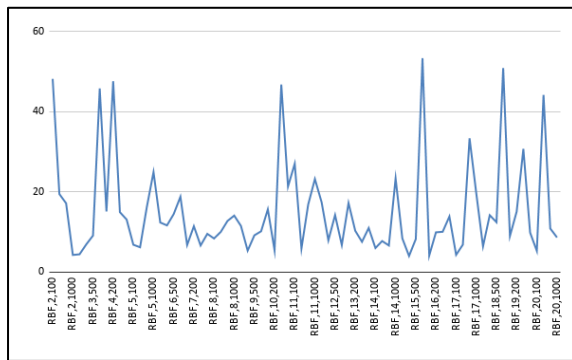
**Gambar E. 1 Hasil Pengujian Input 2 (Lag)**



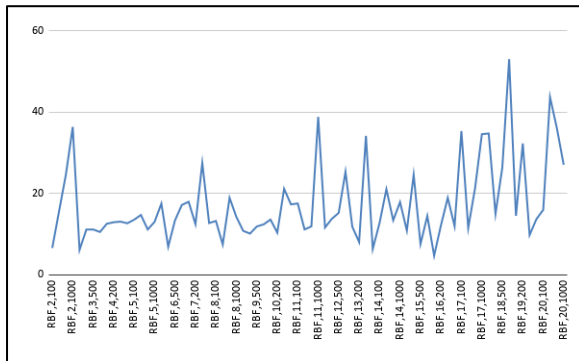
**Gambar E. 2 Hasil Pengujian Input 3 (Lag)**



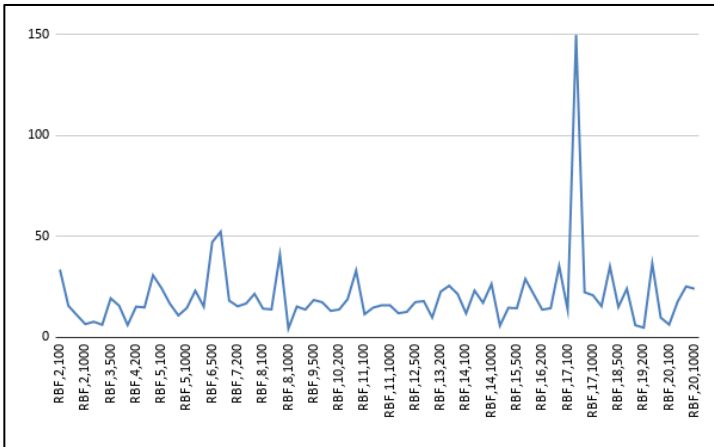
**Gambar E. 3 Hasil Pengujian Input 4 (Lag)**



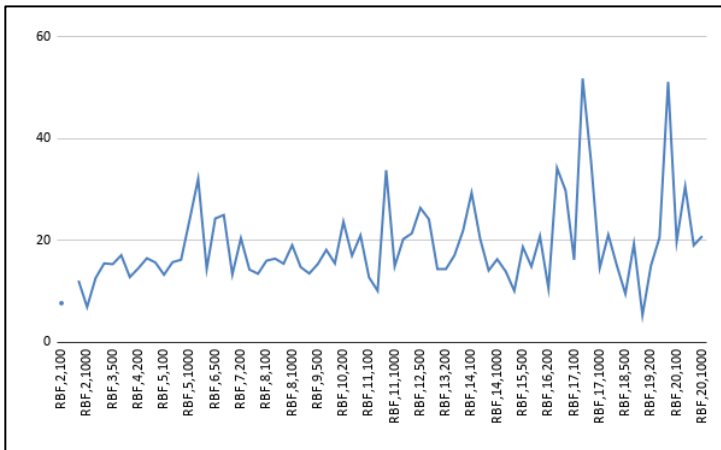
**Gambar E. 4 Hasil Pengujian Input 5 (Lag)**



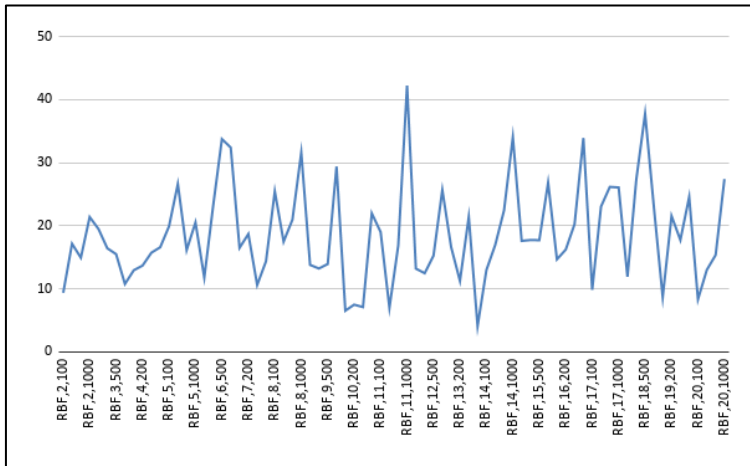
**Gambar E. 5 Hasil Pengujian Input 6 (Lag)**



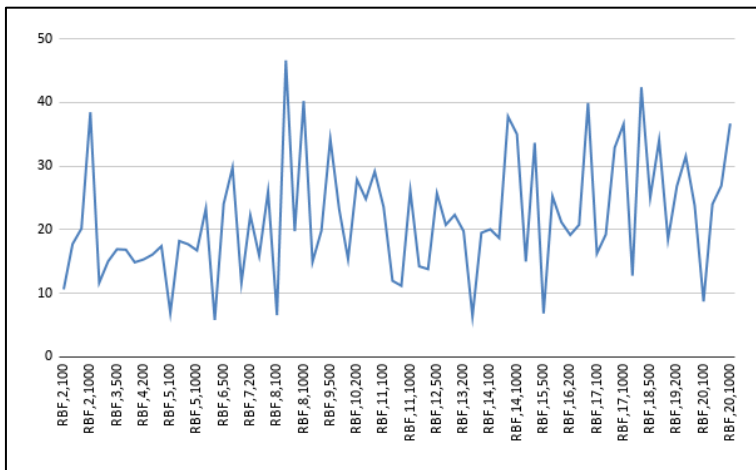
**Gambar E. 6 Hasil Pengujian Input 7 (Lag)**



**Gambar E. 7 Hasil Pengujian Input 8 (Lag)**



**Gambar E. 8 Hasil Pengujian Input 9 (Lag)**



**Gambar E. 9 Hasil Pengujian Input 10 (Lag)**



**LAMPIRAN F:  
HASIL PERMALAN RBFNN**

Periode	Aktual	Forecast	Residu
01/01/2017	8,2		
01/02/2017	8,37		
01/03/2017	9,74	8,27882	1,46118
01/04/2017	11	9,08013	1,91987
01/05/2017	10,15	10,40146	-0,25146
01/06/2017	10,08	10,57611	-0,49611
01/07/2017	9,84	10,12280	-0,28280
01/08/2017	10,28	9,96319	0,31681
01/09/2017	10,26	10,07767	0,18233
01/10/2017	10,61	10,28110	0,32890
1/11/2017	9,79	10,45706	-0,66706
1/12/2017	9,83	10,19672	-0,36672
1/13/2017	9,67	9,81929	-0,14929
1/14/2017	9,67	9,75584	-0,08584
1/15/2017	8,2	8,27882	1,46118
.....	.....	.....	.....
12/17/2019	122,06	138,80493	-16,74493
12/18/2019	132,92	128,45144	4,46856
12/19/2019	128,42	128,89624	-0,47624
12/20/2019	128,7	131,84373	-3,14373
12/21/2019	127,61	129,79370	-2,18370
12/22/2019	132,19	127,17926	5,01074
12/23/2019	128,08	131,81547	-3,73547
12/24/2019	128,05	127,67121	0,37879
12/25/2019	125,24	127,64978	-2,40978
12/26/2019	125,93	124,82013	1,10987
12/27/2019	126,41	125,52618	0,88382
12/28/2019	128,35	126,01985	2,33015
12/29/2019	134,40	127,98846	6,41154
12/30/2019	131,83	134,11201	-2,28201
12/31/2019	129,21	131,51978	-2,30978

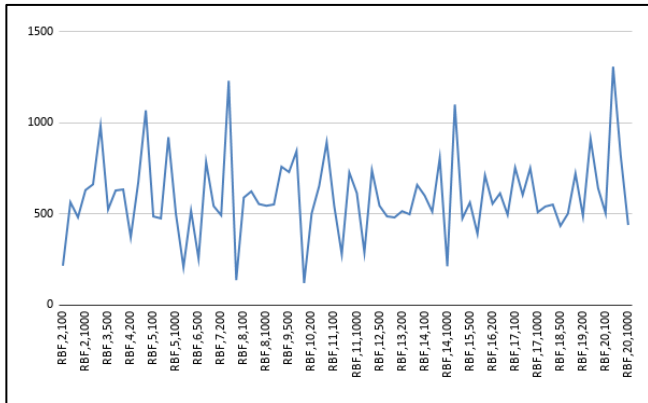
*“Halaman ini sengaja dikosongkan”*

**LAMPIRAN G:  
NILAI RESIDUAL ARIMA (INPUT RBFNN)**

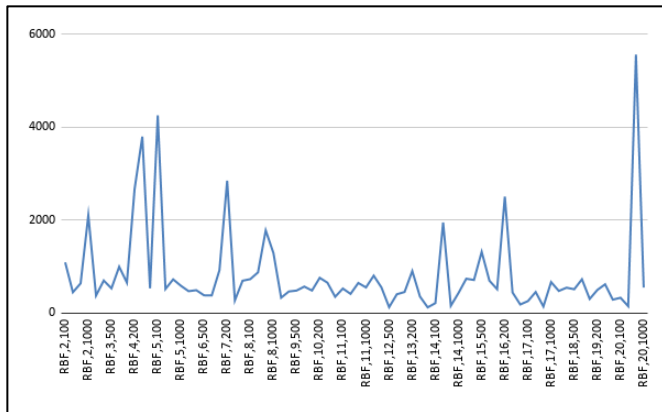
Periode	Residu
1/1/2017	0,017236
1/2/2017	0,168765
1/3/2017	1,356632
1/4/2017	1,209332
1/5/2017	-0,924574
1/6/2017	-0,121555
1/7/2017	-0,287079
1/8/2017	0,397098
1/9/2017	-0,071292
1/10/2017	0,298772
1/11/2017	-0,873415
1/12/2017	0,004958
1/13/2017	-0,194218
1/14/2017	-0,030223
1/15/2017	0,109959
.....	.....
12/17/2019	-11,415268
12/18/2019	10,344550
12/19/2019	-3,519651
12/20/2019	-0,347258
12/21/2019	-0,863462
12/22/2019	4,429156
12/23/2019	-3,730227
12/24/2019	-0,437609
12/25/2019	-2,677902
12/26/2019	0,451097
12/27/2019	0,607944
12/28/2019	1,931512
12/29/2019	6,190650
12/30/2019	-2,182555
12/31/2019	-2,925492

*“Halaman ini sengaja dikosongkan”*

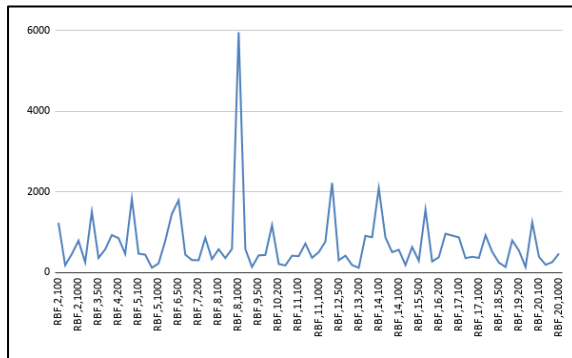
## LAMPIRAN H: HASIL PENGUJIAN RESIDUAL ARIMA (RBFNN)



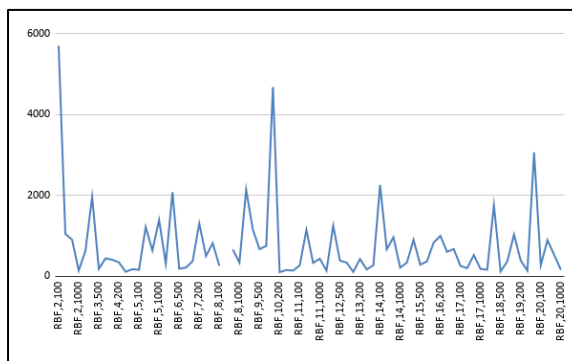
**Gambar H. 1 Hasil Pengujian Input 2 (Lag)**



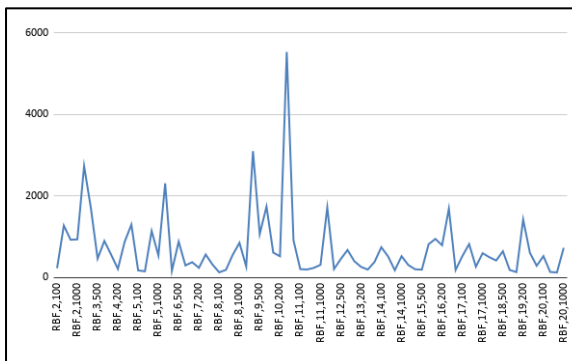
**Gambar H. 2 Hasil Pengujian Input 3 (Lag)**



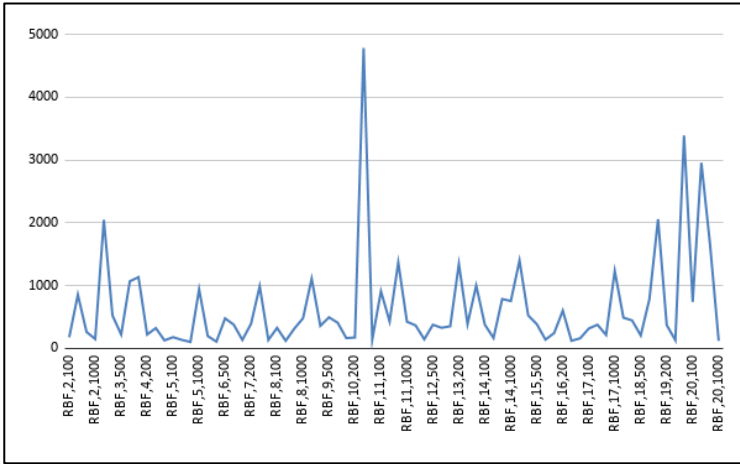
**Gambar H. 3 Hasil Pengujian Input 4 (Lag)**



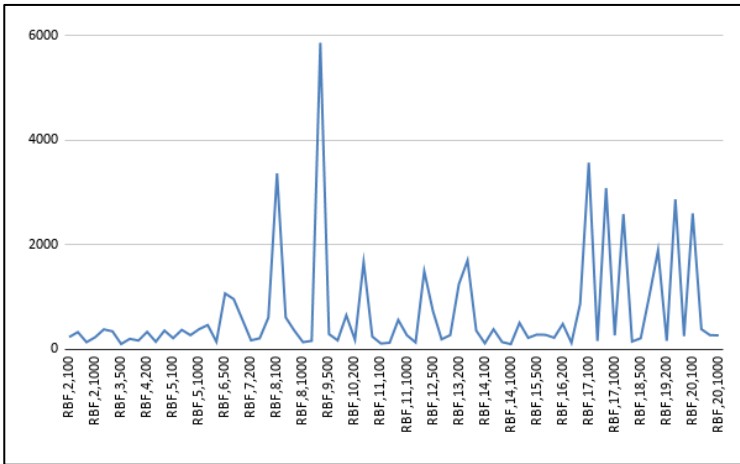
**Gambar H. 4 Hasil Pengujian Input 5 (Lag)**



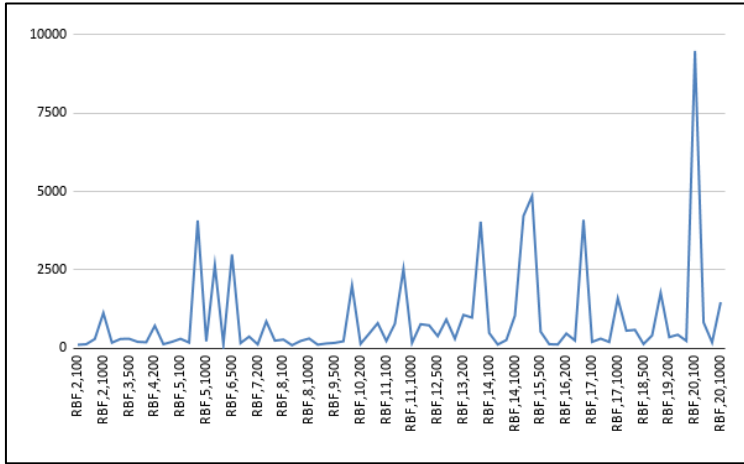
**Gambar H. 5 Hasil Pengujian Input 6 (Lag)**



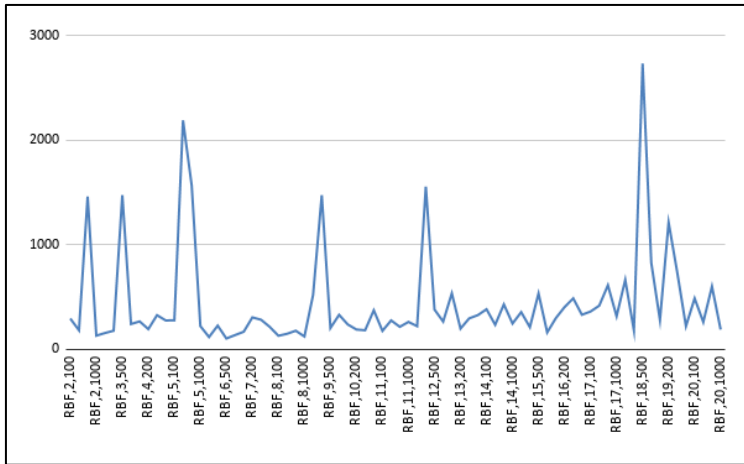
Gambar H. 6 Hasil Pengujian Input 7 (Lag)



Gambar H. 7 Hasil Pengujian Input 8 (Lag)



Gambar H. 8 Hasil Pengujian Input 9 (Lag)



Gambar H. 9 Hasil Pengujian Input 10 (Lag)



**LAMPIRAN I:  
HASIL PERAMALAN HYBRID ARIMA-RBFNN**

Date	Aktual	Forecast ARIMA	Forecast Residu (RBFNN)	Forecast ARIMA-RBFNN
1/1/2017	8,2	8,18276		8,18276
1/2/2017	8,37	8,20124		8,20124
1/3/2017	9,74	8,38337		8,38337
1/4/2017	11	9,79067		9,79067
1/5/2017	10,15	11,07457		11,07457
1/6/2017	10,08	10,20156		10,20156
1/7/2017	9,84	10,12708		10,12708
1/8/2017	10,28	9,88290		9,88290
1/9/2017	10,26	10,33129	-0,09397	10,23733
1/10/2017	10,61	10,31123	-0,07015	10,24108
1/11/2017	9,79	10,66341	-0,01705	10,64636
1/12/2017	9,83	9,82504	-0,05053	9,77451
1/13/2017	9,67	9,86422	-0,04809	9,81613
1/14/2017	9,67	9,70022	-0,06537	9,63486
1/15/2017	9,81	9,70004	-0,03853	9,66151
.....	.....	.....	.....	.....
12/17/2019	122,06	133,47527	0,14546	133,62073
12/18/2019	132,92	122,57545	0,31685	122,89230
12/19/2019	128,42	131,93965	0,08934	132,02899
12/20/2019	128,7	129,04726	0,14351	129,19076
12/21/2019	127,61	128,47346	0,14965	128,62312
12/22/2019	132,19	127,76084	0,16078	127,92163
12/23/2019	128,08	131,81023	-0,01042	131,79981
12/24/2019	128,05	128,48761	-0,53016	127,95745
12/25/2019	125,24	127,91790	0,07655	127,99445
12/26/2019	125,93	125,47890	0,00012	125,47903
12/27/2019	126,41	125,80206	-0,00095	125,80111
12/28/2019	128,35	126,41849	-0,13145	126,28704
12/29/2019	134,4	128,20935	0,07147	128,28082
12/30/2019	131,83	134,01256	-0,20924	133,80332
12/31/2019	129,21	132,13549	-0,15946	131,97603