



TESIS - ST 2309

**PERFORMANSI GPH TERKOREKSI TERHADAP  
*SKIP SAMPLING* PADA PROSES *LONG MEMORY*  
DAN *SPURIOUS LONG MEMORY***

GEDE SUWARDIKA  
NRP. 1311 201 004

DOSEN PEMBIMBING  
Dr. rer. pol. Heri Kuswanto, S.Si., M.Si.

PROGRAM MAGISTER  
JURUSAN STATISTIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2013

# PERFORMANSI GPH TERKOREKSI TERHADAP SKIP SAMPLING PADA PROSES LONG MEMORY DAN SPURIOUS LONG MEMORY

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

**Magister Sains (M.Si.)**

di

**Institut Teknologi Sepuluh Nopember**

oleh:

**GEDE SUWARDIKA**

**NRP. 1311 201 004**

**Tanggal Ujian : 17 Juli 2013**  
**Periode Wisuda : September 2013**

**Disetujui oleh:**

**1. Dr. rer. pol. Heri Kuswanto, S.Si., M.Si**

**(Pembimbing)**

**NIP : 19820326 200312 1 004**

**2. Dr. Irhamah, S.Si., M.Si**

**(Penguji)**

**NIP : 19780406 200112 2 002**

**3. Dr. Brojol Sutijo Suprih Ulama, S.Si., M.Si**

**(Penguji)**

**NIP : 19660125 199002 1 001**

**Direktur Program Pascasarjana,**

**Prof. Dr. Ir. Adi Soeprijanto, MT**

**NIP. 19640405 199002 1 001**

# PERFORMANSI GPH TERKOREKSI TERHADAP SKIP SAMPLING PADA PROSES *LONG MEMORY* DAN *SPURIOUS LONG MEMORY*

Nama mahasiswa : Gede Suwardika  
NRP : 1311 201 004  
Pembimbing : Dr. rer. pol. Heri Kuswanto, S.Si., M.Si.

## ABSTRAK

Proses *long memory* telah diamati dalam banyak hal, seperti hidrologi, telekomunikasi, ekonomi dan keuangan. *Long Memory* adalah salah satu fenomena dalam time series, dimana dependensi antara kejadian masih ada dan dapat diamati untuk waktu yang lama, yang dicirikan oleh nilai *difference* yang tidak bulat (*fractional*). Parameter *differencing* ini biasanya diestimasi menggunakan GPH estimator. Dengan estimator ini, seringkali menghasilkan kesimpulan yang *spurious* untuk model-model seperti *Estar*, *Markov switching*, *STOP-BREAK* dan *level shift*. Tesis ini akan melakukan simulasi model-model tersebut dan estimasi parameter GPH terkoreksi pada proses agregasi. Selanjutnya dilakukan pemodelan menggunakan ARFIMA dan *Markov Switching* pada data *stock price* LQ45. Pengidentifikasian sifat *Long Memory* dalam suatu series data dapat dilakukan dengan agregasi baik *flow aggregation* maupun *stock aggregation*. Dimana pada kasus ini hanya menggunakan *stock aggregation*. Berdasarkan hasil simulasi, *stock aggregation* ini menghasilkan perilaku yang sama dalam parameternya untuk *Spurious Long Memory*, yaitu random, tidak memiliki trend turun atau naik jika seriesnya diagregasi. Pemodelan dari absolut *return* saham dari kedua series terpilih yaitu Indosat dan Telkom, didapatkan bahwa model *Markov Swuching* lebih baik dibandingkan model ARFIMA. Hasil aplikasi saham menunjukkan nilai estimasi GPH untuk data teragregasi memiliki pola yang random, dilihat dari nilai AIC terkecil berdasarkan kedua model, model ARFIMA memiliki nilai AIC terkecil, sehingga GPH standart tidak bisa digunakan untuk mendeteksi *sprurious long memory*, dimana *return* saham dari kedua series mengandung outlier.

**Kata kunci:** *long memory*, *GPH Estimator*, ARFIMA, *Markov Swtiching*, *Stock Price*

# **CORRECTED GPH PERFORMANCE TOWARDS SKIP SAMPLING IN THE LONG MEMORY AND SPURIOUS LONG MEMORY PROCESSES**

Name : Gede Suwardika  
Student's Reg. No. : 1311 201 004  
Supervisor : Dr. rer. pol. Heri Kuswanto, S.Si., M.Si.

## **ABSTRACT**

Long memory process has been observed in many fields such as hydrology, telecommunications, economics, and finance. It is one of the phenomena in time series in which the dependencies between events still exist and can be observed in a long period of time characterized by a fractional difference value. This so-called differencing parameter is usually estimated by using GPH estimator. This estimator often reaches spurious conclusions for models such as Estar, Markov switching, STOP-BREAK, and level shift. This thesis conducted simulations for the models as well as corrected GPH estimation in the process of aggregation. Modeling was then performed using ARFIMA and Markov Switching in the LQ45 stock price data. The identification of the Long Memory nature in a series of data could be carried out through both flow aggregation and stock aggregation, yet only stock aggregation which was applied in this case. Based on the results of the simulations, this aggregation, stock aggregation, produces the same behavior in the parameter for Spurious Long Memory which is random and has no up or down trend if its series is aggregated. Through the modeling of absolute stock return of both chosen series, Indosat and Telkom, it was found that Markov Switching model is better than that of ARFIMA. Stock application results show that estimated values of the GPH for the aggregated data has a random pattern which can be seen from the smallest AIC value based on both models; ARFIMA model has the smallest AIC value that standard GPH cannot be used to detect spurious long memory in which stock returns of both series contain outliers.

Key words : Long Memory, GPH Estimator, ARFIMA, Markov Switching, Stock Price

## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Ida Sang Hyang Widhi Wasa Tuhan YME yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan laporan Tesis dengan judul “Performansi GPH terkoreksi terhadap Skip Sampling pada proses *Long Memory* dan *Spurious Long Memory* (Studi Kasus : Data Saham LQ 45)”.

Dalam penyusunan tesis ini, penulis banyak memperoleh bimbingan dan petunjuk, serta bantuan dan dukungan dari berbagai pihak baik dari institusi maupun luar institusi. Melalui kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada yang terhormat: Penulis menyampaikan rasa terima kasih sedalam-dalamnya kepada :

1. Bapak Dr. Muhammad Mashuri, M.T. selaku Ketua Jurusan Statistika FMIPA ITS Surabaya.
2. Bapak Dr. Suhartono, M.Sc. selaku ketua Program Studi Pascasarjana Jurusan Statistika ITS
3. Bapak Dr. rer. pol. Heri Kuswanto, M. Si. selaku dosen pembimbing laporan Tesis yang telah meluangkan waktu untuk memberi bimbingan, nasihat, dan saran.
4. Bapak Dr. Brodjoi Sutijo, S. Si, M. Si Dan Ibu Dr. Irhamah, S. Si, M. Si. selaku dosen penguji Tesis saya.
5. Kedua Orang Tua yang tercinta, Bapak Nyoman Putra yang tak henti-hentinya mengingatkanku untuk selalu menjaga diri dan ibadahku, Ibuku Wayan Sasih yang dengan tulus dan penuh kasih sayang menjaga dan merawatku hingga sekarang, dan adik-adikku Kade Suwartawa, Nyoman Satryawan serta Ketut Agus Sastrawan yang selalu riang dan penuh suka cita menghiburku dan selalu mendukung saya.
6. Istriku Putu Yusi Pramandari yang selalu setia mendampingi, mendukung setiap langkahku, dan memberikan support selalu.
7. Moch. Koesniawanto, Dian Anggraeni, Ketut Putu Suniantara, Zulhan Bagaskara, Gung Mas , Ikoh, Arinda, Lely, Gangga, Jerhi, Feni dan Indah

terima kasih atas bantuan, dukungan dan semangatnya yang selalu berbagi sukacita dan memberikan refreshing dengan cara tersendiri, terima kasih untuk persahabatan yang sudah terjalin selama ini.

8. Kepada teman-teman dan seluruh sahabat seperjuangan di Jurusan Statistika ITS yang selalu mendoakan, mendukung, dan memberi semangat yang tidak bisa saya sebutkan satu per satu sehingga penulis dapat menyelesaikan Tesis ini dengan baik.

Akhir kata, penulis menyadari sepenuhnya bahwa dalam penyusunan laporan Tesis ini masih banyak terdapat kekurangan. Oleh karena itu penulis akan berterima kasih apabila ada saran dan kritik yang membangun dari berbagai pihak demi perkembangan statistika yang lebih baik di kemudian hari.

Surabaya, Juli 2013

Penulis

## DAFTAR ISI

Judul .....	i
Lembar Pengesahan .....	ii
Abstrak .....	iii
Abstract .....	iv
Kata Pengantar .....	v
Daftar Isi .....	vii
Daftar Gambar .....	ix
Daftar Tabel .....	x
Daftar Lampiran .....	xi
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan Penelitian .....	3
1.4 Manfaat Penelitian .....	4
1.5 Batasan Masalah .....	4
<b>BAB II TINJAUAN PUSTAKA</b>	
2.1 <i>Long Memory</i> .....	5
2.2 <i>Spurious Long Memory</i> dan Implikasinya .....	5
2.3 Agregasi Temporal dalam <i>Long Memory Time Series</i> .....	6
2.4 Estimator GPH ( <i>Geweke dan Porter-Hudak</i> ) .....	8
2.5 ARFIMA ( <i>Autoregressive Fractionally Integrated Moving Average</i> ) .....	11
2.6 Pemilihan Model Terbaik .....	14
2.7 Cek Diagnosa .....	15
2.8 <i>Markov Switching</i> .....	16
2.9 Data Saham LQ45 .....	19
<b>BAB III METODOLOGI PENELITIAN</b>	
3.1 Sumber Data .....	21
3.2 Langkah Analisis .....	21
3.3 Diagram Alir Penelitian .....	24
<b>BAB IV ANALISIS PEMBAHASAN</b>	
4.1 Simulasi Stock Aggregate ( <i>skip sampling</i> ) .....	27
4.2 Simulasi Stock Aggregate pada model <i>Spurious Long Memory</i> .....	29
4.2.1 Model <i>Markov Switching</i> .....	30
4.2.2 Model <i>Stationary Random Level Shift</i> .....	35
4.2.3 Model <i>Stop Break</i> .....	38
4.3 Hasil dan Pembahasan Simulasi .....	42
4.4 Aplikasi ke Saham LQ45 .....	42
4.4.1 Identifikasi <i>Long Memory</i> .....	43
4.4.1.a Plot ACF .....	43
4.4.1.b Identifikasi data dengan <i>stock aggregation</i> .....	45
4.5 Pemodelan <i>Markov Switching</i> .....	49
4.5.1 Series data Indosat .....	49
4.5.2 series data Telkom .....	54

**BAB V KESIMPULAN DAN SARAN**

5.1 Kesimpulan .....	61
5.2 Saran.....	61
<b>DAFTARPUSTAKA</b> .....	<b>63</b>
<b>LAMPIRAN</b> .....	<b>67</b>

UNIVERSITAS TERBUKA



## DAFTAR GAMBAR

Gambar 3.1	Diagram alir langkah-langkah analisis .....	24
Gambar 4.1	Simulasi <i>Long Memory</i> GPH terkoreksi $n=500$ .....	28
Gambar 4.2	Simulasi <i>Long Memory</i> GPH tak terkoreksi $n=500$ .....	28
Gambar 4.3	Simulasi <i>Long Memory</i> GPH terkoreksi $n=2000$ .....	29
Gambar 4.4	Simulasi <i>Long Memory</i> GPH tak terkoreksi $n=2000$ .....	30
Gambar 4.5	Simulasi <i>Markov Switching</i> GPH terkoreksi $n=500$ .....	32
Gambar 4.6	Simulasi <i>Markov Switching</i> GPH tak terkoreksi $n=500$ .....	32
Gambar 4.7	Simulasi <i>Markov Switching</i> GPH terkoreksi $n=2000$ .....	34
Gambar 4.8	Simulasi <i>Markov Switching</i> GPH tak terkoreksi $n=2000$ .....	34
Gambar 4.9	Simulasi <i>Random Level Shift</i> GPH terkoreksi $n=500$ .....	36
Gambar 4.10	Simulasi <i>Random Level Shift</i> GPH tak terkoreksi $n=500$ .....	36
Gambar 4.11	Simulasi <i>Random Level Shift</i> GPH terkoreksi $n=2000$ .....	37
Gambar 4.12	Simulasi <i>Random Level Shift</i> GPH tak terkoreksi $n=2000$ .....	38
Gambar 4.13	Simulasi <i>Stop Break</i> GPH terkoreksi $n=500$ .....	39
Gambar 4.14	Simulasi <i>Stop Break</i> GPH tak terkoreksi $n=500$ .....	40
Gambar 4.15	Simulasi <i>Stop Break</i> GPH terkoreksi $n=2000$ .....	41
Gambar 4.16	Simulasi <i>Stop Break</i> GPH tak terkoreksi $n=2000$ .....	41
Gambar 4.17	Deskripsi series data <i>return</i> saham Indosat dan Telkom .....	43
Gambar 4.18	<i>Time series</i> plot dan plot ACF Indosat serta Telkom .....	44
Gambar 4.19	Plot bias estimasi nilai $d$ indosat .....	46
Gambar 4.20	Plot bias estimasi nilai $d$ Telkom .....	46
Gambar 4.21	Perbandingan nilai fit dan actual indosat pada model AR(1) .....	53
Gambar 4.22	Perbandingan nilai fit dan actual indosat pada model MA(1) .....	53
Gambar 4.23	Perbandingan nilai fit dan actual Telkom pada model AR(1) .....	57
Gambar 4.24	Perbandingan nilai fit dan actual telkom pada model MA(1) .....	58
Gambar 4.25	Boxplot Indosat .....	59
Gambar 4.25	Boxplot Telkom .....	59

## DAFTAR TABEL

Tabel 4.1	Hasil Simulasi Stock Aggreagtion <i>Long Memory</i> n=500 .....	27
Tabel 4.2	Hasil Simulasi Stock Aggreagtion <i>Long Memory</i> n=2000 .....	29
Tabel 4.3	Hasil Simulasi Stock Aggregation <i>Markov Switching</i> n=500 .....	31
Tabel 4.4	Hasil Simulasi Stock Aggregation <i>Markov Switching</i> n=2000 .....	33
Tabel 4.5	Hasil Simulasi Stock Aggregation <i>Random Shift</i> n=500 .....	35
Tabel 4.6	Hasil Simulasi Stock Aggregation <i>Random Shift</i> n=2000 .....	37
Tabel 4.7	Hasil Simulasi Stock Aggregation <i>Stop Break</i> n=500 .....	39
Tabel 4.8	Hasil Simulasi Stock Aggregation <i>Stop Break</i> n=2000 .....	40
Tabel 4.9	Uji Stationeritas dalam varians .....	45
Tabel 4.10	Estimasi Nilai <i>d</i> dengan <i>m</i> Level Aggregation .....	45
Tabel 4.11	Estimasi Parameter model ARFIMA series data Indosat .....	47
Tabel 4.12	Estimasi Parameter model ARFIMA series data Telkom .....	48
Tabel 4.13	<i>Normality Test</i> Model ARFIMA series data Telkom .....	49
Tabel 4.14	Estimasi Parameter model <i>Markov Switching</i> series data Indosat ....	51
Tabel 4.15	<i>Nomality Test</i> Model <i>Markov Switching</i> series data indosat .....	51
Tabel 4.16	Estimasi Parameter model <i>Markov Switching</i> series data Telkom ...	55
Tabel 4.17	<i>Nomality Test</i> Model <i>Markov Switching</i> series data Telkom .....	56

## DAFTAR LAMPIRAN

Lampiran A	Code Simulasi Stock Aggregation Untuk <i>Long Memory</i> .....	67
Lampiran B	Code Simulasi Stock Aggregation Untuk <i>Markov Switching</i> ....	68
Lampiran C	Code Simulasi Stock Aggregation Level <i>Shift</i> .....	71
Lampiran D	Code Simulasi Stock Aggregation <i>Stop Break</i> .....	73
Lampiran E	Code <i>Evolution Modification0_Hfromdatanonaggregate</i> .....	75
Lampiran F	Code Output Nilai <i>d</i> .....	78
Lampiran G	Ouput ARFIMA Indosat dengan <i>Time Series Modelling</i> .....	79
Lampiran H	Ouput ARFIMA Telkom dengan <i>Time Series Modelling</i> .....	80
Lampiran I	Uji <i>White Noise</i> dan <i>Normality Test</i> Series Indosat .....	83
Lampiran J	Uji <i>White Noise</i> dan <i>Normality Test</i> Series Telkom .....	85
Lampiran K	<i>Markov Switching</i> Dengan TSM Series Indosat .....	89
Lampiran L	Uji <i>Whitenoise</i> Dan Uji <i>Normality Markov Switching</i> Indosat ..	91
Lampiran M	<i>Markov Switching</i> Dengan TSM Series Telkom .....	93
Lampiran N	Uji <i>Whitenoise</i> Dan Uji <i>Normality Markov Switching</i> Telkom ..	95

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Deret waktu (*time series*) merupakan serangkaian data pengamatan yang diamati berdasarkan indeks waktu  $t$  secara berurutan dan dengan interval waktu yang tetap. Setiap pengamatan dapat dinyatakan sebagai *variable*  $Z_t$  yang mempunyai fungsi kepadatan  $F(Z_t)$  dimana  $Z_t$  dapat dipasangkan dengan indeks waktu  $t_i$ , yaitu  $Z_{t_1}, Z_{t_2}, \dots, Z_{t_n}$  dengan  $t_1, t_2, \dots, t_n$  adalah urutan waktu pengamatan (Makridakis, *et.al* (1999)). Suatu rangkaian waktu akan dapat diketahui apakah peristiwa atau gejala tersebut berkembang mengikuti pola-pola perkembangan yang teratur atau tidak. Jika rangkaian waktu menunjukkan pola yang teratur, maka akan dapat dibuat ramalan yang cukup kuat mengenai tingkah laku gejala yang dicatat, dan atas dasar ramalan itulah dapat dibuat rencana-rencana yang cukup untuk dipertanggungjawabkan.

Analisis *time series* dimaksudkan untuk mendapatkan sebuah model matematis yang mampu menggambarkan pola historis data. Model yang baik akan dapat menghasilkan prediksi atau peramalan yang akurat dan handal. Sebuah model *time series* mengasumsikan bahwa beberapa pola atau kombinasi pola akan berulang sepanjang waktu. Jadi, langkah awal pemodelan biasanya dimulai dengan identifikasi pola historis data. Penentuan dan langkah pengidentifikasian mempunyai peranan penting dalam pemodelan sebab pengidentifikasian yang tepat akan mengarahkan pada proses pemodelan yang tepat pula sehingga dapat dihasilkan model yang terbaik. Kesalahan dalam proses identifikasi akan menghasilkan sebuah model palsu yang berujung pada hasil perkiraan yang buruk atau *error* yang tinggi dari hasil peramalan. Oleh sebab itu, statistik uji yang tepat dapat membantu menentukan identifikasi model yang tepat.

Proses *long memory* telah diamati dalam banyak hal, seperti hidrologi, telekomunikasi, ekonomi dan keuangan. *Long memory* adalah salah satu fenomena dalam *time series*, dimana dependensi antara kejadian masih ada dan dapat diamati untuk waktu yang lama. *Long memory* merupakan kondisi dimana

setiap observasi memiliki korelasi yang cukup kuat dengan observasi lainnya meskipun jarak waktu antar observasi cukup jauh. *Long memory* dicirikan oleh plot *Autocorrelation Function* (ACF) yang turun secara *hyperbolic* atau juga dari nilai *difference* yang tidak bulat (*fractional*). Parameter *differencing* ini biasanya diestimasi menggunakan GPH estimator yang diperkenalkan oleh Geweke dan Porter Hudak (1982). Untuk mendeteksi adanya *long memory* pada data, dapat dilakukan estimasi  $d$  dengan menggunakan *Hurst Exponent* maupun uji *Rescaled range statistics* (R/S) dan *Modified rescaled range statistics* (MR/S), *rescaled varians* (V/S), GPH (*long periodgram estimation*) dan lain sebagainya. Pada kenyataannya, indikator parameter fractional differencing maupun uji-uji statistika seperti V/S test, R/S test mempunyai kekuatan yang lemah untuk bisa mendeteksi fenomena *long memory* yang sesungguhnya (Kuswanto dan Sibbertsen (2011)). Dalam penelitian tersebut, model-model lain seperti ESTAR, Markov Switching juga bisa terdeteksi sebagai *long memory* yang akhirnya disebut dengan model *spurious long memory*. Kenyataannya model-model tersebut merupakan model dengan *short memory*. Proses identifikasi yang salah ini bisa menyebabkan kesalahan peramalan karena kesalahan dalam pemilihan model. Oleh karena itu, mengembangkan pengujian yang dapat membedakan *real long memory* dari *spurious long memory* masih sangat penting dalam rangka untuk memiliki pilihan model yang tepat.

Pada penelitian sebelumnya telah dikembangkan untuk menguji *spurious long memory* adalah Ohanissian *et al.* (2005) dan Kuswanto (2011). Kedua peneliti menggunakan sifat invariance dari parameter *long memory* terhadap *temporal aggregation* (*flow aggregation*). Adapun hasil dari penelitian tersebut menunjukkan bahwa *flow aggregation* dapat diterapkan dan dapat membedakan *spurious long memory* dengan tingkat akurasi yang bagus.

Namun, tes yang sudah dikembangkan tidak cukup efisien jika diterapkan pada data hasil *stock aggregasi* (*skip sampling*). Artinya, jika dilakukan pengujian terhadap data hasil stock aggregation dengan tes tersebut, kesimpulan yang didapatkan hanya sebatas prosesnya *long memory* atau *spurious*. Pemangku kebijakan biasanya tertarik untuk mengetahui proses yang terjadi pada level aggregasi yang lebih kecil. Hal ini bisa dijawab dengan mengaplikasikan tes yang

berbasis pada sifat *stock aggregasi*. Penelitian tentang kekuatan tes yang diterapkan pada kasus *stock aggregation* belum pernah dilakukan. Chambers (1998) dan Souza (2008) telah membuktikan secara teori bahwa kedua prosedur agregasi memiliki sifat yang sama tentang invariance dari parameter *long memory*. Pada kenyataannya, kedua agregasi memiliki prosedur sampling yang berbeda sehingga konsekuensinya tidak bisa begitu saja digeneralisasi bahwa kekuatan tes dari *stock aggregation* sama dengan *flow aggregation*. Oleh karena itu, pada penelitian ini akan dilakukan simulasi untuk mempelajari pattern yang terbentuk dari estimasi GPH pada data *long memory* dan *spurious process* yang teragregasi. Estimasi GPH yang diaplikasikan adalah *bias corrected* GPH estimator yang diperkenalkan oleh Souza dan Smith (2005). Diharapkan dari estimator ini, akan didapatkan kesimpulan mengenai sifat *invariance long memory* pada proses tersebut, sehingga hasil simulasi bisa dijadikan sebagai dasar membedakan *long memory* dan *spurious long memory*. Saham merupakan contoh aplikasi data series yang dihasilkan dari proses *stock aggregation* atau *skip sampling*. Hasil simulasi akan divalidasi dengan data harga saham untuk bisa mengetahui apakah kesimpulan yang didapatkan dari simulasi sesuai dengan data empiris.

### 1.2 Perumusan Masalah

Berdasarkan latar belakang tersebut, maka permasalahan yang akan dibahas pada penelitian ini adalah :

1. Bagaimana performansi *bias corrected* GPH terhadap proses *long memory* dan *spurious long memory* yang disampling secara sistematis untuk model ARFIMA  $(p,d,q)$  dan *Markov Switching*.
2. Bagaimana hasil validasi dari simulasi terhadap data return saham LQ45.

### 1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini berdasarkan permasalahan tersebut adalah sebagai berikut :

1. Mendapatkan hasil simulasi pengidentifikasian *Long Memory* dengan prinsip *stock aggregation*, dilihat dari *pattern bias estimator d* terkoreksi

terhadap  $d$  asli untuk data teragregasi stock model ARFIMA  $(p,d,q)$  dan *Markov Switching*.

2. Mengetahui apakah hasil simulasi bisa digeneralisasi untuk kasus riil dimana pemilihan model terbaik dilihat dari nilai AIC terkecil dari model ARFIMA  $(p,d,q)$  dan *Markov Switching*.

#### 1.4 Manfaat Penelitian

Beberapa manfaat yang dapat diperoleh dari penelitian ini adalah sebagai berikut :

1. Bagi pengembangan ilmu pengetahuan, manfaat yang diperoleh yaitu didapatkan pengetahuan baru mengenai kekuatan identifikasi fenomena *Long Memory* dengan metode agregasi.
2. Dalam aplikasi pemodelan saham, agregasi ini dapat dilakukan untuk proses identifikasi fenomena *Long Memory* atau bukan, sehingga dapat ditentukan model terbaik yang menghasilkan pada peramalan yang akurat.

#### 1.5 Batasan Masalah

Proses *spurious long memory* yang disimulasikan dibatasi pada empat model yaitu *Markov switching*, *Level Shift*, *ESTAR*, dan *STOPBREAK* sebagaimana yang dibahas pada Kuswanto (2011), Ohanissian *et al.* (2008). Namun dalam penelitian ini, model *spurious long memory* yang dibahas akan dibatasi pada model *Markov Switching*.

## BAB II TINJAUAN PUSTAKA

### 2.1 Long Memory

*Long memory* ditandai oleh fungsi autokorelasi yang turun lambat secara hiperbolik. Sebaliknya, ACF dari proses jangka pendek turun secara eksponensial (Iglesias, Jorquera dan Parma, 2005). Fungsi autokorelasi antara  $Z_t$  dengan  $Z_{t+k}$  pada proses ARMA  $\{Z_t\}$  yang turun secara cepat atau eksponensial sering dinyatakan sebagai proses memori jangka pendek (*short memory*).

*Long Memory* memiliki arti bila fungsi autokorelasi antara  $Z_t$  dengan  $Z_{t+k}$  turun secara hiperbolik dan lag yang signifikan semakin banyak maka dapat diidentifikasi adanya ketergantungan jangka panjang dalam data yang merupakan ciri dari memori jangka panjang (*long memory*). Suatu *time series stasioner*  $Z_t, t = 1, 2, \dots, N$  tergolong ke dalam *Long Memory* ketika mempunyai fungsi korelasi  $\rho(k)$  berlaku untuk  $k \rightarrow \infty$  adalah sebagai berikut.

$$\lim_{k \rightarrow \infty} \frac{\rho(k)}{C_p(k)^{2d-1}} = 1$$

(2.1)

Adapun  $C_p$  adalah konstanta dan  $d \in (0, 0.5)$  menunjukkan parameter *Long Memory*. Korelasi dari suatu proses *Long Memory* akan cenderung turun perlahan secara hiperbolik. Proses dikatakan *Short Memory* jika  $d \in (-0.5, 0)$ . Pembahasan lebih lanjut tentang *Long Memory* dikemukakan oleh Beran (1994).

### 2.2 Spurious Long Memory dan Implikasinya

Penelitian mengenai *spurious long memory* pada pemodelan *time series* menggunakan proses *Long Memory* dikembangkan oleh Granger dan Joyeux (1980) dan Hosking (1981). Proses *long memory* telah diaplikasikan pada fenomena makroekonomi dan keuangan seperti, Ding *et al.* (1993) menyatakan bahwa pendapatan saham volatilitas baik dijelaskan dengan proses *long Memory*. Dependensi *long memory* secara umum terdeteksi dalam kuadrat atau harga mutlak dari nilai balik modal (*return*). Sibbertsen (2006) mendeteksi perilaku



*Long Memory* dalam volatilitas *return* saham di Jerman. Andersen *et al.* (2003) menggunakan model *Long Memory* (ARFIMA) untuk meramalkan volatilitas direalisasikan untuk nilai tukar Deutschmark / Dolar dan Yen / Dolar. Pada Andersen *et al.* (2001) dan Deo *et al.* (2006), dapat ditemukan aplikasi *long memory* lainnya.

*Spectral density* serta nilai fungsi autokorelasi yang turun secara hiperbolik adalah dua fitur penting dari proses *Long Memory*. Namun, proses *Short Memory* juga dapat menyerupai fitur ini dipengaruhi oleh beberapa faktor yang mengarah *Spurious Long Memory* (Diebold dan Inoue (2001), dan Gouriéroux Jasiak (2001), Lobato dan Savin (1998), Parke (1999)). Secara substansi yang paling penting adalah menentukan apakah observasi tergolong *Real Long Memory* atau tergolong *Spurious Long Memory*. Namun kenyataannya agak sulit untuk membedakan antara *Real Long Memory* dan *Spurious Long Memory*, karena sebagian tes yang ada belum dapat mendeteksi *long memory* secara benar (Granger dan Hyung, 2004).

Pada penelitian *Spurious Long Memory* baik empiris dan teoritis telah menunjukkan bahwa beberapa perubahan struktural model seperti *switching Markov* model Hamilton (1989) dan *Threshold Autoregressiv* (TAR) model Lim dan Tong (1980) dapat menghasilkan perilaku *Long Memory* dalam hal penurunan autokorelasi dan estimasi parameter *Long Memory* ( $d$ ). Simulasi yang dilakukan Kuswanto dan Sibbertsen (2007) menunjukkan bahwa dengan menggunakan tes standar, secara umum proses *Nonlinear Short Memory* dengan mudah dapat menyesatkan *Long Memory* dan sebaliknya. Pembahasan teoritis mengenai sumber *spuriousness* dengan baik dibahas Kuswanto dan Sibbertsen (2008).

### 2.3 Agregasi Temporal dalam *Long Memory Time Series*

*Temporal Aggregation* adalah proses yang diamati pada frekuensi yang lebih kecil dari yang dihasilkan pada proses sebenarnya. Misalkan  $n$  menunjukkan tingkat agregasi. Jika variabel agregasi adalah *stock variable*, maka diamati setiap periode ke- $n$  sementara jika itu adalah *flow variable*, maka jumlahan dari ke- $n$  dan  $n-1$  periode sebelumnya diamati setiap periode ke- $n$ . Definisi formal tentang agregasi diberikan dalam Definisi 1.

**Definisi 1:** Variabel agregat  $Y_t$  diamati sebagai berikut:

- Jika  $X_t$  adalah *stock variable*, maka  $Y_t = X_{nt}$ ,  $t = 1, \dots, T$ .
- Jika  $X_t$  adalah *flow variable*, maka

$$Y_t = \sum_{i=0}^{n-1} X_{nt-i} = \sum_{i=1}^{n-1} B^i X_{nt}, \quad t = 1, \dots, T; T_y = \frac{T_x}{n} \quad (2.2)$$

Perbedaan antara keduanya adalah pada filter *moving average*  $\sum_{i=0}^{n-1} B^i = (1 + B + \dots + B^{n-1})$  diterapkan untuk  $X_t$ . *Stock aggregation* dikenal sebagai *skip sampling* (Souza dan Smith (2002)) atau *sampling sistematis* (Brewer (1973) dan Weiss (1984)). Chambers (1998) dan Teles dan Wei (2002) menggunakan istilah *temporal aggregation* untuk menggambarkan Definisi 1. *Temporal aggregation* sebagaimana didefinisikan termasuk dalam Definisi 1a menyebabkan fenomena lain, dikenal dalam literatur pemrosesan sinyal untuk *continuous-time processes* diamati pada *discrete-time intervals*.

Efek fenomena lain yang timbul dari menggabungkan proses *discrete-time* diberikan dalam teorema berikut (Souza (2008)):

**Teorema 1:** Misalkan  $X_t$  adalah proses *discrete-time stationer* dalam kovarian dengan fungsi kepadatan spektral  $f_x(\omega)$  dan  $Y_t = \sum_{i=0}^{n-1} X_{nt-i} = \sum_{i=0}^{n-1} B^i X_{nt}$ . Maka fungsi kepadatan spektral  $Y_t$ ,  $f_y(\lambda)$  diberikan oleh :

$$f_y(\lambda) = 2\pi \sum_{i=-\frac{n-1}{2}}^{\frac{n-1}{2}} \left[ F_n \left( \frac{\lambda}{n} + \frac{2i\pi}{n} \right) f_x \left( \frac{\lambda}{n} + \frac{2i\pi}{n} \right) \right], \quad -\pi < \lambda < \pi \quad (2.3)$$

Bentuk lain untuk kerapatan spektral  $Y_t$  variabel agregat adalah :

$$f_y(\lambda) = \frac{\sigma_\varepsilon^2}{2n\pi} \sum_{i=0}^{n-1} \left[ \left| 1 - e^{-j(\lambda+i2\pi)/n} \right|^{-2d} W(e^{-j(\lambda+i2\pi)/n}) W(e^{j(\lambda+i2\pi)/n}) \right] \quad (2.4)$$

Dimana,

$$W(B)\varepsilon_t = \sum_{i=0}^{\infty} w_i \varepsilon_{t-i}$$

Kepadatan spektral di atas adalah sifat penting yang membangun invariance dari parameter *Long Memory* untuk agregasi. Oleh karena itu sangat mudah untuk menunjukkan bahwa proposisi berikut ini berlaku:

**Proposisi 1:**

- a. Jika  $X_t$  memenuhi sifat *Long Memory* stationer dengan  $d < 0,5$  maka agregat prosesnya  $Y_t$  juga memenuhi sifat yang sama dengan integrasi yang sama.
- b. Jika  $X_t$  adalah kovarians stasioner dan memenuhi persamaan estimator GPH dengan  $d > 0$  dan memiliki fungsi spektral yang terbatas dan dengan turunan pertama terbatas di lingkungan kelipatan nol dari frekuensi Nyquist ( $2\pi / n$ ), maka  $Y_t$  juga memenuhi sifat serupa dengan  $X_t$  dengan  $d$  integrasi urutan yang sama.
- c. Jika  $X_t$  memenuhi sifat estimator GPH dengan  $d < 0$  dan memiliki fungsi spektral yang positif dan terbatas dan dengan turunan pertama terbatas di lingkungan kelipatan nol dari frekuensi Nyquist, maka  $Y_t$  memenuhi sifat yang sama dengan urutan yang sama integrasi  $d$  jika  $X_t$  adalah *flow variable* tapi tidak jika  $X_t$  adalah *stock variable*.

#### 2.4 Estimator GPH (*Geweke dan Porter-Hudak*)

Estimator GPH diperkenalkan oleh Geweke dan Porter-Hudak (1983) adalah salah satu yang paling populer dan banyak digunakan untuk menguji fraksi integrasi  $d$ . Hal ini didasarkan pada koordinat  $m$  periodogram pertama.

$$I_j = \frac{1}{2\pi N} \left| \sum_{t=1}^N Y_t \exp(i\lambda_j t) \right|^2 \text{ untuk } j = 1, \dots, m \quad (2.5)$$

dimana  $\lambda_j = 2\pi j / N$  dan  $m$  adalah bilangan bulat positif lebih kecil dari  $N$ . Identy adalah untuk memperkirakan *spectral density* dengan periodogram dan mengambil logaritma pada kedua sisi persamaan. Ini memberikan model regresi linier dalam parameter *Long Memory* yang dapat diperkirakan dengan metode kuadrat terkecil.

Estimator diberikan oleh  $-1/2$  kali estimator *Least Square* dari parameter *slope* dalam regresi  $\{\log I_j : j = 1, \dots, m\}$  pada konstanta dan variabel *regress*.

$$\tilde{X}_j = \log |1 - \exp(-i\lambda_j)| = (1/2) \log(2 - 2 \cos \lambda_j) \quad (2.6)$$

Menurut definisi tersebut, estimator GPH adalah :

$$\hat{d}_{GPH} = \frac{-0.5 \sum_{j=1}^m (\tilde{X}_j - \bar{\tilde{X}}) \log I_j}{\sum_{j=1}^m (\tilde{X}_j - \bar{\tilde{X}})^2} \quad (2.7)$$

dimana  $\bar{\tilde{X}} = (1/m) \sum_{j=1}^m \tilde{X}_j$ . Estimator ini dapat dimunculkan dengan menggunakan model:

$$\log I_j = \log c_f - 2d \tilde{X}_j + \log \xi_j \quad (2.8)$$

dimana  $\tilde{X}_j$  menunjukkan frekuensi  $j$ -th Fourier dan  $\xi_j$  adalah error yang berdistribusi dan identik dengan  $-E[\log \xi_j] = 0.577$ , yang dikenal sebagai konstanta *Euler*.

Jika  $X_t$  adalah proses *long memory* dengan kepadatan *spectral*  $f_x(\lambda)$ , maka kepadatan *spectral* agregat proses *long memory* menjadi (Souza,2005) :

$$\log f_{yn}(\lambda) = \frac{1}{n} \sum_{j=0}^{n-1} \left( 2 \sin\left(\frac{\lambda + 2\pi j}{n}\right) \right)^{-2d} g\left(\frac{\lambda + 2\pi j}{n}\right) \quad (2.9)$$

Persamaan (2.9) juga dapat ditulis sebagai berikut :

$$\log f_{yn}(\lambda) = \left( \left( 2 \sin\left(\frac{\lambda}{2n}\right) \right)^{-2d} g\left(\frac{\lambda}{n}\right) \right) H_n(\lambda) \quad (2.10)$$

dimana  $H_n(\lambda_j) = \sum_{i=1}^{n-1} \left( \frac{\sin(i\pi/n)}{\tan(\lambda_j)} + \cos(i\pi n) \right)^{-2d}$ , dengan menggunakan *log* pada kedua sisi pada persamaan diatas dan estimasi mengimplikasikan prosedur *least square* untuk mengestimasi  $d$  akan menjadi estimator GPH agregasi proses *long memory*. Dari ini, jelas bahwa  $H(\cdot)$  mengasilkan bias. Seperti dilihat secara

langsung bahwa  $H(\cdot)$  tergantung kepada  $n$  dan  $d$ . Beberapa literatur mengatakan, Hwang (2008), Souza dan Smith (2003) menemukan bahwa penurunan sampling interval akan mengurangi bias terhadap nol.

Estimator GPH bias terkoreksi diharapkan akan mengurangi bias menjadi nol terlepas dari sampling yang digunakan. Souza (2008) menyampaikan formula bias terkoreksi untuk ARFIMA (0,d,0) (disebut sebagai *formula heuristic*) didasarkan pada GPH estimator. Formula ditentukan sebagai berikut :

$$HB(T, n, d) = \frac{-d[\sum_{j=1}^m (z_j - \bar{z}) \log f_{yn} + 4 \sum_{j=1}^m (\log f_{yn} - \log f_{yn})^2]}{\sum_{j=1}^m (z_{nj} - \bar{z}_n)^2} \frac{\sum_{j=1}^m (z_{nj} - \bar{z}_n) \log H_n(\lambda_j)}{\sum_{j=1}^m (z_{nj} - \bar{z}_n)^2} \quad (2.11)$$

Dimana  $z_j = \log(2 \sin(\lambda_j))^2$ ,  $z_{yn} = z_j + 2 \log f_{yn}$  dengan rata-rata adalah masing-masing  $\bar{z}$  dan  $\bar{z}_{yn}$  secara bersesuaian. Selanjutnya  $f_{yn}$  dan  $H_n(\lambda_j)$  didefinisikan sebagai :

$$\log f_{yn}(\lambda) = \left( \left( 2 \sin \left( \frac{\lambda}{2n} \right) \right) \right)^{-2d} g \left( \frac{\lambda}{n} \right) H_n(\lambda) \quad (2.12)$$

Pada persamaan (2.11) terlihat bahwa bias terkoreksi tergantung pada nilai  $d$ . souza et al.(2009) menyampaikan untuk menggunakan estimator tidak bias secara asymptotic, misalnya estimasi GPH dengan menggunakan sampel penuh dimana berhubungan dengan situasi  $n-1$ . Estimator yang diajukan dengan bias terkoreksi ditentukan sebagai berikut :

$$\hat{d}_n^* = \frac{1}{n^*} \sum_{n-1}^n [\bar{d}_{nj} - HB(T, n, \bar{d}_1)] \quad (2.13)$$

Dimana bentuk  $\bar{d}_n$  adalah rata-rata estimasi dengan menggunakan  $n$  sampling interval dan  $\bar{d}_{nj}$  menjadi estimasi  $d$  yang dihasilkan dari masing-masing sampling interval. Penting untuk dicatat bahwa jumlah seri agregasi harus cukup panjang untuk estimasi dan estimator diatas telah terbukti menjadi estimator yang konsisten.

## 2.5 ARFIMA (*Autoregressive Fractionally Integrated Moving Average*)

Beberapa model data deret waktu yang telah diterapkan seperti *autoregressive* (AR), *moving average* (MA), *autoregressive moving average* (ARMA), *autoregressive integrated moving average* (ARIMA), *seasonal autoregressive integrated moving average* (SARIMA) dan lain sebagainya. Pada tahun 1989, Haslet dan Raftery mengatakan bahwa data dikategorikan sebagai data *long memory* ditandai dengan plot fungsi autokorelasi (*autocorrelation function* (ACF)) yang tidak turun secara eksponensial melainkan menurun secara lambat atau hiperbolik. Fenomena *long memory* dalam deret waktu pertama kali diperkenalkan oleh Hurst (1951,1956). Granger dan Joyeux (1980), serta Hoskings (1981), mengembangkan model *Autoregressive Fractionally Integrated Moving Average* (ARFIMA) untuk memodelkan *long memory* pada data deret waktu. Model ARIMA merupakan suatu metode yang digunakan untuk memodelkan data *time series* yang memiliki ketergantungan jangka pendek (*short memory*), yaitu data dengan periode terpisah jauh diasumsikan tidak berkorelasi. Sebaliknya apabila diantara observasi dengan periode yang terpisah jauh masih mempunyai korelasi yang tinggi, maka data tersebut memiliki ketergantungan jangka panjang (*long memory*). Data *time series* dengan ketergantungan jangka panjang cenderung stasioner dalam mean, sehingga parameter pembeda  $d$  bernilai sangat kecil. Hal tersebut yang menjadi dasar bahwa model ARFIMA, yang pertama kali dikembangkan oleh Granger dan Joyeux (1980), adalah metode yang tepat untuk meramalkan data *time series* dengan sifat *long memory*.

Model ARFIMA dapat mengatasi kelemahan model ARIMA, dimana ARIMA hanya dapat menjelaskan data *short memory* dengan differencing ( $d$ ) bernilai bilangan bulat. Granger dan Joyeux (1980), dalam artikel Moulines dan Soulier tahun 1999, mengatakan bahwa model ARFIMA merupakan model terbaik yang dapat menjelaskan data deret waktu baik berupa *short memory* maupun *long memory* dengan differencing ( $d$ ) dapat bernilai bilangan riil.

Model ARFIMA secara umum sama dengan model ARIMA. Perbedaan ARFIMA dengan model ARIMA terletak pada parameter pembedanya yaitu :

$$\varphi(B)(1-B)^d(Z_t - \mu) = \theta(B)a_t \quad (2.14)$$

dengan,

$t$  = indeks dari pengamatan

$d$  = parameter pembeda (bilangan pecahan)

$\mu$  = rata-rata dari pengamatan

$a_t \sim IIDN(0, \sigma^2)$

$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$  adalah polynomial AR(p)

$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$  adalah polynomial MA(q)

$(1 - B)^d = \sum_{r=0}^{\infty} \binom{d}{r} (-1)^r B^r$  adalah pembeda pecahan

$$\sum_{r=0}^{\infty} \binom{d}{r} (-1)^r B^r = F(-d, 1; 1; B)$$

$$= \sum_{r=0}^{\infty} \frac{\Gamma(r-d)}{\Gamma(r+1)\Gamma(-d)} B^r$$

Dengan  $F$  merupakan fungsi hipergeometrik yang dirumuskan seperti berikut.

$$F(a; b; c; B) = 1 + \frac{ab}{1 \cdot c} B + \frac{a(a+1)b(b+1)}{1 \cdot 2 \cdot c(c+1)} B^2 + \dots \quad (2.15)$$

Dengan nilai  $a, b, c$  adalah bilangan real dan  $B$  adalah operator *Backshift*. Untuk suatu  $d$  yang bernilai pecahan, operator differencing fraksional  $(1-B)^d$  didefinisikan sebagai berikut.

$$(1 - B)^d = 1 + \sum_{r=1}^{\infty} \frac{\Gamma(-d+r)}{\Gamma(-d)r!} B^r \quad (2.16)$$

Jika persamaan  $\lambda_k(d) = \frac{\Gamma(-d+r)}{\Gamma(-d)r!}$  pada persamaan (2.14) dijabarkan untuk berbagai nilai  $r$  maka :

Untuk  $r = 1$ , diperoleh :

$$\frac{\Gamma(-d+1)}{\Gamma(-d)1!} = \frac{-d!}{(-d-1)!1!} = -d$$

Untuk  $r = 2$ , diperoleh :

$$\frac{\Gamma(-d+2)}{\Gamma(-d)2!} = \frac{(-d+1)!}{(-d-1)!2!} = \frac{-d(1-d)}{2}$$

Untuk  $r = 3$ , diperoleh :

$$\frac{\Gamma(-d+3)}{\Gamma(-d)3!} = \frac{(-d+2)!}{(-d-1)!3!} = \frac{-d(1-d)(2-d)}{6}$$

dan seterusnya,

sehingga persamaan (2.15) dapat ditulis kembali menjadi :

$$(1-B)^d = 1 + \sum_{r=1}^{\infty} \lambda_r(d) B^r$$

Dengan,

$$\lambda_0(d) = 1,$$

$$\lambda_1(d) = -d,$$

$$\lambda_2(d) = -\frac{1}{2}d(1-d),$$

$$\lambda_3(d) = \frac{1}{6}d(1-d)(2-d), \text{ dan seterusnya}$$

Apabila persamaan tersebut dijabarkan untuk berbagai nilai  $r$  maka akan dihasilkan persamaan operator differencing fraksional sebagai berikut.

$$(1-B)^d = 1 - dB - \frac{d(1-d)}{2}B^2 - \frac{d(1-d)(2-d)}{6}B^3 + \dots \quad (2.17)$$

Menurut Boutahar dan Khalfaoui (2011), serta Hoskings (1981), karakteristik utama dari sebuah model ARFIMA (p,d,q) adalah sebagai berikut :

1. Jika  $d > -\frac{1}{2}$ , maka  $X_t$  adalah *invertible*.
2. Jika  $d < \frac{1}{2}$ , maka  $X_t$  adalah *stasioner*.
3. Jika  $-\frac{1}{2} < d < 0$ , maka fungsi autokorelasi  $\rho(h)$  menurun secara lebih cepat daripada kasus  $0 < d < \frac{1}{2}$ , model ini disebut *intermediate memory*.



4. Jika  $0 < d < \frac{1}{2}$ , maka  $X_t$  adalah sebuah model *long memory* yang stasioner dimana fungsi autokorelasi menurun secara hiperbolik menuju nol.
5. Jika  $d = \frac{1}{2}$ , maka *spektral density* tidak terbatas pada frekuensi nol.

Selanjutnya setelah didapatkan kemungkinan model ARFIMA, langkah berikutnya yang dilakukan adalah mendapatkan model terbaik dan melakukan cek diagnosa.

## 2.6 Pemilihan Model Terbaik

Kriteria model terbaik digunakan untuk memilih model yang paling baik dalam merepresentasikan kejadiannya. Ada dua kriteria yang diperhatikan, yaitu kriteria *in-sample (training)* menggunakan nilai *Akaike's Information Criterion (AIC)* dan kriteria *out-sample (testing)* menggunakan ukuran *Mean Square Error (MSE)* dan *Mean Absolute Percentage Error (MAPE)*, (Wei, 2006). Untuk memperoleh nilai AIC dapat dituliskan sebagai berikut.

$$AIC(m) = n \ln \hat{\sigma}_\alpha^2 + 2m \quad (2.18)$$

dimana,

$m$  = jumlah parameter yang ditaksir.

$n$  = banyak pengamatan.

$\hat{\sigma}_\alpha^2$  = nilai varians residual.

Sedangkan pada kriteria *out-sample*, ukuran MSE dan MAPE ditentukan pada persamaan berikut ini.

$$MSE = \frac{1}{n} \sum_{t=1}^n (Z_t - \hat{Z}_t)^2 \quad (2.19)$$

$$MAPE = \left( \frac{1}{n} \sum_{t=1}^n \left| \frac{Z_t - \hat{Z}_t}{Z_t} \right| \right) 100\% \quad (2.20)$$

dimana  $n$  adalah banyaknya sampel yang diramalkan. Model terbaik adalah model yang mempunyai nilai MSE dan MAPE terkecil.

Dalam pemodelan data deret waktu sering ditemukan kondisi rata-rata dan varians tidak stasioner. Pada kondisi rata-rata yang tidak stasioner dilakukan *diffrencing* atau biasa ditulis  $(1 - B)^d$  untuk menstasionerkan data. Untuk *short memory* proses, *diffrencing* dilakukan dengan  $d$  bernilai bilangan bulat, sedangkan untuk *long memory* proses, *diffrencing* dilakukan dengan  $d$  bernilai bilangan rill terletak antara  $0 < d < 0,5$ .

## 2.7 Cek Diagnosa

Setelah melalui tahapan estimasi parameter, tahapan selanjutnya dalam pemodelan ARFIMA adalah pengujian terhadap residual yaitu melakukan pengujian apakah residual salih bebas, mempunyai rata-rata nol dan varians konstan. Model ARFIMA sama halnya seperti model ARIMA juga dibangun melalui prosedur yang sistematis, yaitu dimulai dengan identifikasi model dan penaksiran parameter. Selanjutnya ditaksir kecukupan model dengan memeriksa asumsi yaitu asumsi *white noise* dan kenormalan pada residual (Wei, 2006). Untuk mengetahui apakah residual sudah memenuhi asumsi *white noise*, dilakukan pengujian dengan Uji *Ljung-Box-Pierce* (LBQ). Hipotesis yang digunakan antara lain :

$$H_0 : \rho_1 = \rho_2 = \dots = \rho_K = 0$$

$$H_1 : \text{minimal terdapat satu nilai } \rho_k \neq 0 ; k = 1, 2, \dots, K$$

Dengan menggunakan statistik uji  $Q$  yang diuraikan pada persamaan berikut.

$$Q = n(n + 2) \sum_{k=1}^K (n - k)^{-1} \hat{\rho}_k^2 \quad (2.21)$$

Dimana,

$Q$  : statistik uji *Ljung-Box-Pierce*.

$\hat{\rho}_k$  : ACF dari residual pada lag ke- $k$ .

$n$  : jumlah pengamatan.

maka  $H_0$  ditolak jika  $Q > \chi^2_{(\alpha, K-m)}$ , dimana  $K$  adalah maksimum lag dan  $m = p + q$ .

Sedangkan untuk mengetahui apakah residual sudah mengikuti distribusi normal, dilakukan pengujian dengan Uji *Kolmogorov-Smirnov*, dengan hipotesis yang digunakan adalah :

$H_0$  : Residual berdistribusi normal.

$H_1$  : Residual tidak berdistribusi normal.

dengan menggunakan statistik uji  $D$  sebagai berikut.

$$D = \text{Sup}_x |S(x) - F_0(x)| \quad (2.22)$$

dimana,

$S(x)$  : nilai distribusi kumulatif sampel.

$F_0(x)$ : nilai distribusi kumulatif dari distribusi normal

Maka  $H_0$  ditolak jika  $D \geq K_{(1-\alpha, n)}$  dengan  $K_{(1-\alpha, n)}$  adalah nilai tabel *Kolmogorov-Smirnov* pada kuantil  $1 - \alpha$  dan  $n$  jumlah pengamatan. Apabila ternyata model tidak memenuhi kedua asumsi tersebut, maka harus diidentifikasi model yang baru, yang selanjutnya model tersebut diestimasi dan parameternya diuji kembali.

## 2.8 Markov Switching

Model *Markov switching* merupakan salah satu metode dalam menangani adanya “lompatan” model dari *regime* satu ke *regime* lainnya. Konsep *Markov switching* diperkenalkan oleh Hamilton (1989), merupakan salah satu dari model *regime switching* yang paling populer. Model ini terbukti efektif diterapkan pada *nonlinear dynamic* yang biasanya terjadi pada *time series* financial ekonomi. Beberapa penelitian tentang model *Markov Switching* antara lain dilakukan oleh Kim (1994) dan Hamilton (1994). Fitur utama dari model ini adalah untuk memungkinkan proses switch ke *regime* yang berbeda dengan probabilitas transisi tertentu.

$$y_t = \mu_{S_t} + \sigma_{S_t} \varepsilon_t \quad (2.23)$$

Persamaan (2.23) adalah model dasar *Markov Switching* (Timmerman, 2000), dimana  $\varepsilon_t \sim N(0, \sigma_{S_t}^2)$  dan  $S_t = 1, 2, \dots, k$  menunjukkan indikator *state* yang

*unobserved* (tidak teramati), dimana  $k$ -state mengikuti proses *Markov ergodik* yang mengikuti persamaan (2.24). Model ini telah digunakan dalam pekerjaan empiris oleh Engel dan Hamilton (1990).

$$\text{Prob}(S_{t+j} = j | S_t = i, S_{t-1} = k, \dots) = \text{Prob}(S_{t+1} = j | S_t = i) = P_{ij},$$

$$0 \leq P_{ij} \leq 1, \sum_{j=i}^k P_{ij} = 1 \text{ for all } i,$$

(2.24)

dimana  $i, j = 1, 2, \dots, k$  yang menunjukkan terdapat  $k$  kemungkinan *state* atau *regime* yang berbeda dan matriks transisinya sebagai berikut.

$$P = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1k} \\ P_{21} & P_{22} & \dots & P_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ P_{k1} & P_{k2} & \dots & P_{kk} \end{bmatrix}$$

Kemungkinan perubahan dari satu keadaan ke keadaan yang lain dalam proses Markov disebut kemungkinan transisi, ditampilkan dengan matriks diatas.  $k$  adalah jumlah keadaan dalam proses dan  $P_{ij}$  adalah kemungkinan transisi dari keadaan saat  $i$  ke keadaan  $j$ . Jika saat ini berada pada keadaan  $i$  maka baris  $i$  dari tabel di atas berisi angka-angka  $P_{i1}, P_{i2}, \dots, P_{ik}$  merupakan kemungkinan berubah ke keadaan berikutnya. Oleh karena angka tersebut melambangkan kemungkinan, maka semuanya melupakan bilangan non negatif dan tidak lebih dari satu. Secara matematis.

$$0 \leq P_{ij} \leq 1, i = 1, 2, \dots, k$$

$$\sum_{j=i}^k P_{ij} = 1, i = 1, 2, \dots, k$$

Pada Persamaan (2.23) dapat melihat *spurious long memory* (Kuswanto dan Sibbertsen, 2008), walaupun proses ARMA merupakan proses *short memory* dengan ACF yang turun membentuk pola geometrik, namun proses ARMA tertentu memiliki ACF yang turun secara lambat mengikuti proses *long memory*.

$$\phi_{p,St}(B)Y_t = a_t \quad \text{untuk } t = 1, 2, \dots, n$$

$$y_t = \mu_{St} + \phi_{1,St}y_{t-1} + \phi_{2,St}y_{t-2} + \dots + \phi_{p,St}y_{t-p} + a_t$$

$$y_t = \mu_{s_t} + \mathbf{X}_t \boldsymbol{\phi}_{s_t} + a_t \quad (2.25)$$

Pada model *Markov Switching* dengan proses  $AR(p)$  sesuai dengan persamaan (2.22) yang dijelaskan oleh Zivot dan Wang (2006), dimana  $\mathbf{X}_t = (y_{t-1}, y_{t-2}, \dots, y_{t-p})$  dan  $\boldsymbol{\phi}_{s_t}$  adalah vektor koefisien AR berukuran  $p \times 1$ .

Umumnya, jika *states*  $\mathbf{S} = (S_{p+1}, \dots, S_n)$  diketahui, parameter  $\boldsymbol{\Theta}$  model *Markov Switching*  $AR(p)$  yang tidak diketahui dan memuat intersep, maka koefisien AR dan varian residual pada *regime* yang berbeda dapat diestimasi dengan memaksimalkan fungsi *log-likelihood*.

$$L(\boldsymbol{\Theta}|\mathbf{S}) = \sum_{t=p+1}^n \log f(y_t | \mathcal{Y}_{t-1}, S_t) \quad (2.26)$$

dimana  $\mathcal{Y}_{t-1}$  menunjukkan semua informasi yang tersedia pada waktu  $t - 1$  dan memuat semua observasi  $\mathbf{X}_t$  untuk  $j \leq t$ , dan

$$f(y_t | \mathcal{Y}_{t-1}, S_t) \propto \exp \left( -\frac{1}{2} \log \sigma_{s_t}^2 - \frac{[y_t - (\mu_{s_t} + \mathbf{X}_t \boldsymbol{\phi}_{s_t})]^2}{2\sigma_{s_t}^2} \right) \quad (2.27)$$

*States*  $\mathbf{S}$  biasanya tidak terobservasi (tidak teramati) dan harus disimpulkan dari data. Ketika  $\mathbf{S}$  diketahui, parameter model *Markov-switching*  $AR(p)$  diperluas dengan mengikutkan probabilitas transisi  $\mathcal{P}$ . Dengan mengaplikasikan hukum probabilitas total, fungsi *log-likelihood* dapat ditulis pada (2.28).

$$\begin{aligned} L(\boldsymbol{\Theta}) &= \sum_{t=p+1}^n \log f(y_t | \mathcal{Y}_{t-1}) \\ &= \sum_{t=p+1}^n \log \left\{ \sum_{j=1}^k f(y_t | \mathcal{Y}_{t-1}, S_t = j) P(S_t = j | \mathcal{Y}_{t-1}) \right\} \end{aligned} \quad (2.28)$$

dimana  $f(y_t | \mathcal{Y}_{t-1}, S_t = j)$  sesuai dengan (2.24), dan dengan *teorema Bayes* diperoleh probabilitas prediksi  $P(S_t = j | \mathcal{Y}_{t-1})$  dapat ditunjukkan akan menjadi:

$$\begin{aligned}
 P(S_t = j | \mathcal{Y}_{t-1}) &= \sum_{i=1}^k P(S_t = j | S_{t-1} = i, \mathcal{Y}_{t-1}) P(S_{t-1} = i | \mathcal{Y}_{t-1}) \\
 &= \sum_{i=1}^k P_{ij} \frac{f(y_{t-1} | \mathcal{Y}_{t-2}, S_{t-1} = i) P(S_{t-1} = i | \mathcal{Y}_{t-2})}{\sum_{m=1}^k f(y_{t-1} | \mathcal{Y}_{t-2}, S_{t-1} = m) P(S_{t-1} = m | \mathcal{Y}_{t-2})}
 \end{aligned}
 \tag{2.29}$$

Sehingga estimasi dari inisial probabilitas pada setiap *state*  $P(S_{p+1} = i | \mathcal{Y}_{t-2})$  untuk  $i = 1, 2, \dots, k$  dan fungsi *log-likelihood* model *Markov-switching* AR( $p$ ) dapat dihitung secara iterasi berdasarkan (2.25) dan (2.26), sedangkan parameter  $\Theta$  yang tidak diketahui dapat diestimasi dengan menggunakan *maximum likelihood estimation* (MLE).

## 2.9 Data Saham LQ45

Indeks LQ 45 hanya terdiri dari 45 saham yang telah terpilih melalui berbagai kriteria pemilihan, sehingga akan terdiri dari saham-saham dengan likuiditas dan kapitalisasi pasar yang tinggi. Saham-saham pada indeks LQ 45 harus memenuhi kriteria dan melewati seleksi utama sebagai berikut :

1. Masuk dalam ranking 60 besar dari total transaksi saham di pasar reguler (rata-rata nilai transaksi selama 12 bulan terakhir).
2. Ranking berdasar kapitalisasi pasar (rata-rata kapitalisasi pasar selama 12 bulan terakhir).
3. Telah tercatat di BEJ minimum 3 bulan.
4. Keadaan keuangan perusahaan dan prospek pertumbuhannya, frekuensi dan jumlah hari perdagangan transaksi pasar reguler.

Saham-saham yang termasuk didalam LQ 45 terus dipantau dan setiap enam bulan akan diadakan review (awal Februari, dan Agustus). Apabila ada saham yang sudah tidak masuk kriteria maka akan diganti dengan saham lain yang memenuhi syarat. Pemilihan saham - saham LQ 45 harus wajar, oleh karena itu BEJ mempunyai komite penasehat yang terdiri dari para ahli di BAPEPAM, Universitas, dan Profesional di bidang pasar modal.



*Halaman Ini Sengaja Dikosongkan*

## BAB III METODE PENELITIAN

### 3.1 Sumber Data

Data yang digunakan dalam penelitian ini adalah data sekunder, yaitu data harian dari dua saham Indosat dan Telkom, yang termasuk ke dalam saham LQ45 dari tahun 2000 sampai 2011 yang diambil adalah saham-saham bidang telekomunikasi.

### 3.2 Langkah Analisis

Langkah analisis yang dilakukan untuk mencapai tujuan dari penelitian ini adalah sebagai berikut :

1. Melakukan proses simulasi dengan data bangkitan dan mengestimasi GPH terkoreksi dengan  $n = 500$  dan  $n = 2000$ . Simulasi dilakukan pada masing-masing jenis agregasi. Setiap jenis agregasi dilakukan pengulangan simulasi sebanyak 1000 kali pada tiap-tiap level agregasi  $m$ , dimana  $m = 1, 2, \dots, 10$  dan dilakukan untuk setiap fraksi integrasi  $d$  dengan  $d = 0,1; 0,2; 0,3; 0,4$ . Setiap data hasil dari agregasi akan diestimasi nilai parameter  $d$  dan standart deviasinya dengan estimator GPH dengan *bandwith* optimum 0,5 dan 0,8.

Jenis-jenis agregasi yang digunakan adalah :

#### a. *Markov-Switching Process*

Model yang digunakan dalam proses *Markov-switching* adalah :

$$x_t = \begin{cases} \Phi_1 x_{t-1} + \varepsilon_t & \text{jika } \delta_t = 1 \\ \Phi_2 x_{t-1} + \varepsilon_t & \text{jika } \delta_t = 2 \end{cases}$$

Dengan  $\varepsilon_t \sim N(0,1)$  dan peluang  $p_{00}$  dan  $p_{11}$

Model *Markov Switching* yang dibangkitkan ada 3 macam, yaitu:

1. MS 1 : ( $\Phi_1 = 0.5, \Phi_2 = -0.5, P_{00} = 0.9, P_{11} = 0.9, \text{bandwith} = 0.5$ )
2. MS 2 : ( $\Phi_1 = 0.7, \Phi_2 = -0.2, P_{00} = 0.9, P_{11} = 0.9, \text{bandwith} = 0.5$ )
3. MS 3 : ( $\Phi_1 = 0.5, \Phi_2 = -0.4, P_{00} = 0.9, P_{11} = 0.9, \text{bandwith} = 0.5$ )



b. *STOP-BREAK Process*

Model yang digunakan dalam proses *STOP-BREAK* adalah :

$$x_t = \mu_t + \varepsilon_t, \mu_t = \mu_{t-1} + \frac{\varepsilon_{t-1}^2}{\gamma + \varepsilon_{t-1}^2} \varepsilon_{t-1}$$

Dengan  $\varepsilon_t \sim N(0,1)$

Model *Stop Break* yang yang dibangkitkan ada 3 macam, yaitu:

1. *Stop Break 1* : ( $\gamma = 50$ )
2. *Stop Break 2* : ( $\gamma = 100$ )
3. *Stop Break 3* : ( $\gamma = 180$ )

c. *Stationary Random Level Shift Process*

Model yang digunakan dalam proses *Stationary random level shift* adalah :

$$x_t = \mu_t + \varepsilon_t, \mu_t = (1 - j_t)\mu_{t-1} + j_t\varepsilon_t$$

Dengan  $j_t$  mengikuti IID Bernoulli ( $p$ ),  $\varepsilon_t$  dan  $\varepsilon_t$  adalah proses *short memory* dengan mean 0 dan varian  $\sigma_{\varepsilon_t}^2$

Model *Random Level Shift* yang dibangkitkan ada 3 macam, yaitu:

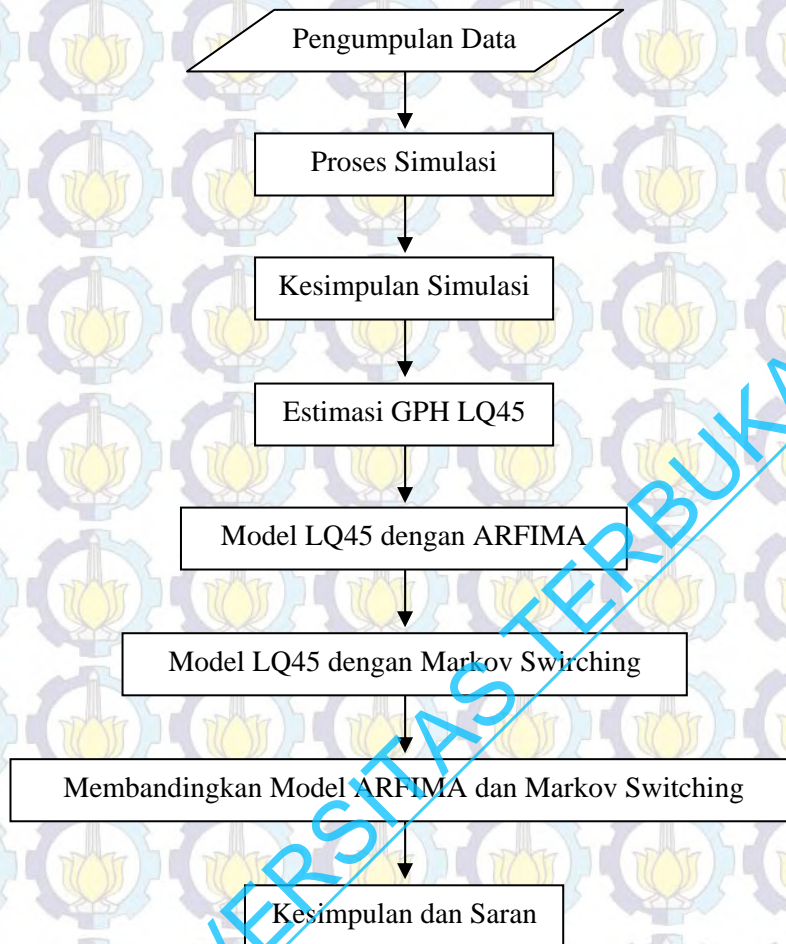
1. *Random Level Shift 1* (Probabilitas = 0.005)
  2. *Random Level Shift 2* (Probabilitas = 0.01)
  3. *Random Level Shift 3* (Probabilitas = 0.1)
2. Mengevaluasi *pattern* yang teramati dari hasil simulasi untuk masing-masing proses
- a. Mengestimasi nilai GPH terkoreksi pada 2 data return saham LQ45 dengan sampling interval 10 serta menduga model *long memory/spurious* berdasarkan hasil simulasi
  - b. Setiap data saham dibuat plot ACP dan PACF
  - c. Pada data saham akan dilakukan agregasi dengan *stock aggregation* dimana level agregasi  $m = 1, 2, \dots, 10$ , dan diestimasi nilai  $d$  serta diamati polanya
  - d. Memodelkan data return saham LQ45 dengan ARFIMA sebagai pendekatan long memory
    - Plot ACF dan PACF
    - Estimasi nilai  $d$  dengan GPH estimator
    - Estimasi parameter AR dan MA

- Pengujian parameter model ARFIMA
  - Pengujian residual untuk diagnostic check
- e. Memodelkan data return saham LQ45 dengan *Markov Switching* sebagai pendekatan *spurious long memory* dengan cara Membentuk dan memperoleh hasil pemodelan *Markov Switching* dengan langkah sebagai berikut.
- Melakukan estimasi probabilitas pada setiap *state*  $P(S_{t+1} = i | Y_{t-2})$  untuk  $i = 1, 2, \dots, k$  seperti persamaan :
 
$$P(S_t = j | Y_{t-1}) = \sum_{i=1}^k P(S_t = j | S_{t-1} = i, Y_{t-1}) P(S_{t-1} = i | Y_{t-1})$$

$$= \sum_{i=1}^k P_{ij} \frac{f(y_{t-1} | Y_{t-2}, S_{t-1} = i) P(S_{t-1} = i | Y_{t-2})}{\sum_{m=1}^k f(y_{t-1} | Y_{t-2}, S_{t-1} = m) P(S_{t-1} = m | Y_{t-2})}$$
  - Melakukan estimasi parameter  $\Theta$  model *Markov Switching* sesuai persamaan
 
$$L(\Theta | S) = \sum_{t=p+1}^n \log f(y_t | Y_{t-1}, S_t)$$
  - Pengujian terhadap parameter *Markov Switching*
  - Pengujian terhadap residual apakah memenuhi asumsi *white noise* dan berdistribusi normal
  - Membentuk model *Markov Switching* sesuai persamaan
 
$$y_t = \mu_{S_t} + X_t \phi_{S_t} + a_t$$
- f. Membandingkan keakurasian hasil pemodelan antara ARFIMA dengan *Markov Switching* dari kedua data saham telekomunikasi tersebut dengan memilih model terbaik berdasarkan nilai AIC terkecil.
- g. Membandingkan hasil simulasi dengan hasil pemodelan data empiris.

### 3.3 Diagram Alir Langkah-Langkah Analisis

Langkah-langkah analisis secara ringkas, dapat dilihat dalam diagram alir di bawah ini :



Gambar 3.1 : Diagram Alir Langkah-Langkah Analisis

## BAB IV ANALISIS DAN PEMBAHASAN

Metode GPH (Geweke and Porter-Hudak) adalah metode penaksiran parameter *long memory* dari suatu rangkaian data *time series*. Kelebihan metode GPH dibandingkan dengan yang lainnya seperti Metode Maksimum *Likelihood* (Sowell,1992), metode *Nonlinier Least Square* (Beran,1995) adalah fleksibilitas dalam penaksiran parameter-nya. Artinya, penaksiran parameter *long memory d* pada metode GPH dapat dilakukan secara langsung tanpa harus melakukan estimasi terhadap parameter model seperti maksimum *likelihood* untuk model ARFIMA (*Autoregressive Fractionally Integrated Moving Average*). Dalam perkembangannya, *property* dari penaksir GPH telah dikembangkan untuk kepentingan yang lain yaitu mendeteksi pola dari suatu data *time series* yang diaggregasi. Modifikasi dari GPH standar telah dilakukan oleh Souza *et al.* (2008) untuk mengurangi bias dari parameter *long memory* pada data *time series* yang tersampling secara acak.

Estimator GPH terkoreksi diharapkan akan mengurangi bias menjadi nol terlepas dari sampling yang digunakan. Souza *et al.*(2008) menyampaikan formula bias terkoreksi untuk ARFIMA (0,d,0) (disebut sebagai *formula heuristic*) didasarkan pada GPH estimator. Formula ditentukan sebagai berikut :

$$HB(T, n, d) = \frac{-d \left[ \sum_{j=1}^m (z_j - \bar{z}) \log f_{y_n} + 4 \sum_{j=1}^m (\log f_{y_n} - \bar{\log f_{y_n}})^2 \right]}{\sum_{j=1}^m (z_{y_j} - \bar{z}_n)^2} \frac{\sum_{j=1}^m (z_{y_j} - \bar{z}_n) \log H_n(\lambda_j)}{\sum_{j=1}^m (z_{y_j} - \bar{z}_n)^2} \quad (4.1)$$

dimana  $z_j = \log(2 \sin(\lambda_j))^2$ ,  $z_{y_n} = z_j + 2 \log f_{y_n}$  dengan rata-rata adalah masing-masing  $\bar{z}$  dan  $\bar{z}_{y_n}$  secara bersesuaian. Selanjutnya  $f_{y_n}$  dan  $H_n(\lambda_j)$  didefinisikan sebagai :

$$\log f_{y_n}(\lambda) = \left( \left( 2 \sin \left( \frac{\lambda}{2n} \right) \right) \right)^{-2d} g \left( \frac{\lambda}{n} \right) H_n(\lambda) \quad (4.2)$$

Pada persamaan (4.1) dapat kita lihat bahwa bias terkoreksi tergantung pada nilai  $d$ . Souza *et al.*(2009) menyampaikan untuk menggunakan estimator tidak bias secara *asymptotic*, misalnya estimasi GPH dengan menggunakan sample penuh

dimana berhubungan dengan situasi  $n-1$ . Estimator yang diajukan dengan bias terkoreksi ditentukan sebagai berikut :

$$\hat{d}_n^* = \frac{1}{n^*} \sum_{n=1}^n [\bar{d}_{n_j} - HB(T, n, \bar{d}_1)] \quad (4.3)$$

Dimana bentuk  $\bar{d}_n$  adalah rata-rata estimasi dengan menggunakan  $n$  sampling interval dan  $\bar{d}_{n_j}$  menjadi estimasi  $d$  yang dihasilkan dari masing-masing sampling interval. Penting untuk dicatat bahwa jumlah seri agregasi harus cukup panjang untuk estimasi dan estimator diatas telah terbukti menjadi estimator yang konsisten.

Formula di atas diturunkan dari persamaan GPH standar yang diaplikasikan pada data agregasi sehingga muncullah komponen bias  $H(\cdot)$ . Dari formula di atas, sulit ditentukan pola hubungan antara parameter  $d$  dan panjang agregasi dengan estimator GPH. Hal ini dikarenakan kompleksitas dari formula yang ada, sehingga evaluasi terhadap bias yang ditimbulkan oleh estimator paling memungkinkan dilakukan melalui simulasi.

Estimator GPH berdasarkan penelitian yang dilakukan oleh Souza *et al.* (2005), sangat dipengaruhi oleh pemilihan *bandwith* yang tepat. Pemilihan *bandwith* mempengaruhi bias estimasi yang dihasilkan. Berdasarkan hasil simulasi pada penelitian tersebut, bias yang dihasilkan oleh GPH dengan *bandwith* 0.5 dan 0.8 menunjukkan bahwa bias yang dihasilkan GPH saat menggunakan *bandwith* 0.5 lebih kecil daripada *bandwith* 0.8 seiring bertambahnya jumlah sampel, sehingga pada penelitian ini diputuskan untuk menggunakan *bandwith* 0.5.

Salah satu cara untuk mendeteksi fenomena *long memory* adalah dengan *temporal aggregation*, dimana sifat *temporal aggregation* terbagi menjadi dua, yaitu *flow aggregation* dan *stock aggregation*. Kedua agregasi tersebut telah diyakini dapat mendeteksi fenomena *long memory*, tetapi kedua agregasi tersebut dikembangkan dari prosedur sampling yang berbeda sehingga performa dari kedua sifat ini tidak bisa disamakan. Dalam tesis ini akan dilakukan simulasi

untuk mengetahui kemampuan dalam mendeteksi *long memory* pada *stock aggregation* atau *skip sampling*.

#### 4.1 Simulasi Stock Aggregation (*skip sampling*)

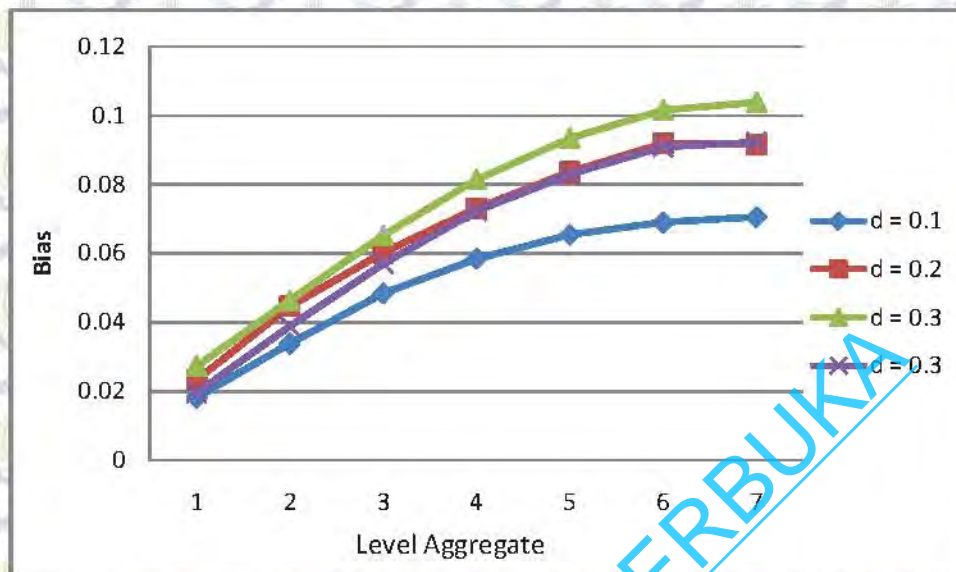
Berbeda dengan *flow aggregation*, prosedur dari *stock aggregation* sama seperti *skip sampling* atau sampling sistematis. Data agregasi diambil dari data asli pada kelipatan level unit waktunya. Secara otomatis prosedur ini akan menghilangkan informasi dari data asli yang bukan merupakan kelipatan dari level unit waktu yang ditentukan. Hal ini akan berbeda dengan data *flow aggregation* dimana tidak ada data yang dihilangkan sehingga informasi dari data aslinya masih ada pada data teragregasi. Namun dalam penelitian Souza *et al.* (2008) dan Chambers *et al.* (1998) telah dibuktikan secara matematis bahwa kedua macam agregasi mempunyai propertis yang sama yaitu jika data aslinya adalah *long Memory*, maka parameter  $d$  yang diestimasi dengan GPH akan tetap. Sebagaimana telah ditunjukkan oleh Souza, untuk hasil estimasi GPH stock aggregation dapat dilihat pada penelitian tersebut. Berikut hasil estimasi GPH terkoreksi (GPH1) dan GPH tak terkoreksi (GPH2) untuk data *long memory* yang di *skip sampling*.

Tabel 4.1 Hasil Simulasi Stock Aggregation *Long Memory* dengan  $n = 500$

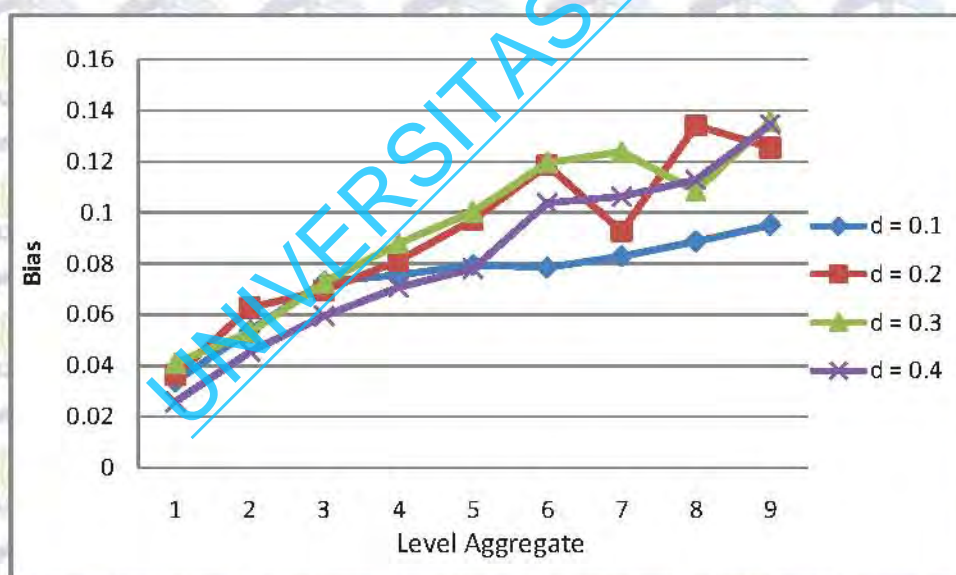
m	Bandwith = 0.5							
	0.1		0.2		0.3		0.4	
	GPH 1	GPH 2	GPH 1	GPH 2	GPH 1	GPH 2	GPH 1	GPH 2
1	0.1047	0.1047	0.1990	0.1990	0.3073	0.3073	0.4107	0.4107
2	0.0865	0.0708	0.1756	0.1627	0.2800	0.2662	0.3912	0.3848
3	0.0708	0.0509	0.1543	0.1362	0.2609	0.2542	0.3718	0.3652
4	0.0563	0.0317	0.1387	0.1294	0.2420	0.2347	0.3538	0.3512
5	0.0463	0.0291	0.1259	0.1178	0.2259	0.2194	0.3384	0.3400
6	0.0393	0.0252	0.1154	0.1017	0.2139	0.2069	0.3277	0.3326
7	0.0356	0.0262	0.1071	0.0803	0.2058	0.1878	0.3199	0.3069
8	0.0341	0.0218	0.1074	0.1063	0.2035	0.1834	0.3184	0.3043
9	NaN	0.0159	NaN	0.0648	NaN	0.1989	NaN	0.2979
10	NaN	0.0095	NaN	0.0735	NaN	0.1716	NaN	0.2761

Dari table di atas, GPH terkoreksi hanya mampu mengestimasi sampai dengan level agregasi 8. Hal ini dikarenakan jumlah sampel yang kecil, sehingga dengan agregasi 10, hanya didapatkan 500 sampel yang mana tidak mencukupi untuk estimasi GPH terkoreksi. Salah satu komponen penting dari untuk

mendekteksi pattern hasil estimasi adalah bias estimasi. Plot dari hasil estimasi untuk lebih jelas menunjukkan bias yang terjadi disajikan pada Gambar 4.1 dan Gambar 4.2.



Gambar 4.1. GPH Terkoreksi untuk n = 500



Gambar 4.2. GPH Tak Terkoreksi untuk n = 500

Pada Tabel 4.1, Gambar 4.1 serta Gambar 4.2, dijelaskan bahwa untuk data dengan level *aggregation* yang semakin besar akan menghasilkan bias yang cenderung lebih besar juga. Namun, terdapat perbedaan jelas dari kedua plot diatas, dimana bias yang dihasilkan oleh GPH terkoreksi cenderung monoton naik seiring dengan naiknya level aggregasi. Begitu juga dengan bias hasil dari GPH

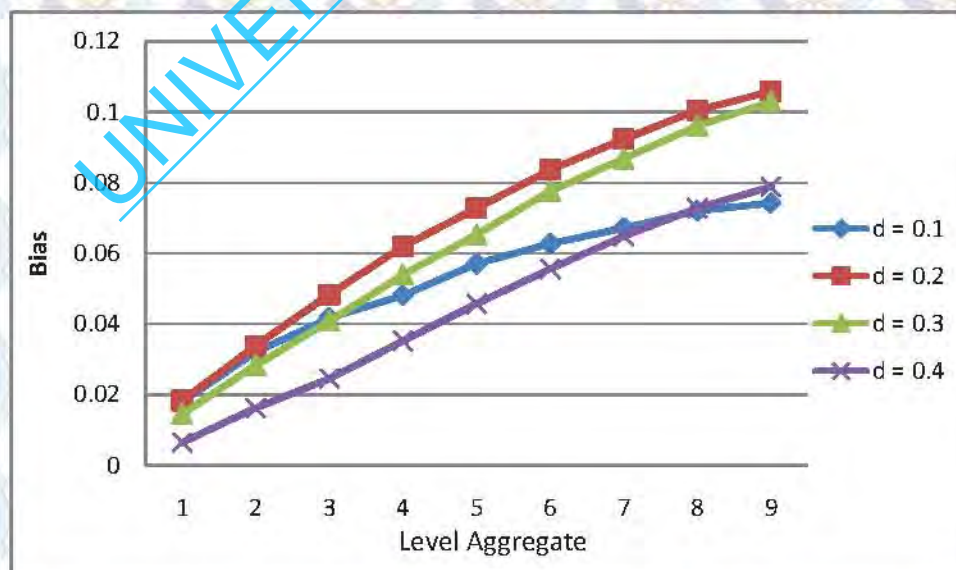
tak terkoreksi. Pada kondisi  $m = 10$  menghasilkan estimasi  $d$  rata-rata yang selisihnya paling besar dengan estimasi  $d$  saat  $m = 1$  atau saat kondisi *no aggregation*. Dari gambar di atas, dapat dilihat pula bahwa parameter *long memory* tidak dominan dalam mempengaruhi besaran bias.

Untuk melihat apakah hasil di atas apakah cukup konsisten, selanjutnya akan disimulasikan dengan sampel yang lebih besar yaitu 2000. Berikut adalah hasil simulasi penggunaan *stock aggregation*.

Tabel 4.2 Hasil Simulasi Stock Aggregation *Long Memory* dengan  $n = 2000$

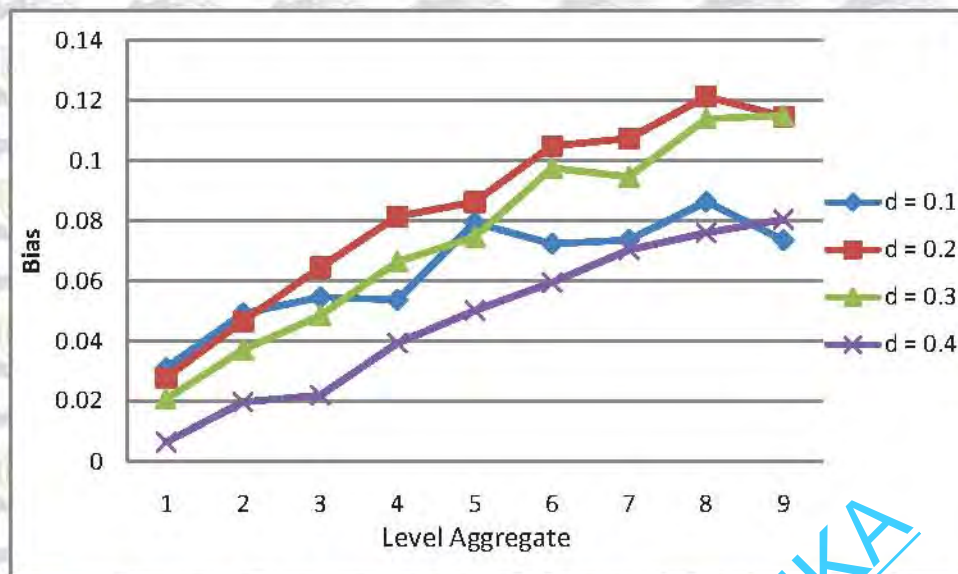
m	Bandwith = 0.5							
	0.1		0.2		0.3		0.4	
	GPH 1	GPH 2	GPH 1	GPH 2	GPH 1	GPH 2	GPH 1	GPH 2
1	0.1001	0.1001	0.2065	0.2065	0.3039	0.3039	0.4032	0.4032
2	0.0820	0.0691	0.1881	0.1786	0.2894	0.2833	0.3968	0.3969
3	0.0680	0.0509	0.1726	0.1599	0.2756	0.2669	0.3870	0.3835
4	0.0583	0.0455	0.1581	0.1420	0.2631	0.2554	0.3787	0.3812
5	0.0519	0.0464	0.1445	0.1249	0.2500	0.2375	0.3680	0.3639
6	0.0430	0.0208	0.1337	0.1202	0.2386	0.2294	0.3575	0.3531
7	0.0373	0.0277	0.1228	0.1016	0.2263	0.2064	0.3477	0.3438
8	0.0329	0.0265	0.1141	0.0991	0.2171	0.2093	0.3384	0.3329
9	0.0280	0.0137	0.1059	0.0851	0.2078	0.1900	0.3304	0.3272
10	0.0258	0.0265	0.1006	0.0918	0.2010	0.1890	0.3244	0.3228

Plot dari hasil simulasi untuk lebih jelas menunjukkan bias yang terjadi disajikan pada Gambar 4.3 dan Gambar 4.4.



Gambar 4.3. GPH Terkoreksi untuk  $n = 2000$





Gambar 4.4. GPH Tak Terkoreksi untuk  $n = 2000$

Tabel 4.2, Gambar 4.3 dan Gambar 4.4, jelas menunjukkan bahwa dengan  $n = 2000$ , pada bandwidth 0,5 menghasilkan bias yang cukup besar. Hal ini ditunjukkan dengan rata-rata parameter  $d$  yang dihasilkan untuk masing-masing level agregasi berbeda cukup jauh. Pada semua setting simulasi dengan estimator GPH terkoreksi, teramat jelas bahwa level bias naik dengan naiknya level agregasi, Bias yang dihasilkan oleh GPH tak terkoreksi juga monoton naik, namun dengan besaran yang tidak konstan.

#### 4.2 Simulasi *Stock Aggregation* pada model *spurious long memory*

Tujuan utama dari penelitian ini adalah mengetahui performa dari *temporal aggregation* untuk mendeteksi *Long Memory* dan *Spurious Long Memory*, yang mana *spurious long memory* yang dibahas adalah beberapa model *time series nonlinier dan level shift*, sebagaimana yang telah dibuktikan dalam penelitian-penelitian sebelumnya oleh Sibbertsen (2004), Kuswanto dan Sibbertsen (2011).

##### 4.2.1 Model Markov Switching

Model *Spurious Long Memory* pertama yang dibahas dalam tesis ini adalah Markov Switching. Hasil dengan menggunakan sifat *flow aggregation* sudah dilakukan oleh Kuswanto dan Sibbertsen (2011) namun untuk sifat *stock aggregation* belum pernah dilakukan pada penelitian-penelitian sebelumnya. Oleh

karena itu, berikut akan disajikan hasil simulasi penerapan GPH baik terkoreksi maupun tidak terkoreksi pada data *stock aggregation* dalam mendeteksi model *Markov Switching*. Model yang digunakan dalam proses *Markov Switching* adalah:

$$x_t = \begin{cases} \Phi_1 x_{t-1} + \varepsilon_t & \text{jika } \delta_t = 1 \\ \Phi_2 x_{t-1} + \varepsilon_t & \text{jika } \delta_t = 2 \end{cases}$$

dengan  $\varepsilon_t \sim N(0,1)$  dan peluang  $P_{00}$  dan  $P_{11}$ .

Model *Markov Switching* yang dibangkitkan ada 3 macam, yaitu :

1. MS 1 : MS ( $\Phi_1 = 0.5, \Phi_2 = -0.5, P_{00} = 0.9, P_{11} = 0.9, \text{bandwidth} = 0.5$ )
2. MS 2 : MS ( $\Phi_1 = 0.7, \Phi_2 = -0.2, P_{00} = 0.9, P_{11} = 0.9, \text{bandwidth} = 0.5$ )
3. MS 3 : MS ( $\Phi_1 = 0.5, \Phi_2 = -0.4, P_{00} = 0.9, P_{11} = 0.9, \text{bandwidth} = 0.5$ )

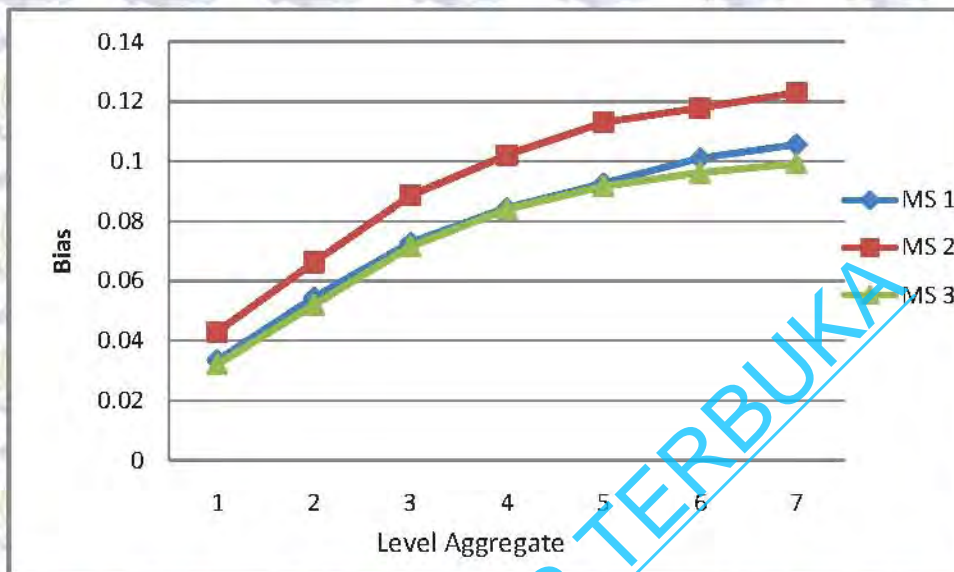
Setting parameter untuk model *Markov Switching* di atas akan menghasilkan serangkaian data *time series* yang persisten, yang merupakan salah satu ciri dari *spurious long memory*. Dalam hal ini, parameter *transition probability* yang sangat besar menyebabkan *switching* dari *regime 1* ke *regime 2* menjadi persisten. Berikut disajikan hasil estimasi GPH untuk data *Markov switching* yang telah teragregasi secara *skip sampling*.

Tabel 4.3 Simulasi *Stock Aggregation* untuk Model *Markov Switching*  $n = 500$

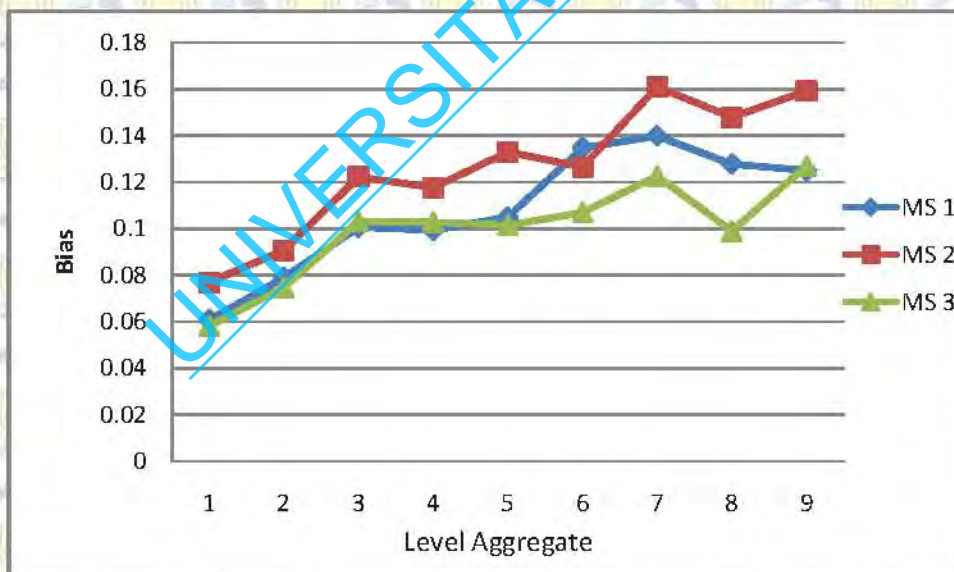
m	bandwidth = 0.5					
	$\Phi_1 = 0.5$ $\Phi_2 = -0.5$ $P_{00} = 0.9$ $P_{11} = 0.9$	$\Phi_1 = 0.5$ $\Phi_2 = -0.5$ $P_{00} = 0.9$ $P_{11} = 0.9$	$\Phi_1 = 0.7$ $\Phi_2 = -0.2$ $P_{00} = 0.9$ $P_{11} = 0.9$	$\Phi_1 = 0.7$ $\Phi_2 = -0.2$ $P_{00} = 0.9$ $P_{11} = 0.9$	$\Phi_1 = 0.5$ $\Phi_2 = -0.4$ $P_{00} = 0.9$ $P_{11} = 0.9$	$\Phi_1 = 0.5$ $\Phi_2 = -0.4$ $P_{00} = 0.9$ $P_{11} = 0.9$
	GPH 1	GPH 2	GPH 1	GPH 2	GPH 1	GPH 2
1	0.1404	0.1404	0.1715	0.1715	0.1419	0.1419
2	0.1069	0.0797	0.1285	0.0945	0.1098	0.0837
3	0.0859	0.0614	0.1053	0.0808	0.0899	0.0671
4	0.0676	0.0397	0.0830	0.0493	0.0704	0.0387
5	0.0559	0.0410	0.0694	0.0538	0.0579	0.0390
6	0.0477	0.0354	0.0585	0.0386	0.0503	0.0404
7	0.0393	0.0057	0.0536	0.0449	0.0457	0.0349
8	0.0348	0.0005	0.0486	0.0106	0.0427	0.0192
9	NaN	0.0126	NaN	0.0236	NaN	0.0429
10	NaN	0.0153	NaN	0.0123	NaN	0.0148

Seperti halnya pada simulasi sebelumnya, untuk jumlah sampel 500 tidak mencukupi dalam estimasi parameter GPH terkoreksi jika diagregasi 9 atau 10.

Pemilihan parameter model *Markov Switching* di atas sudah tepat dalam arti telah mampu menghasilkan parameter  $d$  yang berada pada domain *long memory*. Hal ini terlihat dari parameter  $d$  pada agregasi 1. Plot dari hasil simulasi untuk lebih jelas menunjukkan bias yang terjadi disajikan pada Gambar 4.5 dan Gambar 4.6



Gambar 4.5. GPH Terkoreksi untuk n = 500



Gambar 4.6. GPH Tak Terkoreksi untuk n = 500

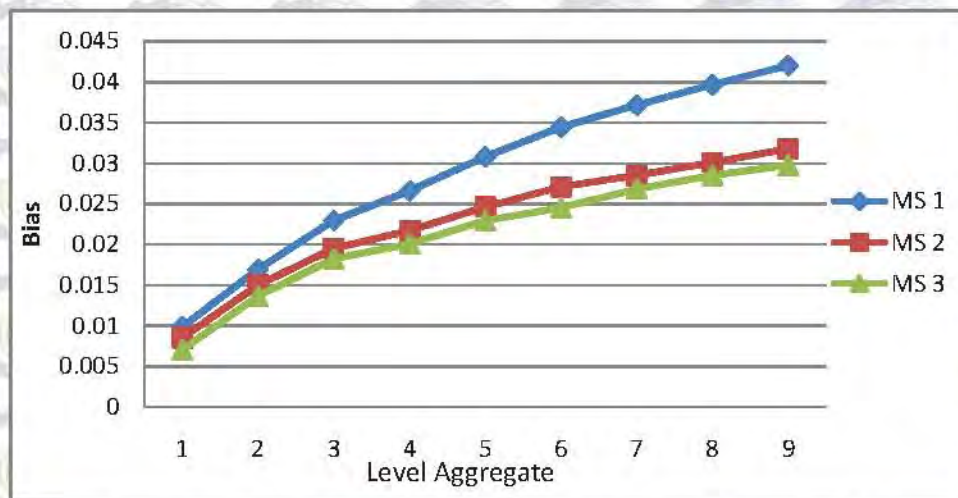
Model *markov switching* merupakan salah satu metode dalam menangani adanya lompatan model dari *regime* satu ke *regime* yang lainnya. Tabel 4.3 menunjukkan bukti bahwa model *markov switching* secara umum menghasilkan parameter integrasi  $d$  cenderung mendekati nol (*short memory*). Penggunaan

properti *stock aggregation* dalam kasus ini dapat menyebabkan nilai  $d$  yang dihasilkan naik dan turun atau tidak monoton. Sebagaimana yang ditampilkan pada Gambar 4.6, jelas menunjukkan bahwa pada GPH Terkoreksi terlihat bahwa tidak adanya trend tertentu dari estimasi parameter  $d$  seiring dengan naiknya level aggregasi., tetapi pada GPH tak terkoreksi tidak terlihat trend yang monoton seiring dengan bertambahnya level aggregasi. Selanjutnya akan disimulasikan dengan sampel yang lebih besar yaitu 2000. Berikut adalah hasil simulasi penggunaan *stock aggregation* pada bangkitan model *markov switching* dengan sampel 2000.

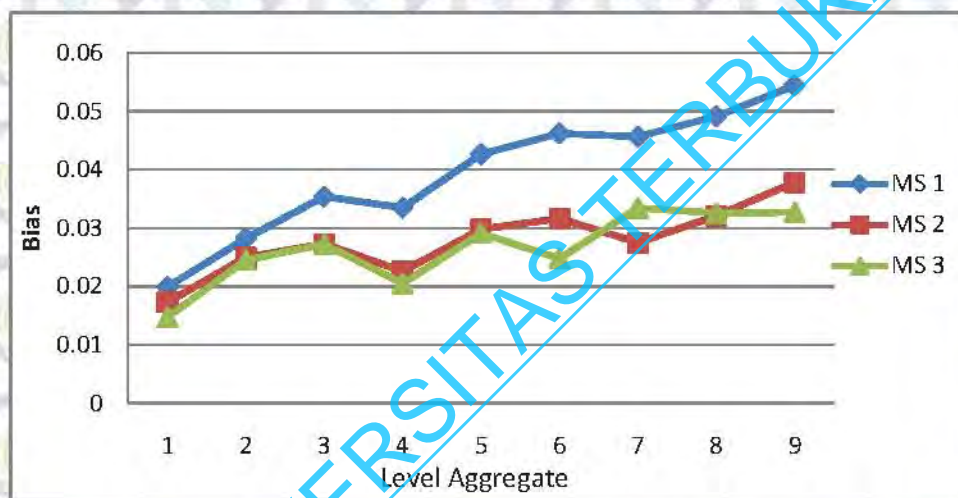
Tabel 4.4 Simulasi *Stock Aggregation* untuk Model Markov Switching  $n = 2000$

m	Bandwith = 0.5					
	$\Phi_1 = 0.5$	$\Phi_1 = 0.5$	$\Phi_1 = 0.7$	$\Phi_1 = 0.7$	$\Phi_1 = 0.5$	$\Phi_1 = 0.5$
	$\Phi_2 = -0.5$	$\Phi_2 = -0.5$	$\Phi_2 = -0.2$	$\Phi_2 = -0.2$	$\Phi_2 = -0.4$	$\Phi_2 = -0.4$
	$P_{00} = 0.9$	$P_{00} = 0.9$	$P_{00} = 0.9$	$P_{00} = 0.9$	$P_{00} = 0.9$	$P_{00} = 0.9$
	$P_{11} = 0.9$	$P_{11} = 0.9$	$P_{11} = 0.9$	$P_{11} = 0.9$	$P_{11} = 0.9$	$P_{11} = 0.9$
	GPH 1	GPH 2	GPH 1	GPH 2	GPH 1	GPH 2
1	0.0460	0.04599	0.0473	0.04729	0.0424	0.04244
2	0.0361	0.02607	0.0388	0.03002	0.0354	0.02772
3	0.0290	0.01779	0.0323	0.02232	0.0288	0.01791
4	0.0230	0.01059	0.0278	0.01997	0.0242	0.01519
5	0.0194	0.01250	0.0256	0.02470	0.0224	0.02196
6	0.0152	0.00333	0.0227	0.01749	0.0195	0.01335
7	0.0115	-0.00023	0.0202	0.01564	0.0179	0.01767
8	0.0088	0.00033	0.0188	0.01978	0.0156	0.00899
9	0.0063	-0.00316	0.0172	0.01520	0.0139	0.00986
10	0.0040	-0.00842	0.0155	0.00947	0.0127	0.00974

Plot dari hasil simulasi untuk lebih jelas menunjukkan bias yang terjadi disajikan pada Gambar 4.7 dan Gambar 4.8.



Gambar 4.7. GPH Terkoreksi untuk n = 2000



Gambar 4.8. GPH Tak Terkoreksi untuk n = 2000

Tabel 4.4 kembali menunjukkan sekaligus memperkuat keterangan bahwa model *markov switching* menghasilkan *pattern* yang berbeda untuk kedua macam GPH dengan harapan akan semakin memperkuat kesimpulan di awal saat mensimulasikan *markov switching* dengan sampel sebanyak 500. Gambar 4.4 menunjukkan bahwa penambahan jumlah sampel  $n$  tidak memberikan efek signifikan terhadap hasil estimasi parameter  $d$  ketika diaggregasi, plot yang ditampilkan memang jelas tetap menunjukkan bahwa tidak adanya pola yang jelas (random) terhadap naiknya level aggregasi untuk GPH tidak terkoreksi.

#### 4.2.2 Model Stationary Random Level Shift Process

Model kedua yang disimulasikan adalah proses Stationary random level shift adalah :

$$x_t = \mu_t + \varepsilon_t, \mu_t = (1 - j_t)\mu_{t-1} + j_t\varepsilon_t$$

dengan  $j_t$  mengikuti IID Bernoulli ( $p$ ),  $\varepsilon_t$  adalah proses *short memory* dengan mean 0 dan varian  $\sigma_{\varepsilon_t}^2$ . Model ini telah banyak dibuktikan mampu menghasilkan parameter  $d$  yang *spurious*. Dari model di atas, parameter yang paling menentukan pergerakan seriesnya adalah parameter  $p$ , sehingga pembangkitan data simulasi difokuskan pada  $p$  yang berbeda.

Model *random shift* yang dibangkitkan ada 3 macam dengan probabilitas dan *bandwith* yang berbeda, yaitu :

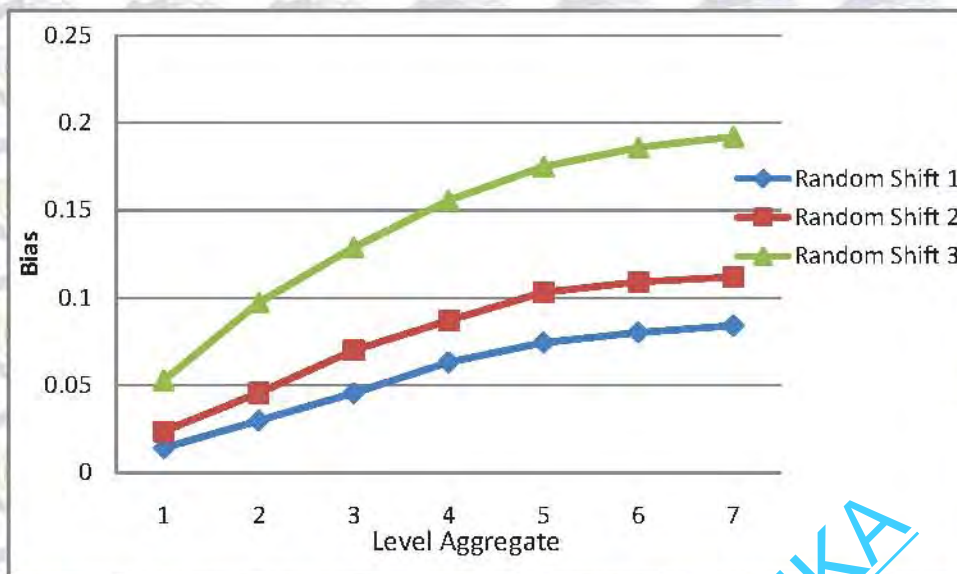
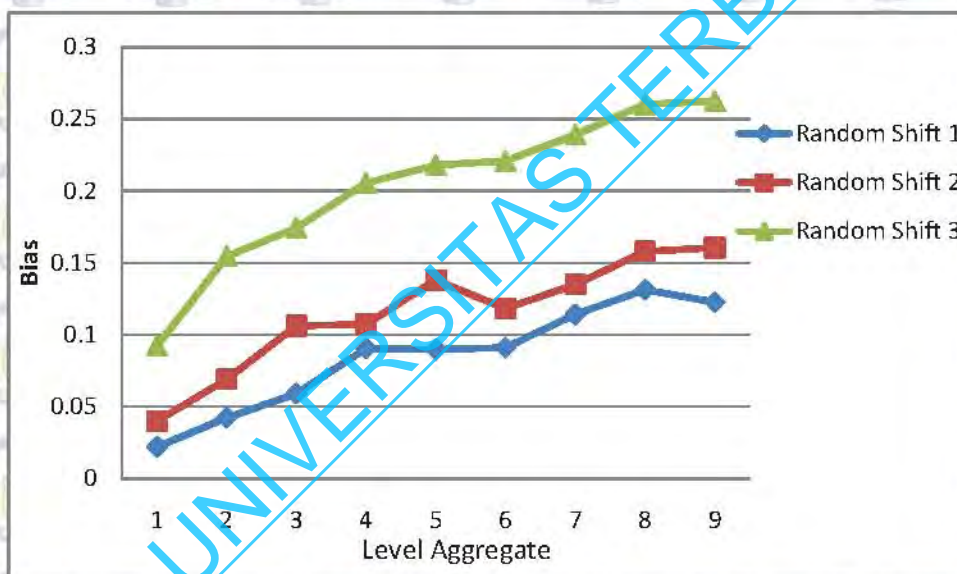
1. Random Level Shift 1 : RS (Probabilitas = 0.005)
2. Random Level Shift 2 : RS (Probabilitas = 0.01)
3. Random Level Shift 3 : RS (Probabilitas = 0.1)

Tabel 4.5 Simulasi *Stock Aggregation* untuk Model *Random Shift*  $n = 500$

m	Bandwith = 0.5					
	0.005		0.01		0.1	
	GPH 1	GPH 2	GPH 1	GPH 2	GPH 1	GPH 2
1	0.4864	0.4864	0.6053	0.6053	0.3222	0.3222
2	0.4722	0.4645	0.5816	0.5655	0.2694	0.2302
3	0.4566	0.4442	0.5596	0.5365	0.2249	0.1678
4	0.4408	0.4274	0.5353	0.4992	0.1934	0.1479
5	0.4233	0.3962	0.5183	0.4980	0.1664	0.1167
6	0.4120	0.3966	0.5022	0.4673	0.1472	0.1044
7	0.4062	0.3954	0.4962	0.4872	0.1362	0.1013
8	0.4023	0.3723	0.4933	0.4705	0.1301	0.0832
9	NaN	0.3552	NaN	0.4474	NaN	0.0620
10	NaN	0.3638	NaN	0.4448	NaN	0.0600

Terlihat jelas dari hasil estimasi parameter di atas, bahwa model *random level shift* dengan parameter di atas mampu menghasilkan parameter  $d$  yang termasuk *long memory*. Namun, untuk  $p = 0.01$ , didapatkan  $d$  yang merupakan *long memory nonstasioner*. Untuk memudahkan evaluasi hasil simulasi, akan ditampilkan plot bias estimator.

Plot dari hasil simulasi untuk lebih jelas menunjukkan bias yang terjadi disajikan pada Gambar 4.9 dan Gambar 4.10.

Gambar 4.9. GPH Terkoreksi untuk  $n = 500$ Gambar 4.10. GPH Tak Terkoreksi untuk  $n = 500$ 

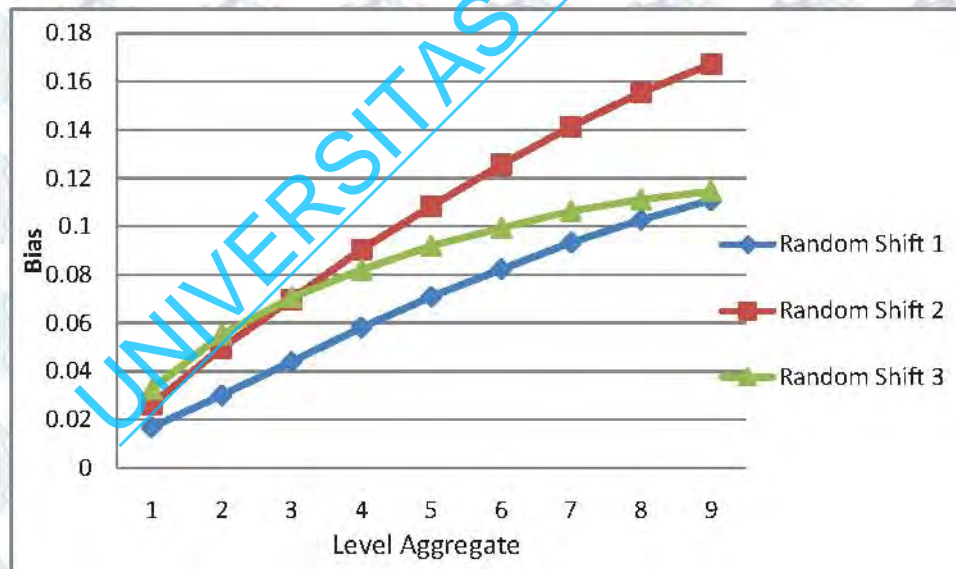
Pada model *random level shift* pertama kali diperkenalkan oleh Chen dan Tiao (1990), pada sampel 500 baik stasioner maupun tidak stasioner, pengujian bekerja dengan baik, dimana terlihat bahwa dengan penambahan jumlah level tidak memberikan efek signifikan terhadap hasil estimasi parameter  $d$  ketika diaggregasi, plot yang ditampilkan memang jelas menunjukkan bahwa tidak adanya pola yang jelas (random) terhadap naiknya level agregasi.

Selanjutnya akan disimulasikan dengan sampel yang lebih besar yaitu 2000. Berikut adalah hasil simulasi penggunaan *stock aggregation* pada bangkitan model *random shift* dengan sampel 2000.

Tabel 4.6 Simulasi *Stock Aggregation* untuk Model *random shift*  $n = 2000$

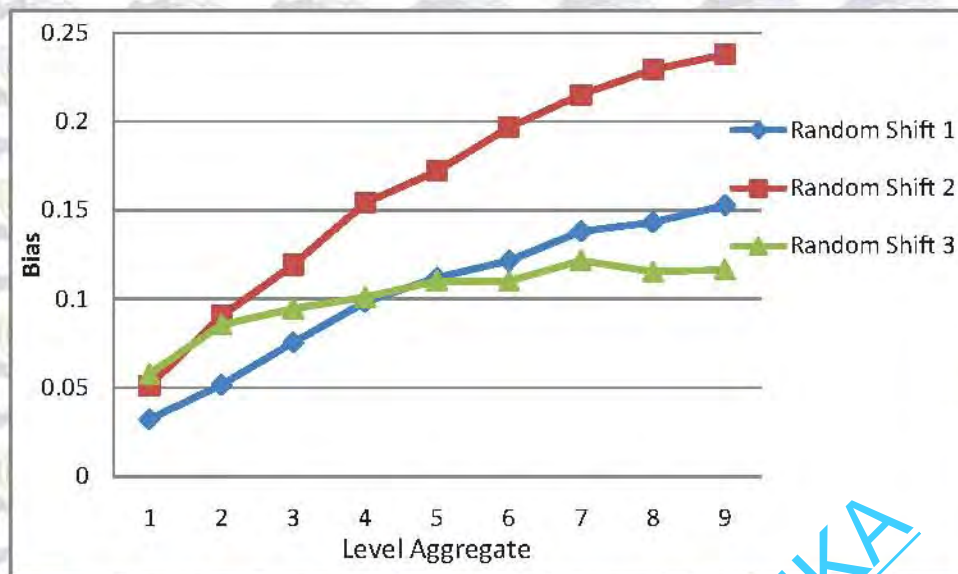
m	Bandwith = 0.5					
	0.005		0.01		0.1	
	GPH 1	GPH 2	GPH 1	GPH 2	GPH 1	GPH 2
1	0.7232	0.7232	0.6912	0.6912	0.1433	0.1433
2	0.7063	0.6909	0.6647	0.6401	0.1107	0.0855
3	0.6931	0.6716	0.6415	0.6008	0.0880	0.0579
4	0.6792	0.6477	0.6212	0.5719	0.0727	0.0488
5	0.6650	0.6249	0.6006	0.5371	0.0612	0.0422
6	0.6522	0.6112	0.5827	0.5190	0.0513	0.0332
7	0.6409	0.6017	0.5655	0.4945	0.0439	0.0332
8	0.6298	0.5851	0.5497	0.4759	0.0368	0.0217
9	0.6204	0.5800	0.5356	0.4619	0.0321	0.0279
10	0.6123	0.5705	0.5239	0.4534	0.0287	0.0268

Plot dari hasil simulasi untuk lebih jelas menunjukkan bias yang terjadi disajikan pada Gambar 4.11 dan Gambar 4.12.



Gambar 4.11. GPH Terkoreksi untuk  $n = 2000$





Gambar 4.12. GPH Tak Terkoreksi untuk  $n = 2000$

Tabel 4.6 pada probabilitas dan pattern yang berbeda pada kedua macam GPH menunjukkan kembali keterangan bahwa model *random shift* kembali menghasilkan nilai  $d$  yang kecil dengan harapan akan semakin jelas menunjukkan kerandoman dari pergerakan parameter  $d$  dan memperkuat kesimpulan di awal saat mensimulasikan *random shift* dengan sampel sebanyak 500. Gambar 4.11 dan Gambar 4.12, menunjukkan bahwa penambahan jumlah sampel  $n$  tidak memberikan efek signifikan terhadap hasil estimasi parameter  $d$  ketika diaggregasi, plot yang ditampilkan memang jelas tetap menunjukkan bahwa tidak adanya pola yang jelas (random) terhadap naiknya level agregasi untuk GPH tidak terkoreksi.

#### 4.2.3 Model Stop Break

Model yang digunakan dalam proses *STOP-BREAK* adalah :

$$x_t = \mu_t + \varepsilon_t, \mu_t = \mu_{t-1} + \frac{\varepsilon_{t-1}^2}{\gamma + \varepsilon_{t-1}^2} \varepsilon_{t-1}$$

dengan  $\varepsilon_t \sim N(0,1)$

Model *Stop Break* yang yang dibangkitkan ada 3 macam, yaitu:

1. *Stop Break 1* : ( $\gamma = 50$ )
2. *Stop Break 2* : ( $\gamma = 100$ )

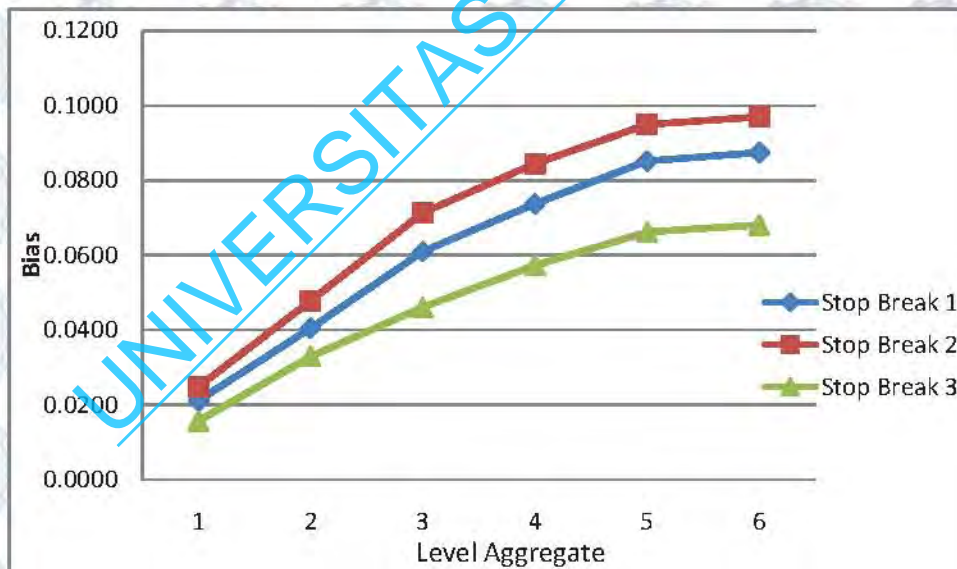
### 3. *Stop Break 3* : ( $\gamma = 180$ )

Pada model *Stop Break* berikut menggunakan *gamma* yang berbeda. Dalam simulasi ini akan diketahui pengaruh *gamma* terhadap estimator *long memory*.

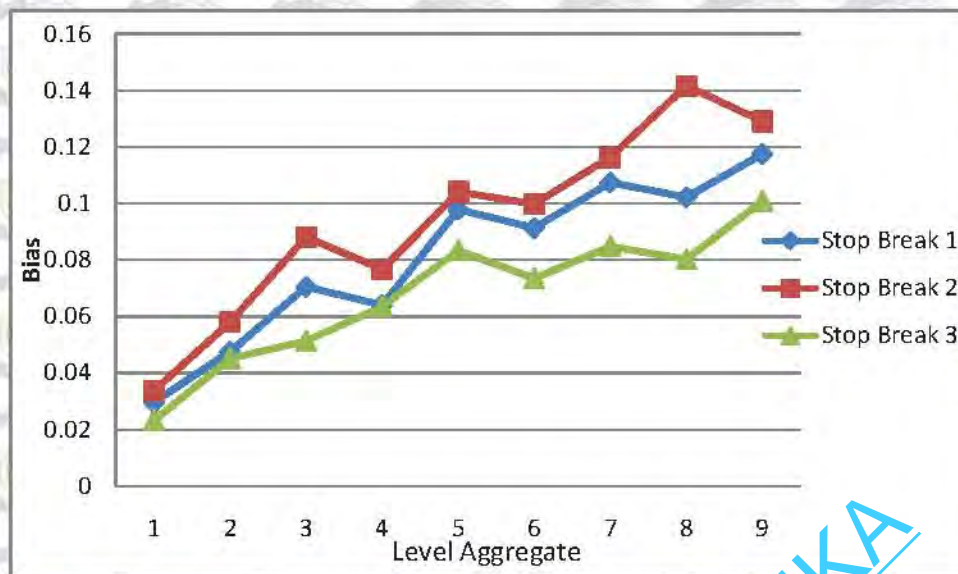
Tabel 4.7 Simulasi *Stock Aggregation* untuk Model *Stop Break*  $n = 500$

m	bandwith = 0.5					
	Gamma 50		Gamma 100		Gamma 180	
	GPH 1	GPH 2	GPH 1	GPH 2	GPH 1	GPH 2
1	0.5236	0.5236	0.3291	0.3291	0.1829	0.1829
2	0.5025	0.4940	0.3043	0.2953	0.1672	0.1597
3	0.4831	0.4761	0.2813	0.2711	0.1500	0.1378
4	0.4627	0.4532	0.2578	0.2410	0.1368	0.1316
5	0.4499	0.4596	0.2446	0.2524	0.1256	0.1194
6	0.4385	0.4258	0.2341	0.2250	0.1167	0.0996
7	0.4362	0.4323	0.2321	0.2293	0.1148	0.1094
8	NaN	0.4162	NaN	0.2128	NaN	0.0982
9	NaN	0.4214	NaN	0.1875	NaN	0.1027
10	NaN	0.4062	NaN	0.2000	NaN	0.0821

Plot dari hasil simulasi untuk lebih jelas menunjukkan bias yang terjadi disajikan pada Gambar 4.13 dan Gambar 4.14.



Gambar 4.13. GPH Terkoreksi untuk  $n = 500$



Gambar 4.14. GPH Tak Terkoreksi untuk  $n = 500$

Model *Stop Break* diperkenalkan oleh Engle dan Smith (1999), dengan pengujian menggunakan tiga parameter berbeda dimana terlihat dengan semakin besarnya penggunaan  $\gamma$  akan dihasilkan nilai  $d$  yang kecil. Tabel 4.7 pada *bandwidth* 0.5 menunjukkan keterangan bahwa model *stop break* menghasilkan nilai  $d$  yang kecil dan jelas menunjukkan kerandoman dari pergerakan parameter  $d$ , dimana nilai  $d$  akan cenderung monoton turun dengan naiknya level agregasi. Gambar 4.13 dan Gambar 4.14 menunjukkan bahwa penambahan jumlah level tidak memberikan efek signifikan terhadap hasil estimasi parameter  $d$  ketika diagregasi, plot yang ditampilkan memang jelas tetap menunjukkan bahwa tidak adanya pola yang jelas (random) terhadap naiknya level agregasi.

Selanjutnya akan disimulasikan dengan sampel yang lebih besar yaitu 2000. Berikut adalah hasil simulasi penggunaan *stock aggregation* pada bangkitan model *Stop Break* dengan sampel 2000.

Tabel 4.8 Simulasi *Stock Aggregation* untuk Model *Stop break*  $n = 2000$

m	Bandwidth = 0.5					
	Gamma 50		Gamma 100		Gamma 180	
	GPH 1	GPH 2	GPH 1	GPH 2	GPH 1	GPH 2
1	0.8099	0.8099	0.6298	0.6298	0.4506	0.4506
2	0.8029	0.7967	0.6201	0.6132	0.4388	0.4327
3	0.7977	0.7905	0.6090	0.5952	0.4231	0.4065
4	0.7930	0.7855	0.6034	0.6020	0.4139	0.4109
5	0.7892	0.7849	0.5954	0.5869	0.4040	0.3995

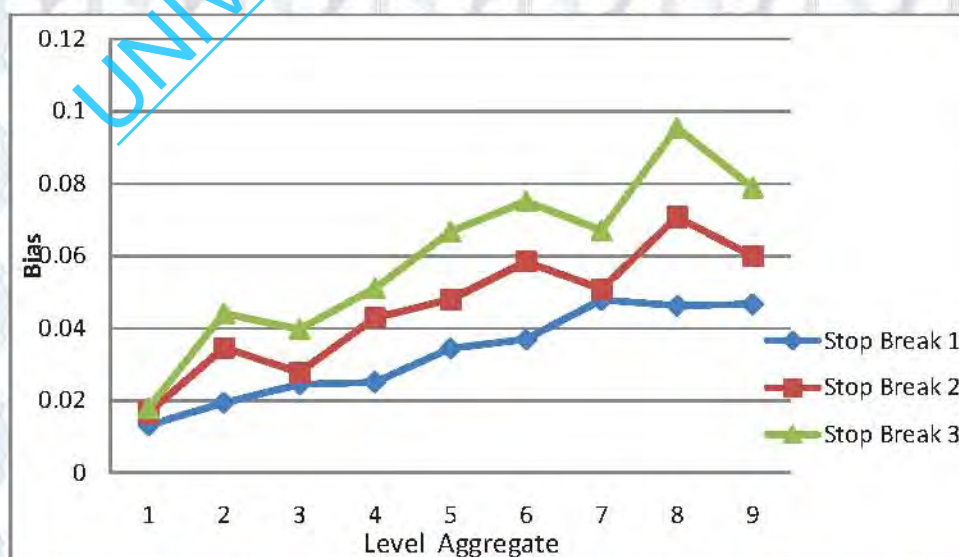
Tabel 4.8 Simulasi *Stock Aggregation* untuk Model *Stop break*  $n = 2000$   
(Lanjutan)

m	Bandwith = 0.5					
	Gamma 50		Gamma 100		Gamma 180	
	GPH 1	GPH 2	GPH 1	GPH 2	GPH 1	GPH 2
6	0.7843	0.7755	0.5878	0.5818	0.3933	0.3838
7	0.7798	0.7730	0.5799	0.5714	0.3833	0.3755
8	0.7746	0.7621	0.5743	0.5789	0.3763	0.3835
9	0.7707	0.7637	0.5676	0.5590	0.3677	0.3551
10	0.7677	0.7633	0.5637	0.5698	0.3630	0.3717

Plot dari hasil simulasi untuk lebih jelas menunjukkan bias yang terjadi disajikan pada Gambar 4.15 dan Gambar 4.16.



Gambar 4.15. GPH Terkoreksi untuk  $n = 2000$



Gambar 4.16. GPH Tak Terkoreksi untuk  $n = 2000$

Dengan bertambahnya sampel  $n$  terlihat kembali bawah pada Tabel 4.8 dengan *bandwith* 0.5 bahwa model stop break menghasilkan nilai  $d$  kecil dan random monoton turun yang sangat jelas menunjukkan kerandoman dari pergerakan parameter  $d$ . Penambahan jumlah level tidak memberikan efek signifikan juga terhadap hasil estimasi parameter  $d$  ketika diaggregasi, plot yang ditampilkan juga tetap menunjukkan bahwa tidak adanya pola yang jelas (random) terhadap naiknya level aggregasi GPH tidak terkoreksi.

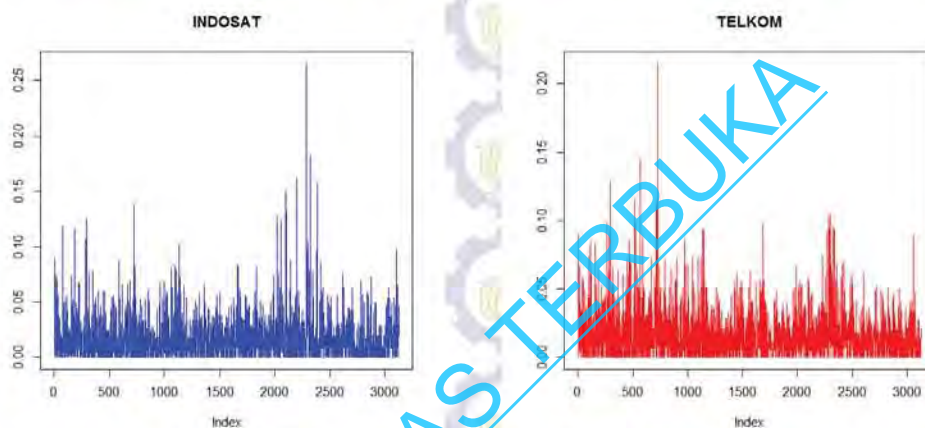
### 4.3 Hasil Dan Pembahasan Simulasi

Properti *stock aggregation* dalam mendeteksi fenomena *Long Memory* dimana bias yang dihasilkan dengan menggunakan aggregasi ini lebih besar. Baik menggunakan estimator GPH dengan *bandwith* 0,5. Trend estimasi nilai  $d$  yang dihasilkan dengan aggregasi adalah menurun, artinya semakin besar level aggregasi yang digunakan akan cenderung menghasilkan estimasi nilai  $d$  yang lebih kecil bahkan bisa saja hampir setengah dari parameter  $d$  awalnya dengan tanpa aggregasi. Hal ini tidak sesuai dengan gagasan awal bahwa untuk mendeteksi fenomena *Long Memory*, parameter  $d$  hasil aggregasi tidak berbeda dengan parameter  $d$  tanpa aggregasi. Namun perlu diakui bahwa meskipun parameter  $d$  yang dihasilkan berbeda, namun masih berada dalam range nilai  $d$  untuk *Long Memory*. Hal ini bisa saja terjadi karena dengan menggunakan metode *stock aggregation* yang mirip dengan *skip sampling* ini akan menghilangkan beberapa informasi data dan semakin besar level aggregasinya akan semakin besar pula informasi yang hilang. Pada kasus bangkitan *Markov Switching*, *Random Level Shift* dan *Stop Break*, pengaruh aggregasi ini menunjukkan bahwa tidak ditemukan trend, dimana terlihat pada ketiga model dengan menggunakan sampel 500 dan sampel 2000 tidak terlihat memberikan efek yang signifikan terhadap hasil estimasi parameter  $d$  ketika diaggregasi dengan GPH terkoreksi maupun GPH tidak terkoreksi.

### 4.4 Aplikasi ke Saham LQ45

Bagian ini dimaksudkan untuk memvalidasi hasil simulasi terhadap kasus riil data *return* saham LQ45 yang dijadikan bahan studi kasus dalam tesis ini adalah

saham Indosat dan Telkom. Kedua jenis indeks saham tersebut diambil datanya mulai awal Januari 2001 hingga akhir Desember 2011. Seperti biasa, yang dianalisis adalah absolut dari nilai *return* harga saham, bukan harga saham secara langsung sebab kenyataannya investor lebih tertarik mengetahui informasi pergerakan *return* saham daripada harga saham itu sendiri (Ding *et al.*(1993), Andersen *et al.*(2003), Sibbertsen (2006)). Berikut adalah sajian deskripsi series data absolute return saham dari masing-masing indeks saham.



**Gambar 4.17.** Deskripsi Series Data Return Saham Telkom dan Indosat

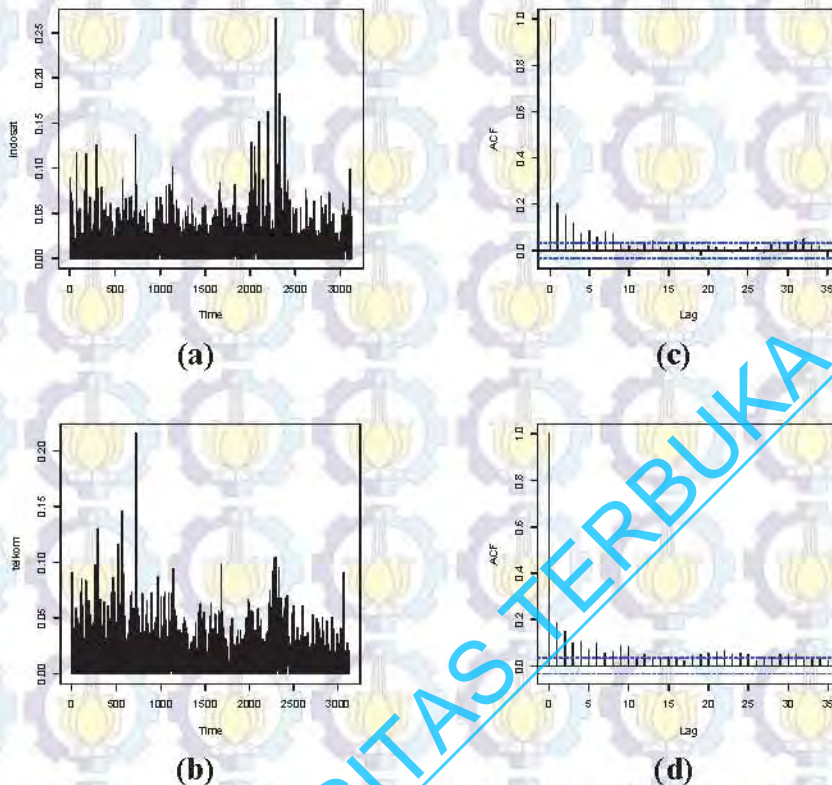
Pada deskripsi series saham yang ditunjukkan pada Gambar 4.17 *time series* plot yang berwarna biru menggambarkan series dari data absolut *return* dari kedua gambar tersebut sangat sulit diidentifikasi apakah series *return* mempunyai pola *nonlinier long memory* dimana series *return* saham Indosat yang berwarna biru, merah untuk series return Telkom. Kedua saham ini selanjutnya akan dimodelkan dengan beberapa pilihan model, namun sebelumnya akan diidentifikasi terlebih dahulu apakah series data tersebut tergolong fenomena *Long Memory* atau bukan.

#### 4.4.1 Identifikasi *Long Memory*

##### A. Plot ACF

Cara paling sederhana untuk mendeteksi fenomena *Long Memory* adalah secara visual melalui plot ACF. Jika plot ACF turun secara hiperbolik naik turun seperti gelombang, maka ada indikasi awal bahwa series tersebut masuk ke dalam

fenomena *Long Memory*. Hal serupa terjadi pada series saham Indosat dan Telkom, plot ACF dapat dilihat pada gambar 4.18.



**Gambar 4.18.** Time Series Plot data (a) Indosat, (b) Telkom, dan Plot ACF (c) Indosat, (d) Telkom

Pada Gambar tersebut terlihat bahwa plot ACF untuk series Indosat turun secara hiperbolik namun hanya pada beberapa lag saja, sedangkan plot ACF untuk series Telkom nampak jelas turun secara hiperboliknya. Hal ini memberikan dugaan awal bahwa kedua series tersebut tergolong *Long Memory*. Namun begitu, plot ACF hanyalah cara visual untuk deteksi, kenyataannya seringkali plot ACF tidak menampilkan pola yang sempurna untuk bisa benar-benar memutuskan apakah datanya long memory atau tidak.

Pengujian *stationarity* dilanjutkan untuk menguji stasioneritas dalam varians dengan menggunakan *box cox transformation*. Jika dalam pengujian menghasilkan initial value  $\lambda = 1$  atau *confidence interval* dari  $\lambda$  melewati nilai 1, maka series dapat dikatakan memiliki varians yang konstan. Berikut adalah hasil pengujian stasioneritas dalam varians pada kedua series tersebut.

Table 4.9 Uji Stationeritas dalam varians

Series	Initial Value ( $\lambda$ )	Lower CL	Upper CL
Telkom	0.50	-0.85	1.46
Indosat	1.00	0.01	1.93

Berdasarkan Tabel 4.9 dapat diketahui bahwa kedua series menghasilkan nilai  $\lambda = 1$  masuk dalam interval UCL dan LCL, maka hal ini jelas menunjukkan bahwa ketiga series tersebut juga telah stasioner dalam varians

### B. Identifikasi Data Dengan *Stock Aggregation*

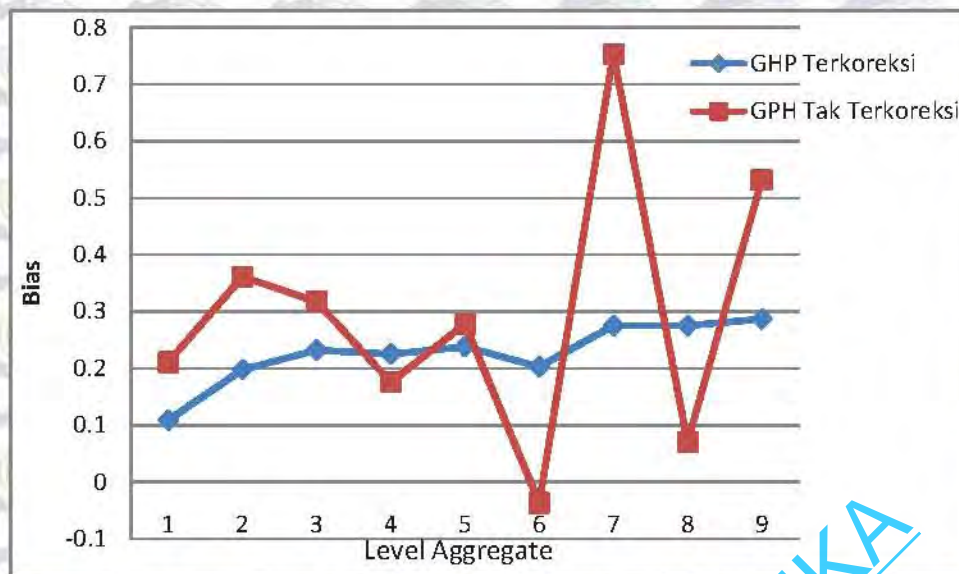
Sebagaimana telah disimpulkan dari hasil simulasi bahwa *temporal aggregation* dapat digunakan untuk mengidentifikasi data bersifat *Long Memory* atau tidak. Oleh karena itu kedua series tersebut akan diidentifikasi dengan menggunakan properti dari salah satu jenis agregasi yang sesuai. Pencatatan harga saham menyerupai sifat dari *skip sampling*, maka untuk identifikasi kedua series ini digunakan *stock aggregation* dapat dilihat pada tabel dibawah ini.

Tabel 4.10 Estimasi Nilai  $d$  dengan  $m$  Level Agregasi

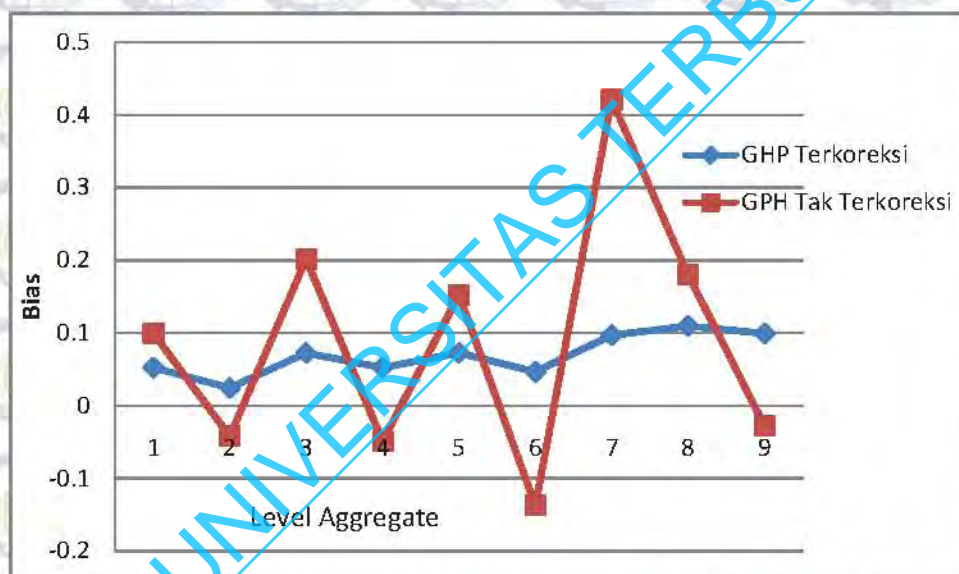
m	Indosat		Telkom	
	GPH Terkoreksi	GPH Tak Terkoreksi	GPH Terkoreksi	GPH Tak Terkoreksi
1	0.211983014	0.211983014	0.2949197	0.2949197
2	0.102858952	0.001012839	0.2423878	0.1952800
3	0.014360236	-0.149350304	0.2704356	0.3372464
4	-0.020065229	-0.105274667	0.2223215	0.0934484
5	-0.013138544	0.036458563	0.2426874	0.3437983
6	-0.026155397	-0.066251649	0.2221489	0.1427549
7	0.009181427	0.248732510	0.2483249	0.4318707
8	-0.063335406	-0.541313456	0.1978951	-0.1258277
9	-0.044013181	0.141971004	0.1851135	0.1146025
10	-0.074813144	-0.319116749	0.1954738	0.3226214

Model GPH tak terkoreksi tidak digunakan untuk identifikasi model, sebagaimana yang dihasilkan dari simulasi.





**Gambar 4.19** Plot Bias Estimasi Nilai  $d$  Indosat



**Gambar 4.20** Plot Bias Estimasi Nilai  $d$  Telkom

Berdasarkan hasil simulasi, fenomena *Long Memory* dapat dijelaskan dengan *Stock Aggregation* ditandai dengan pola nilai  $d$  yang monoton turun seiring dengan bertambahnya level agregasi. Kenyataannya hal tersebut tidak nampak terjadi pada kedua series saham. Tabel di atas jelas menunjukkan bahwa baik series Indosat ataupun Telkom, nilai estimasi  $d$  yang dihasilkan cenderung random seiring dengan bertambahnya level agregasi. Kondisi ini membuat dugaan awal yang semula kedua series ini tergolong *Long Memory* menjadi terbantahkan. Namun jika melihat nilai estimasi saat  $m = 1$  pada kedua series masih tergolong

dalam parameter *Long Memory*. Sehingga dugaan malah beralih pada kondisi *Spurious Long Memory*.

Telah dijelaskan diatas bahwa kedua series data telah stasioner dalam varians, tetapi jika dilihat dari plot ACF terlihat hasilnya banyak yang keluar dari batas interval, dimana plot ACF yang dihasilkan berbentuk hiperbolik naik turun, seperti ciri sifat *long memory*. Selanjutnya akan dilakukan identifikasi berdasarkan orde-orde model terbaik berdasarkan nilai AIC terkecil untuk memodelkan ARFIMA (*long memory*) dan model *Makov Switching* sebagai model *spurious long memory* sehingga didapat model mana yang terbaik.

Tabel 4.11 Estimasi parameter model ARFIMA Series data Indosat

No.	Model ARFIMA	Parameter	Statistik t	P-Value	AIC
1.	(3,d,3)	$\theta_0 = 0.01724$	11.495	0	8158.22
		$\phi_1 = -0.61924$	-2.837	0.005	
		$\phi_2 = -0.64956$	-5.923	0	
		$\phi_3 = -0.66409$	-3.808	0	
		$\theta_1 = -0.59289$	-2.673	0.008	
		$\theta_2 = -0.6402$	-6.205	0	
		$\theta_3 = -0.68169$	-3.78	0	
		$d = 0.18888$	6.975	0	

Tabel diatas telah didapatkan ordo ARFIMA berdasarkan nilai terkecil, selanjutnya akan dilakukan cek asumsi residual *white noise* dan berdistribusi normal. *White Noise Test* adalah uji untuk mengetahui apakah residual sudah *white noise*. Statistik uji yang digunakan adalah *Ljung-Box Test* dengan  $H_0$  adalah residual *white noise*. Dari hasil pengujian *white noise*, diketahui bahwa residual dari model tidak *white noise* karena plot *p-value* hingga lag-12 berada di bawah nilai  $\alpha = 0.05$  yang berarti tolak  $H_0$ .

Tahapan evaluasi yang lain adalah *Normality Test* yang berguna untuk mengetahui apakah residual sudah berdistribusi normal. Statistik uji yang akan digunakan adalah *Kolmogorov-Smirnov Test* dengan  $H_0$  adalah residual berdistribusi normal. Uji normalitas untuk residual dari model ARFIMA untuk series data indosat, hasil uji menggunakan *Kolmogorof-Smirnov Test* nilai KS = 0.4868 dengan *p-value* =  $2.2e-16$  kurang dari  $\alpha = 0,05$ , Oleh karena *p-value* <  $\alpha$ ,

maka tolak  $H_0$  dan disimpulkan bahwa residual yang dihasilkan kedua model tidak berdistribusi normal.

Selanjutnya akan dimodelkan untuk series data Telkom, dapat dilihat pada table 4.12.

Tabel 4.12 Estimasi parameter model ARFIMA Series data Telkom

No.	Model ARFIMA	Parameter	Statistik t	P-Value	AIC
1.	(1,d,1)	$\theta_0 = 0.01608$	12.559	0	8406.35
		$\phi_1 = 0.24593$	2.049	0.041	
		$\theta_1 = 0.30794$	2.796	0.005	
		$d = 0.20219$	5.881	0	
2.	(2,d,2)	$\theta_0 = 0.01617$	14.307	0	8402.73
		$\phi_1 = 0.61813$	3.666	0	
		$\phi_2 = -0.37651$	-2.849	0.004	
		$\theta_1 = 0.64879$	3.832	0	
		$\theta_2 = -0.39411$	-3.105	0.002	
3.	(5,d,5)	$\theta_0 = 0.01615$	13.575	0	8400.23
		$\phi_1 = -1.23858$	-4.796	0	
		$\phi_2 = -1.23331$	-5.966	0	
		$\phi_3 = -0.62215$	-2.484	0.013	
		$\phi_4 = -0.40282$	-4.152	0	
		$\phi_5 = -0.37216$	-2.907	0.004	
		$\theta_1 = -1.19504$	-4.371	0	
		$\theta_2 = -1.18646$	-5.73	0	
		$\theta_3 = -0.55768$	-2.125	0.034	
		$\theta_4 = -0.3882$	-4.435	0	
		$\theta_5 = -0.34763$	-2.295	0.022	
		$d = 0.18712$	6.75	0	

Setelah diperoleh ordo ARFIMA, selanjutnya akan dilakukan uji residual *white noise* dan berdistribusi normal untuk series data Telkom. Dari hasil masing-masing model ARFIMA didapatkan *Ljung-Box Test* untuk uji asumsi residual *white noise* untuk model ARFIMA (1,d,1), (2,d,2) dan (5,d,5) sudah *white noise* karena *p-value* hingga lag-12 berada di atas nilai  $\alpha = 0.05$  yang berarti terima  $H_0$ , tetapi pada model ARFIMA (2,d,2) ada yang tidak *white noise* pada lag-6,

sehingga tetap diputuskan bahwa tolak  $H_0$ , yang artinya tidak *white noise* pada model tersebut.

Selanjutnya, uji normalitas untuk residual dari model ARFIMA untuk series data telkom, hasil uji menggunakan *Kolmogorof-Smirnov Test* model ARFIMA pada tabel 4.13.

Tabel 4.13 *Normality Test*

Model Arfima	N	KS	P-Value
(1,d,1)	3128	0.4874	2.2e-16
(2,d,2)	3127	0.4872	2.2e-16
(5,d,5)	3127	0.4872	2.2e-16

Tabel 4.13 adalah hasil uji normalitas diperoleh nilai KS dan p-value, sehingga dapat disimpulkan untuk ketiga model Arfima tersebut menunjukkan bahwa residual tidak berdistribusi normal, dikarenakan nilai *P-value* kurang dari  $\alpha = 0,05$ . Oleh karena  $p\text{-value} < \alpha$ , maka tolak  $H_0$  hanya model ARFIMA (5,d,5) yang nilai *p-valuenya* lebih dari  $\alpha = 0,05$ .

Table 4.11 dan 4.12 diatas dapat dijelaskan bawah, pada series data Indosat, diduga kemungkinan modelnya ARFIMA (3,d,3) dengan AIC = 8158,22. Sedangkan untuk series data Telkom, menunjukkan kemungkinan modelnya adalah ARFIMA (1,d,1) dengan AIC = 8406,35, ARFIMA (2,d,2) dengan AIC = 8402,73, serta ARFIMA (5,d,5) dengan AIC = 8400,23. Sehingga jika dilihat dari estimasi parameter pada table diatas untuk semua model yang signifikan, model dugaan ARFIMA untuk series data Indosat adalah ARFIMA (3, 0.18888, 3) dan series data Telkom adalah ARFIMA (5, 0.18712, 5).

## 4.5 Pemodelan Markov Switching

Pada model *Markov Switching* dengan 2 *regime*, dilakukan *fitting* model secara langsung yang menghasilkan AIC terkecil dan menghasilkan parameter yang signifikan pula, dengan demikian akan dilakukan pemilihan ordo berdasarkan parameter yang sesuai terlebih dahulu.

### 4.5.1 Series Data Indosat

Seperti halnya dalam memodelkan data dengan model ARFIMA, penentuan order dalam pemodelan *Markov Switching* pada penelitian ini dilakukan dengan

memilih model *Markov Switching* dengan AIC terendah, dan memastikan bahwa model memiliki parameter yang signifikan, serta residual yang *white noise* dan berdistribusi Normal  $(0, \sigma^2)$ . Dengan membentuk beberapa kombinasi order, model *Markov Switching* dengan AIC terkecil yang diperoleh adalah model *Markov Switching* AR(1) dan MA (1). Selain memiliki AIC terendah, model tersebut menghasilkan parameter yang seluruhnya signifikan serta residual yang berdistribusi Normal  $(0, \sigma^2)$ .

Tahapan pemodelan Markov-Switching AR (1) dan MA (1) diawali dengan pembentukan matriks transisi dari series data indosat. Dalam pemodelan ini digunakan dua *regime*, sehingga pembentukan matriks didasarkan oleh probabilitas data di setiap *regime* berdasarkan konsep random, series data indosat diperoleh matriks transisi sebagai berikut

Matrik transisinya dari model AR(1) dan MA(1) sebagai berikut :

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

Matrik Markov Transition Probabilities AR (1) :

$$p = \begin{bmatrix} 0.96096 & 0.039038 \\ 0.74292 & 0.25708 \end{bmatrix}$$

Berdasarkan matrik transisi model AR(1) diketahui bahwa probabilitas transisi observasi dari *regime* 1 ( $P_{11}$ ) sebesar 0.96096, artinya apabila awalnya suatu observasi berada di *regime* 1, maka observasi selanjutnya akan tetap di *regime* 1 dengan probabilitas sebesar 0.96096. begitu pula dengan nilai probabilitas yang berada di *regime* 1 ( $P_{21}$ ) sebesar 0.74292, dari *regime* 1 berpindah ke *regime* 2 ( $P_{12}$ ) sebesar 0.039038 dan tetap berada di *regime* 2 ( $P_{22}$ ) sebesar 0.25708. Selanjutnya diperoleh matrik transisi probabilitas untuk model MA (1) sebagai berikut.

Matrik Markov Transition Probabilities MA (1):

$$p = \begin{bmatrix} 0.99349 & 0.0065125 \\ 1.0000 & 1.4525e - 011 \end{bmatrix}$$

Begitu juga untuk matrik transisi model MA (1) diketahui bahwa probabilitas transisi periode residual dari *regime* 1 ( $P_{11}$ ) sebesar 0.99349, artinya apabila awalnya suatu periode residual berada di *regime* 1, maka periode atau residual selanjutnya akan tetap di *regime* 1 dengan probabilitas sebesar 0.99349. begitu

pula dengan periode residual yang berada di *regime* 1 ( $P_{21}$ ) sebesar 1.0000, dari *regime* 1 berpindah ke *regime* 2 ( $P_{12}$ ) sebesar 0.0065125 dan tetap berada di *regime* 2 ( $P_{22}$ ) sebesar 1.4525e-011.

Setelah diperoleh matriks transisi, selanjutnya dilakukan estimasi parameter model *Markov Switching* AR (1) dan MA (1) untuk series data indosat. Hasil estimasi parameter ditampilkan pada tabel berikut ini.

Tabel 4.14 Estimasi parameter model *Markov Switching* Series data Indosat

No.	Model Markov Switching	Parameter	Statistika t	P-value	AIC
1.	AR(1)	$\mu_0 = 0.01578$	42.66	0	8194.54
	Regime 1	$\phi_1 = 0.10024$	2.592	0.01	
	Regime 2	$\phi_1 = 1.07575$	3.148	0.002	
2.	MA(1)	$\mu_0 = 0.01611$	41.312	0	8200.05
	Regime 1	$\theta_1 = -0.13604$	-6.087	0	
	Regime 2	$\theta_1 = -0.84443$	-37.884	0	

Dari ordo ARIMA diatas, selanjutnya akan dilakukan uji residual white noise dan berdistribusi normal untuk series data Indosat. Model ARIMA didapatkan hasil dengan *Ljung-Box Test* untuk uji asumsi residual *white noise* untuk model ARIMA(1,0,0) dan (0,0,1) tidak *white noise* karena *p-value* hingga lag-12 berada di atas nilai  $\alpha = 0.05$  yang berarti terima  $H_0$ , dikarenakan model ARIMA diatas tidak *white noise*, selanjutnya akan dilakukan uji normalitas untuk residual dari model ARFIMA untuk series data telkom, hasil uji menggunakan *Kolmogorof-Smirnov Test* model ARFIMA pada tabel 4.15.

Tabel 4.15. *Normality Test*

Model Markov	N	KS	P-Value
AR(1)	3128	0.4901	2.2e-16
MA(1)	3129	0.4895	2.2e-16

Hasil uji normalitas diperoleh nilai KS dan *p-value*, sehingga dari tabel diatas dapat disimpulkan untuk kedua model ARMA tersebut menunjukkan bahwa residual tidak berdistribusi normal, dikarenakan nilai *P-value* kurang dari  $\alpha = 0,05$ , Oleh karena  $p\text{-value} < \alpha$ , maka tolak  $H_0$ .

Model *Markov Switching* dengan proses  $AR(p)$  sesuai dengan persamaan yang dijelaskan oleh Zivot dan Wang (2006).

$$\phi_{p,St}(B)Y_t = a_t \quad \text{untuk } t = 1, 2, \dots, t$$

$$y_t = \mu_{St} + \phi_{1,St}y_{t-1} + \phi_{2,St}y_{t-2} + \dots + \phi_{p,St}y_{t-p} + a_t$$

$$y_t = \mu_{st} + X_t \phi_{st} + a_t$$

dimana  $X_t = (y_{t-1}, y_{t-2}, \dots, y_{t-p})$  dan  $\phi_{st}$  adalah vektor koefisien AR berukuran  $p \times 1$ .

Berdasarkan tabel 4.14 diperoleh model *Markov Switching* pada *regime 1* dan *regime 2* untuk AR (1) :

$$y_t = \mu_{st} + \phi_{1,st}y_{t-1} + \phi_{2,st}y_{t-2} + \dots + \phi_{p,st}y_{t-p} + a_t$$

$$y_t = \mu_1 + \phi_{1,1}y_{t-1} + a_t$$

$$y_t = 0.01578 + 0.10024y_{t-1} + a_t \quad \dots \dots \text{Regime 1}$$

$$y_t = 0.01578 + 0.1.07575y_{t-1} + a_t \quad \dots \dots \text{Regime 2}$$

Sehingga dari persamaan diatas diperoleh :

$$y_t = \begin{cases} \mu_0 + \phi_1 y_{t-1} + a_t \\ \mu_0 + \phi_2 y_{t-1} + a_t \end{cases}$$

$$y_t = \begin{cases} 0.01578 + 0.10024y_{t-1} + a_t \\ 0.01578 + 0.1.07575 y_{t-1} + a_t \end{cases}$$

Dari model AR (1) yang terbentuk dapat dilihat bahwa *regime 1* dipengaruhi oleh satu observasi sebelumnya sebesar 0.13604, begitu pula pada *regime 2* dipengaruhi oleh satu observasi sebelumnya sebesar 0.01578, artinya pergerakan data dari *regime 1* mengikuti model AR (1) dengan data bergerak secara stasioner berada di *regime 1* dan *regime 2*.

Berdasarkan tabel 4.14 diperoleh model *Markov Switching* untuk *regime 1* dan *regime 2* untuk MA (1) :

$$y_t = \mu_{st} - \theta_{1,st}a_{t-1} - \theta_{2,st}a_{t-2} - \dots - \theta_{p,st}a_{t-p}$$

$$y_t = \mu_1 - \theta_{1,1}a_{t-1}$$

$$y_t = 0.01611 - 0.13604a_{t-1} \quad \dots \dots \text{Regime 1}$$

$$y_t = 0.01611 - 0.84443a_{t-1} \quad \dots \dots \text{Regime 2}$$

Sehingga dari persamaan diatas diperoleh :

$$y_t = \begin{cases} \mu_0 + \theta_1 y_{t-1} + a_t \\ \mu_0 + \theta_2 y_{t-1} + a_t \end{cases}$$

$$y_t = \begin{cases} 0.01611 - 0.13604y_{t-1} + a_t \\ 0.01611 - 0.84443 y_{t-1} + a_t \end{cases}$$

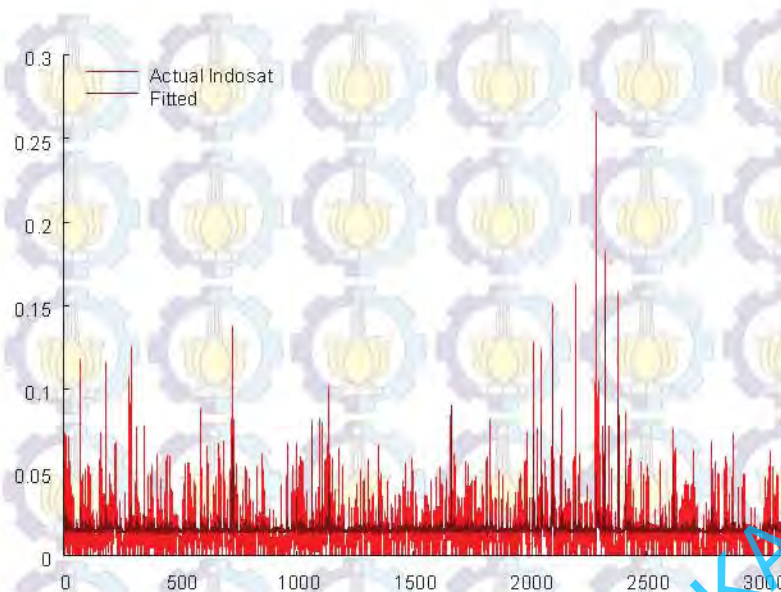
Model *markov switching* yang terbentuk dari model MA (1) bahwa pada *regime* 1 dipengaruhi oleh satu periode residual sebelumnya sebesar 0.13604, begitu pula pada *regime* 2 juga dipengaruhi satu periode residual sebelumnya sebesar 0.84443, artinya pergerakan periode residual dari *regime* 1 mengikuti model MA (1) dimana periode residual bergerak secara stasioner berada di *regime* 1 dan *regime* 2.

Selanjutnya setelah *matrix transisi* dan model *markov switching* AR (1) dan MA (1) dari masing-masing *regime* diperoleh, akan dilanjutkan dengan membandingkan plot nilai *absolute return* dengan nilai fit, seperti terlihat pada gambar 4.21 dan gambar 4.22.



Gambar 2.21 Perbandingan nilai fit vs nilai actual Indosat pada model AR (1)





Gambar 4.22 Perbandingan nilai fit vs nilai actual Indosat pada model MA (1)

Jika dilihat dari gambar 4.21 dan gambar 4.22, terlihat bahwa nilai fit dan nilai absolute *return* relatif hampir sama dimana tingkat fluktuasi juga hampir sama, kemungkinan dari plot diatas dikarenakan data return saham mengandung *outlier* sehingga plot yang terlihat tidak relatif sama sepenuhnya, baik dari nilai *fitted* dan nilai *actual* indosat, tetapi pola dari gambar plot tersebut relatif sama.

#### 4.5.2 Series data Telkom

Pada series data Telkom, tahapan yang dilakukan juga sama seperti tahapan pada series data indosat, untuk series data telkom penentuan order dalam pemodelan *Markov Switching* dilakukan dengan memilih model *Markov Switching* dengan AIC terendah, dan memastikan bahwa model memiliki parameter yang signifikan, serta residual yang *white noise* dan berdistribusi Normal  $(0, \sigma^2)$ . Dengan membentuk beberapa kombinasi order, model *Markov Switching* dengan AIC terkecil yang diperoleh adalah model *Markov Switching* AR(1) dan MA(1). Selain memiliki AIC terendah, model tersebut menghasilkan parameter yang seluruhnya signifikan serta residual yang berdistribusi Normal  $(0, \sigma^2)$ .

Tahapan pemodelan *Markov Switching* AR(1) dan MA(1) diawali dengan pembentukan matriks transisi dari series data telkom. Dalam pemodelan ini digunakan dua *regime*, sehingga pembentukan matriks didasarkan oleh

probabilitas data di setiap *regime* berdasarkan konsep random, series data indosat diperoleh matriks transisi.

Selanjutnya didapat matrik transisinya dari model AR (1) dan model MA (1) sebagai berikut :

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

Matrik Markov Transition Probabilities AR (1) :

$$p = \begin{bmatrix} 0.10782 & 0.89218 \\ 0.044902 & 0.95510 \end{bmatrix}$$

Dari matrik transisi model AR (1) diketahui bahwa probabilitas transisi observasi dari *regime* 1 ( $P_{11}$ ) sebesar 0.10782, artinya apabila suatu observasi berada di *regime* 1, maka observasi selanjutnya akan tetap di *regime* 1 dengan probabilitas sebesar 0.10782. begitu pula dengan nilai probabilitas yang berada di *regime* 1 ( $P_{21}$ ) sebesar 0.044902, dari *regime* 1 berpindah ke *regime* 2 ( $P_{12}$ ) sebesar 0.89218 dan tetap berada di *regime* 2 ( $P_{22}$ ) sebesar 0.95510. Selanjutnya diperoleh matrik transisi probabilitas untuk model MA(1) sebagai berikut.

Matrik Markov Transition Probabilities MA (1):

$$p = \begin{bmatrix} 0.99075 & 0.0092519 \\ 0.68634 & 0.31366 \end{bmatrix}$$

Matrik transisi model MA(1) diketahui bahwa probabilitas transisi observasi dari *regime* 1 ( $P_{11}$ ) sebesar 0.99075, artinya apabila suatu periode residual berada di *regime* 1, maka observasi selanjutnya akan tetap di *regime* 1 dengan probabilitas sebesar 0.99075, begitu pula dengan nilai probabilitas yang berada tetap berada di *regime* 1 ( $P_{21}$ ) sebesar 0.068634, dari *regime* 1 berpindah ke *regime* 2 ( $P_{12}$ ) sebesar 0.0092519 dan tetap berada di *regime* 2 ( $P_{22}$ ) sebesar 0.31366.

Selanjutnya setelah diperoleh matriks transisi, dilakukan estimasi parameter model *Markov Switching* AR(1) dan MA(1) untuk series data indosat. Hasil estimasi paramater ditampilkan pada tabel berikut ini

Tabel 4.16 Estimasi parameter model *Markov Switching* Series data Telkom

No.	Model Markov Switching	Parameter	Statistika t	P-value	AIC
1	AR (1)	$\mu_0 = 0.01552$	40.838	0	8426.57

	Regime 1	$\phi_1 = 1.32619$	5.826	0	
	Regime 2	$\phi_1 = -1.23108$	-5.596	0.002	
2.	MA (2)	$\mu_0 = 0.01583$	47.978	0	8436.77
	Regime 1	$\theta_1 = -0.12431$	-6.651	0	
	Regime 2	$\theta_1 = -0.84802$	-39.963	0	

Dari ordo ARIMA diatas, selanjutnya akan dilakukan uji residual *white noise* dan berdistribusi normal untuk series data telkom Model ARIMA didapatkan hasil dengan *Ljung-Box Test* untuk uji asumsi residual *white noise* untuk model AR (1) dan MA (2) tidak *white noise* karena *p-value* hingga lag-12 berada di atas nilai  $\alpha = 0.05$  yang berarti tolak  $H_0$ , dikarenakan model AR (1) dan MA (1) tidak *white noise*, selanjutnya akan dilakukan uji normalitas untuk residual dari model ARFIMA untuk series data telkom, hasil uji menggunakan *Kolmogorof-Smirnov Test* model ARFIMA pada tabel 4.17.

Tabel 4.17. Normality Test

Model Markov	N	KS	P-Value
AR (1)	3128	0.4899	2.2e-16
MA (1)	3129	0.4897	2.2e-16

Hasil uji normalitas diperoleh nilai KS dan *p-value*, sehingga dari tabel diatas dapat disimpulkan untuk kedua model AR(1) dan MA(1) tersebut menunjukkan bahwa residual tidak berdistribusi normal, dikarenakan nilai *P-value* kurang dari  $\alpha = 0,05$ , Oleh karena  $p\text{-value} < \alpha$ , maka tolak  $H_0$ .

Berdasarkan tabel 4.16 diatas diperoleh model *Markov Switching* pada *regime 1* dan *regime 2* untuk AR (1) :

$$y_t = \mu_{st} + \phi_{1,st}y_{t-1} + \phi_{2,st}y_{t-2} + \dots + \phi_{p,st}y_{t-p} + a_t$$

$$y_t = \mu_1 + \phi_{1,1}y_{t-1} + a_t$$

$$y_t = 0.01552 + 1.32619y_{t-1} + a_t \quad \dots\dots \text{Regime 1}$$

$$y_t = 0.01552 - 1.23108y_{t-1} + a_t \quad \dots\dots \text{Regime 2}$$

Sehingga dari persamaan diatas diperoleh :

$$y_t = \begin{cases} \mu_0 + \phi_1 y_{t-1} + a_t \\ \mu_0 + \phi_2 y_{t-1} + a_t \end{cases}$$

$$y_t = \begin{cases} 0.01552 + 1.32619y_{t-1} + a_t \\ 0.01552 - 1.23108y_{t-1} + a_t \end{cases}$$

Dari model AR (1) yang terbentuk *regime* 1 dipengaruhi oleh satu observasi sebelumnya sebesar 01.32619, begitu pula pada *regime* 2 dipengaruhi oleh satu obeservasi sebelumnya sebesar -1.23108. artinya pergerakan data dari *regime* 1 mengikuti model AR(1) dengan data bergerak secara stasioner berada di *regime* 1 dan *regime* 2.

Selanjutnya berdasarkan (4.16) diperoleh model *Markov Switching* untuk *regime* 1 dan *regime* 2 untuk MA (1) :

$$y_t = \mu_{st} - \theta_{1,st}a_{t-1} - \theta_{2,st}a_{t-2} - \dots - \theta_{p,st}a_{t-p}$$

$$y_t = \mu_1 - \theta_{1,1}a_{t-1}$$

$$y_t = 0.01583 - 0.12431a_{t-1} \quad \dots\dots \text{Regime 1}$$

$$y_t = 0.01583 - 0.84802a_{t-1} \quad \dots\dots \text{Regime 2}$$

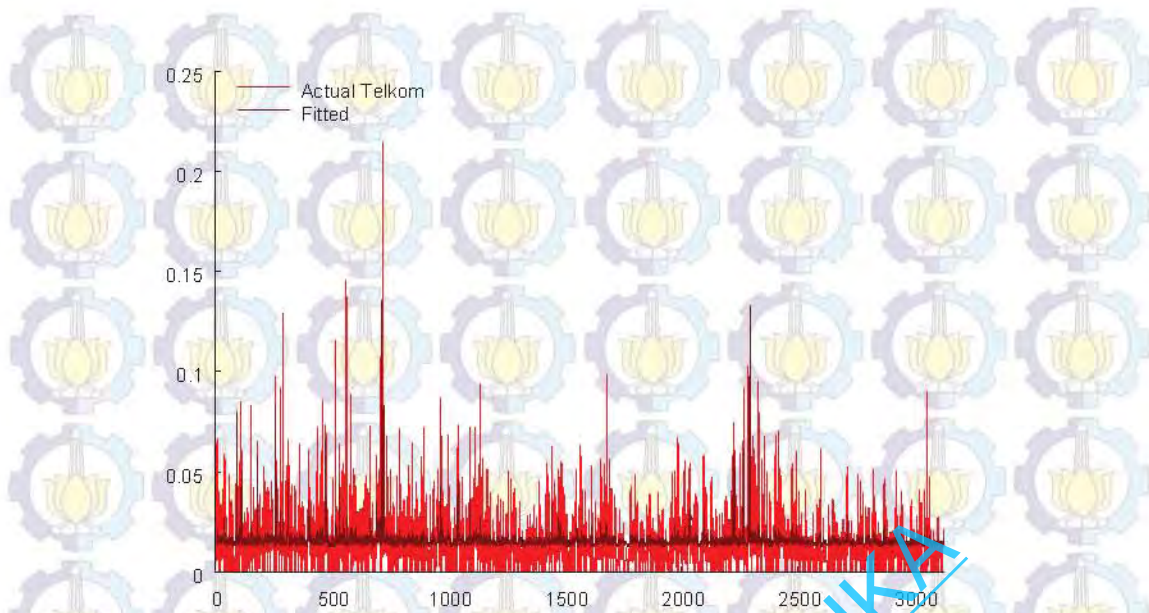
Sehingga dari persamaan diatas diperoleh :

$$y_t = \begin{cases} \mu_0 + \theta_1 y_{t-1} + a_t \\ \mu_0 + \theta_2 y_{t-1} + a_t \end{cases}$$

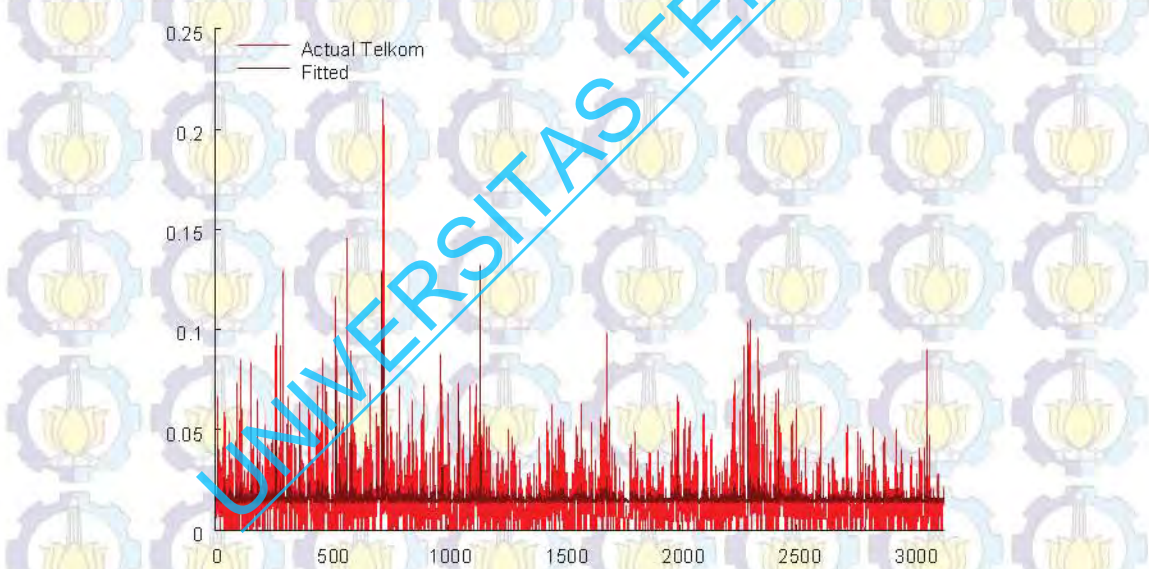
$$y_t = \begin{cases} 0.01583 - 0.12431 y_{t-1} + a_t \\ 0.01583 - 0.84802 y_{t-1} + a_t \end{cases}$$

Dari model *markov switching* yang terbentuk dari model MA(1) bahwa pada *regime* 1 dipengaruhi oleh satu periode residual sebelumnya sebesar 0.12431, begitu pula pada *regime* 2 juga dipengaruhi satu periode residual sebelumnya sebesar 0.84802, artinya pergerakan periode residual dari *regime* 1 mengikuti model MA (1) dengan periode residual bergerak secara stasioner berada di *regime* 1 dan *regime* 2.

Selanjutnya setelah *matrix transisi* dan model *markov switching* AR (1) dan MA (1) dari masing-masing *regime* diperoleh, akan dilanjutkan dengan membandingkan plot nilai *absolute return* dengan nilai fit, seperti terlihat pada gambar 2.23 dan gambar 2.24



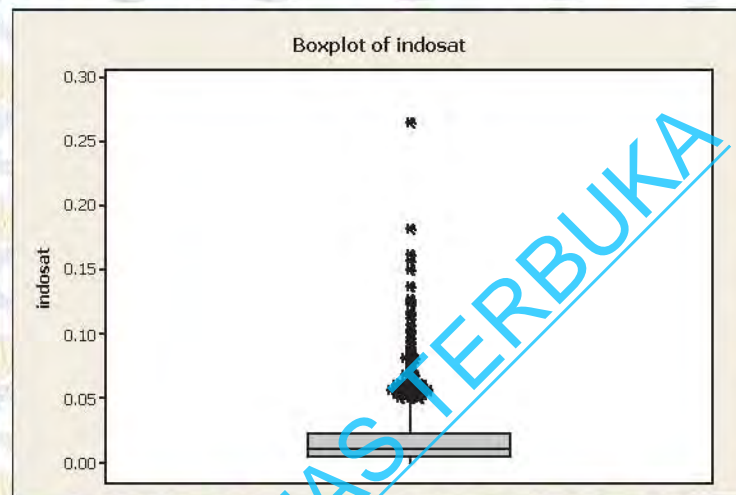
Gambar 2.23 Perbandingan nilai fit vs nilai actual Telkom pada model AR (1)



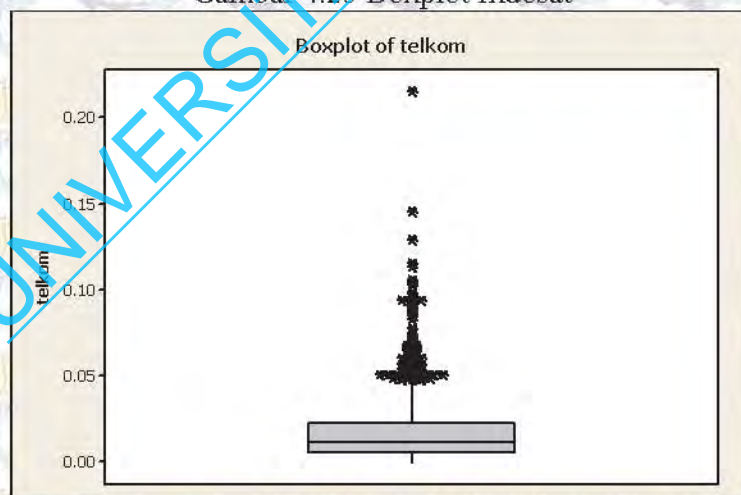
Gambar 2.24 Perbandingan nilai fit vs nilai actual Telkom pada model MA (1)

Pada 4.23 dan gambar 4.24, juga terlihat bahwa nilai fit dan nilai absolute return relatif hampir sama dimana tingkat fluktuasi juga hampir sama, yang mana kemungkinan dari plot diatas dikarenakan data *return* saham mengandung *outlier* sehingga plot yang terlihat tidak relatif sama sepenuhnya, baik dari nilai *fitted* dan nilai *actual* telkom, tetapi pola dari gambar plot tersebut relatif sama.

Dari diagnosa test apakah model ARFIMA dan *Markov Switching white noise* dan berdistribusi normal residuals, ternyata diperoleh hasil bahwa kedua model baik ARFIMA dan *Markov Switching* tidak *white noise* dan tidak berdistribusi normal, jika dilihat hasil dari kedua uji baik uji *white noise* dan uji normalitas dapat disimpulkan tidak sesuai, maka akan dilakukan cek apakah series data indosat maupun telkom tersebut outlier atau tidak dengan metode *boxplot* seperti pada gambar 4.25 dan gambar 4.26.



Gambar 4.25 Boxplot Indosat



Gambar 4.26 Boxplot Telkom

Dilihat dari *boxplot* untuk series data Indosat maupun series data Telkom dapat disimpulkan bahwa data kedua series adalah mengandung *outlier*, *Boxplot* dapat memberikan gambaran yang lebih spesifik tentang distribusi data, terlihat jelas disana series data Indosat maupun Telkom banyak yang outlier dan tidak berada

pada median dan dapat dipastikan series data Indosat adalah *spurious long memory* dan series data Telkom juga *spurious long memory*.

Hasil evaluasi akhir tentang ARFIMA dengan *Markov Switching* menunjukkan hasil simulasi terlihat bahwa sifat stock aggregate memiliki pola yang random baik untuk GPH terkoreksi maupun GPH tak terkoreksi. Sedangkan pada aplikasi model berdasarkan nilai AIC terkecil dan parameter yang signifikan, ARFIMA lebih baik dibandingkan *Markov Switching*, ternyata data dari kedua *return* saham LQ45 datanya mengandung *outlier* sehingga dengan hasil simulasi yang menunjukkan bahwa untuk data *spurious long memory*, maka hasil estimator GPH untuk data teragregasi tidak menunjukkan pola tertentu, tetapi jika dilihat pemodelannya dimana kedua series data menunjukkan mengandung *outlier* dan sudah dapat dipastikan bahwa memang *spurious long memory*.

## BAB V KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan analisis dan pembahasan yang telah dilakukan, berikut beberapa kesimpulan yang diperoleh adalah :

1. Pengidentifikasian sifat *Long Memory* dalam suatu series data dapat dilakukan dengan agregasi baik *flow aggregation* maupun *stock aggregation*. Dimana pada kasus ini hanya menggunakan stock aggregation. Berdasarkan hasil simulasi, stok agregasi ini menghasilkan perilaku yang sama dalam parameteranya untuk *Spurious Long Memory*, yaitu random, tidak memiliki trend turun atau naik jika seriesnya diagregasi.
2. Pemodelan dari absolut *return* saham dari kedua series terpilih yaitu Indosat dan Telkom, didapatkan bahwa model ARFIMA lebih baik dibandingkan model *Markov Switching*. Hasil aplikasi saham menunjukkan nilai estimasi GPH untuk data teragregasi memiliki pola yang random, dilihat dari nilai AIC terkecil berdasarkan kedua model, model ARFIMA memiliki nilai AIC terkecil, sehingga GPH standart tidak bisa digunakan untuk mendeteksi *spurious long memory*, dimana *return* saham dari kedua series adalah mengandung *outlier*.

### 5.2 Saran

Saran yang direkomendasikan untuk penelitian selanjutnya yang serupa adalah agar menggunakan properti pengoreksi bias terkoreksi dan bias tidak terkoreksi dari estimator GPH serta menggunakan contoh model *spurious long memory* lebih banyak lagi dengan sampel dan series data yang lebih banyak pula dari data yang mengandung *outlier*.





*Halaman Ini Sengaja Dikосongkan*

## DAFTAR PUSTAKA

- Aloy, M, Boutahar. M, Gente. K, Feissoll. K.P. (2011). *Purchasing Power Parity And The Long Memory Properties Of Real Exchange Rates: Does One Size fit All?*.
- Andersen, T., T. Bollerslev, F.X. Diebold & H. Ebens. (2001). *The Distribution Of Realized Stock Return Volatility. Journal Of Financial Economics 61, 43-76.*
- Anderson, T., T. Bollerslev, F.X. Diebold & P. Labys. (2003). *Modeling And Forecasting Realized Volatility. Econometrica 71, 579-626.*
- Baillie. R. T and Kapetanios. G. (2006). *Nonlinier Model With Strongly Dependent Processes And Applications To Foward Premia And Real Exchange Rates.*
- Beran, Jan. (1994). *Statistics for Long-Memory Processes (Chapman & Hall, London).*
- Brewer, K.R.W. (1973). *Some Consequences Of Temporal Aggregation And Systematic Sampling For ARMA And ARMAX Models. Journal Of Econometrics 1, 133- 154.*
- Chambers, M. J. (1998). *Long Memory And Aggregation In Macroeconomic Time Series. International Economic Review 39, 1053-1072.*
- Chen, C. and Tiao, C.G. (1990). *Random Level-Shift Time Series Models, ARIMA Approximation And Level-Shift Detection, Journal Of Business And Economic Statistics, 8, 83-97.*
- Davidson, J. and Sibbertsen, P. (2005). *Generating Schemes For Long Memory Processes: Regimes, Aggregation And Linearity. Journal Of Econometrics, 128, 253-282.*
- Deo, R., C. Hurvich and L. Yi. (2006). *Forecasting Realized Volatility Using A Long-Memory Stochastic Volatility Model: Estimation, Prediction And Seasonal Adjustment. Journal Of Econometrics 131, 29.58.*
- Diebold, F. X. and Inoue, C. A. (2001). *Long memory and regime switching. Journal of Econometrics, 105(1), 131-159.*
- Ding, Z., R.F. Engle, and C.W.J. Granger. (1993). *A Long Memory Property Of Stock Market Returns And A New Model. Journal Of Empirical Finance 1, 83-106.*
- Engel, C.M. and J.D. Hamilton. (1990). *Long Swings In The Dollar: Are They In The Data And Do The Markets Know It?, American Economic Review 80:689-713*
- Geweke, J. and S. Porter-Hudak. (1983). *The Estimation And Application Of Long Memory Time Series Models. Journal Of Time Series Analysis 4, 221-237.*
- Granger, C.W.J. and Hyung, N. (2004). *Occasional Structural Breaks And Long Memory With Application To The S&P500 Absolute Stock Returns. Journal Of Empirical Finance 11, 399-421.*
- Granger, C. W. G. and R. Joyeux. (1980). *An introduction to long memory time series models and fractional differencing, Journal of Time Series Analysis 1, 15-29.*

- Granger, C. W. J. and Teräsvirta, T. (1999). *A Simple Nonlinear Time Series Model With Misleading Linear Properties*. *Economics Letters* 62(2), 161-165.
- Gourieroux, C. & J. Jasiak. (2001). *Memory And Infrequent Breaks*. *Economics Letters* 70, 29-41.
- Hamilton, J.D. (1989). *A New Approach To The Economic Analysis Of Non Stationarity Times Series And The Business Cycle*. *Econometrica* 57, 357-384.
- Hamilton, James D. (1994). *Time Series Analysis*, Princeton, Nj: Princeton University Press.
- Heinen, F, Michael. S and Sibbertsen. P. (2012). *Weak Identification In The ESTAR Model And A New Model*.
- Heinen, F, Michael. S and Sibbertsen. P. (2011). *Two Competitive Models And Their Identification Problem: The Estar And Tstar Model*.
- Hosking, J. (1981). *Fractional Differencing*, *Biometrika* 68(1), 165-176.
- Husrt, H.E. (1951). *Long-Term Storage Capacity Of Reservoirs*. *Transactions Of The American Society Of Civil Engineers*, Volume 116, Pages 770-799.
- Husrt, H. E. (1956). *Methods Of Using Long-Term Storage In Reservoirs*, *Proc.Inst. Civil Eng.*, 1,519-543.
- Iglesias, P., Jorquera, H., dan Palma, W. (2005). *Data Analysis Using Regression Model With Missing Observations And Long-Memory: An Application Study*. *Journal Of Computational Statistics And Data Analysis* 50, 2028-2043.
- Kim, Chang Jin. (1994). *Dynamic Linear Models With Markov-Switching*, *Journal Of Econometrics* 60, 1-22.
- Kuswanto, H. (2011). *A New Test Against Spurious Long Memory Using Temporal Aggregation*. *Journal Of Statistical Computation And Simulation*.
- Kuswanto, H. and Sibbertsen. P. (2008). *A Study On Spurious Long Memory In Nonlinear Time Series Models*. *Applied Mathematical Science*, 2(55), 2713-2734.
- Kuswanto, H. and Sibbertsen, P. (2007). *Can We Distinguish Between Common Nonlinear Time Series And Long Memory? Discussion Paper No. 178*, Leibniz Hannover University, Germany.
- Lim, K.S. and Tong, H. (1980). *Threshold Autoregressions, Limit Cycles, And Data*. *Journal Of The Royal Statistical Sociaty*, B 42, 245-92.
- Lobato, I.N. & N.E. Savin. (1998). *Real And Spurious Long-Memory Properties Of Stock-Market Data*. *Journal Of Business And Economics Statistics* 16, 261-268.
- Makridakis, McGee, dan Wheelwright. (1999). *Metode Aplikasi Dan Peramalan*, Edisi Kedua, Bina Rupa Aksara, Jakarta.
- Ohanissian, A., J.R. Russell and R.S. Tsay. (2005). *True Or Spurious Long Memory In Volatility: Does It Matter For Pricing Options? Unpublished Manuscript*. Graduate School Of Business, University Of Chicago.
- Ohanissian, A., J.R. Russell and R.S. Tsay. (2008). *True Or Spurious Long Memory? A New Test*. *Journal Of Business And Economics Statistics* 26, 161-175.

- Payá, I And Peel, D. A. (2004). *Temporal Aggregation Of An Estar Process : Some Implications For Purchasing Power Parity Adjustment*.
- Parke, W.R. (1999). *What Is Fractional Integration? Review Of Economics And Statistics* 81, 632-638.
- Sibbertsen, P. (2004). *Long Memory Versus Structural Change: An Overview. Statistical Papers* 45, 465-515.
- Sibbertsen, P. (2011). *Two Competitive Models And Their Identification Problem: The ESTAR And TSTAR Model*.
- Souza, L. R. (2008). *Spectral Properties Of Temporally Aggregated Long Memory Process. Brazilian Journal Of Probability And Statistics* 22(2), 135-155.
- Souza, L. R. (2005). *Temporal Aggregation And Bandwidth Selection In Estimating Long Memory. Journal Of Time Series Analysis* 28(5), 701-722.
- Souza, L. R. (2003). *Temporal Aggregation And Bandwidth Selection In Estimating Long Memory. Journal Of Time Series Analysis* 28(5), 701-722.
- Taylor, S.J. (2000). *Consequences For Option Pricing Of A Long Memory In Volatility. Unpublished Manuscript. Department Of Accounting And Finance, Lancaster University*.
- Teles, P. and W. W. S. Wei. (2002). *The Use Of Aggregate Time Series In Testing For Gaussianity. Journal Of Time Series Analysis* 23, 1, 95-116.
- Timmermann, A. (2000). *Moments Of Markov Switching Models. Journal Of Econometrics* (96): 75-111.
- Weiss, A. A. (1984). *Systematic Sampling And Temporal Aggregation In Time Series Models. Journal Of Economic* 26, 271-281.
- Wei, W. W. S. (2006). *Time Series Analysis Second Edition: Univariate And Multivariate Methods (2<sup>nd</sup> Eds)*. New York, United States Of America: Pearson Education.
- Zivot, E., & Wang, J. (2006). *Modelling Financial Time Series Models With S-PLUS*. New York: Springer.



*Halaman Ini Sengaja Dikosongkan*

### Lampiran A : Code Simulasi Stock Aggregation Untuk Long Memory

```

library(fracdiff)

skip <- function(k, repli, series, bandw.exp, parameter)
{

d_asli_agg <- matrix(nrow=k, ncol=repli)
long <- matrix(nrow=1, ncol=k)
d_asli <- matrix(nrow=1, ncol=k)
varimemo <- matrix(nrow=1, ncol=k)

jumlah <- matrix(nrow=1, ncol=k)

Z <- matrix(nrow=k, ncol=k)
maxi <- matrix(nrow=1, ncol=k)
longku <- matrix(nrow=k, ncol=repli)
for (rep in 1:repli)
{
dataku <- fracdiff.sim(series, d=parameter)$series
for (s in 1:k)
{
output <- convexion(dataku, s, bandw.exp)
long[,s] <- output$dakhir
d_asli[,s] <- GPHaggregate(dataku, s, bandw.exp)$dd
}
long1 <- as.vector(long)
d_asli1 <- as.vector(d_asli)
longku[,rep] <- long1
d_asli_agg[,rep] <- d_asli1
}
parameterd <- rowMeans(longku)
parameteragg <- rowMeans(d_asli_agg)
deviasidbias <- apply(longku, 1, sd)
deviasiagg <- apply(d_asli_agg, 1, sd)
print(parameterd)
list(parameterd=parameterd, parameteragg=parameteragg, longku=longku,
d_asli_agg=d_asli_agg, deviasidbias=deviasidbias, deviasiagg=deviasiagg)
}

```

### Lampiran B : Code Simulasi Stock Aggregation Untuk Markov Switching

```

skip <- function(k,repli,series, mu1,mu2,eta1,eta2,phi1,phi2,p11,p22, bandw.exp)
{
obs <- series+100

z <- matrix(nrow=obs,ncol=repli)
u1 <- matrix(nrow=obs,ncol=repli)
u2 <- matrix(nrow=obs,ncol=repli)
states <- matrix(nrow=obs,ncol=repli)
inno <- matrix(nrow=obs,ncol=repli)
ms <- matrix(nrow=obs,ncol=repli)

d_asli_agg<- matrix(nrow=k,ncol=repli)
long <- matrix(nrow=1,ncol=k)
d_asli <- matrix(nrow=1,ncol=k)
longku <- matrix(nrow=k,ncol=repli)
varimemo <- matrix(nrow=1,ncol=k)
jumlah <- matrix(nrow=1,ncol=k)
Z <- matrix(nrow=k,ncol=k)
maxi <- matrix(nrow=1,ncol=k)
tt <-matrix(nrow=1,ncol=repli)
xt000 <- matrix(ncol=1,nrow=obs)
dasli <- matrix(ncol=repli,nrow=1)

for (ul in 1:repli)
{
z[,ul] <- runif(obs,0,1)
u1[,ul] <- rnorm(obs,0,1)
u2[,ul] <- rnorm(obs,0,1)

if (z[1,ul] >= (1-p22)/(2-p11-p22))
{
states[1,ul] <- 2
inno[1,ul] <- mu2+u2[1,ul]
}

else if (z[1,ul] < (1-p22)/(2-p11-p22))
{
states[1,ul] <- 1
inno[1,ul] <- mu1+u1[1,ul]
}

ms[1,ul]=inno[1,ul]
}
}

```

```

for(nn in 2:obs)
{
  if (states[nn-1,ul] == 1 && z[nn,ul] >= p11)
  {
    states[nn,ul]=2
  }
  else if (states[nn-1,ul] == 1 && z[nn,ul] < p11)
  {
    states[nn,ul]=1
  }
  if (states[nn-1,ul] == 2 && z[nn,ul] >= p22)
  {
    states[nn,ul]=1
  }
  else if (states[nn-1,ul] == 2 && z[nn,ul] < p22)
  {
    states[nn,ul]=2
  }

  if (states[nn,ul] == 1)
  {
    inno[nn,ul]=mu1+eta1*ms[nn-1,ul]+phi1*inno[nn-1,ul]+u1[nn,ul]
  }
  else if (states[nn,ul] == 2)
  {
    inno[nn,ul]=mu2+eta2*ms[nn-1,ul]+phi2*inno[nn-1,ul]+u2[nn,ul]
  }
  ms[nn,ul] <- inno[nn,ul]
}
dataku <- ms[101:obs,ul]

for (s in 1:k)
{
  output <- convexion(dataku,s, bandw.exp)
  long[,s] <- output$dakhir
  d_asli[,s] <- GPHaggregate(dataku,s,bandw.exp)$dd
}
long1 <- as.vector(long)
d_asli1 <- as.vector(d_asli)
longku[,ul] <- long1
d_asli_agg[,ul] <- d_asli1
}

```



```
parameterd <- rowMeans(longku)
parameteragg <- rowMeans(d_asli_agg)
devisidbias <- apply(longku, 1, sd)
devasiagg <- apply(d_asli_agg, 1, sd)
print(parameterd)
list(parameterd=parameterd, parameteragg=parameteragg, longku=longku,
d_asli_agg=d_asli_agg, devisidbias=devisidbias, deviasiagg=deviasiagg)
}
```

UNIVERSITAS TERBUKA

**Lampiran C : Code Simulasi Stock Aggregation Level Shift**

```

library(fractal)
library(GeneCycle)
library(Rlab)
skip <- function(k,repli,series, pp, bandw.exp)
{
obs <- series

d_asli_agg<- matrix(nrow=k,ncol=repli)
long <- matrix(nrow=1,ncol=k)
d_asli <- matrix(nrow=1,ncol=k)
longku <- matrix(nrow=k,ncol=repli)
varimemo <- matrix(nrow=1,ncol=k)
jumlah <- matrix(nrow=1,ncol=k)
Z <- matrix(nrow=k,ncol=k)
maxi <- matrix(nrow=1,ncol=k)
tt <-matrix(nrow=1,ncol=repli)
xt000 <- matrix(ncol=1,nrow=obs)
mu <- matrix(ncol=1,nrow=obs)
dasli <- matrix(ncol=repli,nrow=1)

for (rep in 1:repli)
{
aa<-rnorm(obs,0,1)
aa <- as.vector(aa)
jjj <- rbern(series, pp)
jj <- as.vector(jjj)

mu[1] <- 0
xt000[1] <- aa[1]
for(ss in 2:obs)
{
mu[ss] <- (1-jj[ss])*mu[ss-1]+jj[ss]*aa[ss]
xt000[ss] <- mu[ss] +aa[ss]
#xt000[ss] <- mu[ss]+aa[ss]#
}
xt00 <- xt000

xt001 <- xt00
dataku <- as.vector(xt001)

for (s in 1:k)
{
output <- convextion(dataku,s, bandw.exp)
}
}

```

```
long[,s] <- output$dakhir
d_asli[,s] <- GPHaggregate(dataku,s,bandw.exp)$dd
}
long1 <- as.vector(long)
d_asli1 <- as.vector(d_asli)
longku[,rep] <- long1
d_asli_agg[,rep] <- d_asli1
}
parameterd <- rowMeans(longku,na.rm=TRUE)
parameteragg <- rowMeans(d_asli_agg,na.rm=TRUE)
deviasidbias <- apply(longku, 1, sd)
deviasiagg <- apply(d_asli_agg, 1, sd)
print(parameterd)
list(parameterd=parameterd, parameteragg=parameteragg, longku=longku,
d_asli_agg=d_asli_agg, deviasidbias=deviasidbias, deviasiagg=deviasiagg)
}
```

**Lampiran D : Code Simulasi Stock Aggregation Stop Break**

```

library(fractal)
library(GeneCycle)
library(Rlab)
skip <- function(k,repli,series, gamma1, alpha1,alpha2, bandw.exp)
{
obs <- series

d_asli_agg<- matrix(nrow=k,ncol=repli)
long <- matrix(nrow=1,ncol=k)
d_asli <- matrix(nrow=1,ncol=k)
longku <- matrix(nrow=k,ncol=repli)
varimemo <- matrix(nrow=1,ncol=k)
jumlah <- matrix(nrow=1,ncol=k)
Z <- matrix(nrow=k,ncol=k)
maxi <- matrix(nrow=1,ncol=k)
tt <-matrix(nrow=1,ncol=repli)
xt000 <- matrix(ncol=1,nrow=obs)
mu <- matrix(ncol=1,nrow=obs)
dasli <- matrix(ncol=repli,nrow=1)

for (rep in 1:repli)
{
aa<-rnorm(obs,0,1)
aa <- as.vector(aa)

mu[1] <- 0
xt000[1] <- aa[1]
for(ss in 2:obs)
{
mu[ss] <- mu[ss-1]+(((aa[ss-1]^2)*aa[ss-1])/(gamma1+(aa[ss-1]^2)))
xt000[ss] <- mu[ss] +aa[ss]
#xt000[ss] <- mu[ss]+aa[ss]#
}
xt00 <- xt000

xt001 <- xt00[101:obs]
dataku <- as.vector(xt001)

for (s in 1:k)
{
output <- convexion(dataku,s, bandw.exp)
long[,s] <- output$dakhir
d_asli[,s] <- GPHaggregate(dataku,s,bandw.exp)$dd
}
}

```

```
}  
long1 <- as.vector(long)  
d_asli1 <- as.vector(d_asli)  
longku[,rep] <- long1  
d_asli_agg[,rep] <- d_asli1  
}  
parameterd <- rowMeans(longku,na.rm=TRUE)  
parameteragg <- rowMeans(d_asli_agg,na.rm=TRUE)  
deviasidbias <- apply(longku, 1, sd)  
deviasiagg <- apply(d_asli_agg, 1, sd)  
print(parameterd)  
list(parameterd=parameterd, parameteragg=parameteragg, longku=longku,  
d_asli_agg=d_asli_agg, deviasidbias=deviasidbias, deviasiagg=deviasiagg)  
}
```

### Lampiran E : Code Evolution\_Modification0\_Hfromdatanonaggregate

```

GPHnonaggregate <- function (x,bandw.exp)
{
  x <- as.numeric(na.fail(as.ts(x)))
  if (any(is.na(x)))
    stop("NAs in x")
  if (NCOL(x) > 1)
    stop("only implemented for univariate time series")
  n <- length(x)
  g <- trunc(n^bandw.exp)
  j <- 1:g
  kk <- 1:(n - 1)
  w <- 2 * pi * j/n
  mx <- mean(x)
  var.x <- sum((x - mx)^2)/n
  cov.x <- numeric(n - 1)
  for (k in kk) cov.x[k] <- sum((x[1:(n - k)] - mx) * (x[(1 +
    k):n] - mx))/n
  periodogram <- numeric(g)
  for (i in 1:g) periodogram[i] <- var.x + 2 * sum(cov.x *
    cos(w[i] * kk))
  pos <- j[periodogram > 0]
  y.reg <- log(periodogram[pos]/(2 * pi))
  x.reg <- 2 * log(2 * sin(w[pos]/2))
  fit <- lm(y.reg ~ x.reg)
  d.GPH <- coef(fit)[2]
  const <- coef(fit)[1]
  names(d.GPH) <- NULL
  x.r2 <- sum((x.reg - mean(x.reg))^2)
  var.d <- pi^2/(6 * x.r2)
  var.reg <- sum(resid(fit)^2)/((g - 1) * x.r2)
  list(d = -d.GPH, const = -const, sd.as = sqrt(var.d), sd.reg = sqrt(var.reg))
}

GPHaggregate <- function (data,M,bandw.exp)
{
  y <- data
  N <- length(y)

  mm <- M
  gg <- round(N/mm)
  x <- matrix(nrow=gg,ncol=1)

  for (t in 1:gg)
  {
    x[t,] <- y[mm*t]
  }
}

```

```

}
xx <- as.vector(x)
xx <- na.omit(xx)

dd <- GPHnonaggregate(xx, bandw.exp)$d
list (dd=dd)
}
biasH <- function (data,m,bandw.exp)
{
  d1 <- GPHnonaggregate(data, bandw.exp)$d
  x <-data
  x <- as.numeric(na.fail(as.ts(x)))
  if (any(is.na(x)))
    stop("NAs in x")
  if (NCOL(x) > 1)
    stop("only implemented for univariate time series")
  n <- length(x)
  g <- trunc(n^bandw.exp)
  j <- 1:g
  kk <- 1:(n - 1)
  w <- 2 * pi * j/n
  mx <- mean(x)
  var.x <- sum((x - mx)^2)/n
  cov.x <- numeric(n - 1)
  for (k in kk) cov.x[k] <- sum((x[1:(n - k)] - mx) * (x[(1 +
    k):n] - mx))/n
  periodogram <- numeric(g)
  for (i in 1:g) periodogram[i] <- var.x + 2 * sum(cov.x *
    cos(w[i] * kk))
  pos <- j[periodogram > 0]
  y.reg <- log(periodogram[pos]/(2 * pi))
  x.reg <- log(2 * sin(w[pos]/2))
  fit <- lm(x.reg~y.reg)
  d.GPH <- - coef(fit)[2]
  names(d.GPH) <- NULL
  x.r2 <- sum((x.reg - mean(x.reg))^2)
  var.d <- pi^2/(6 * x.r2)
  var.reg <- sum(resid(fit)^2)/((g - 1) * x.r2)

  f <- matrix(nrow=g,ncol=m)
  for(i in 1:m)
  {
    f[,i]=(cos(w[pos]/2)^(i-1))*cos((m-i)*w[pos]/2)
  }
  fnj <- rowSums(f)

```

```

fnj_mean=mean(fnj)
logfnj = log(fnj)
logfnj_mean=mean(logfnj)

zj <- log(2 * sin(w[pos]/2))^2
zj_mean <- mean(zj)

zjn <- zj + 2*log (fnj)
zjn_mean <- mean (zjn)

H <- matrix(nrow=g,ncol=m)
for(i in 0:m-1)
{
H[,i+1] <- ((sin(i*pi/m)/tan(w[pos]/2))+cos(i*pi/m))^(2*d1)
}

Hjn <- rowSums(H)

nom1 <- sum((zj-zj_mean)*log(fnj))
nom2 <- sum((logfnj-logfnj_mean)^2)
nom3 <- sum((zjn-zjn_mean)*log(Hjn))
denom <- sum((zjn-zjn_mean)^2)

HBT <-(-d1*(2*nom1+4*nom2)-nom3)/denom
list(HBT=HBT, d = -d.GPH, d1=d1, sd.as = sqrt(var.d), sd.reg = sqrt(var.reg))
}

convexion <- function(data,m,bandw.exp=bandw.exp)
{
dn <- numeric(m)
dagg <- numeric(m)
for (i in 1:m)
{
dn[i]<-GPHaggregate(data,i,bandw.exp=bandw.exp)$dd-
biasH(data,i,bandw.exp=bandw.exp)$HBT
dagg[i] <-GPHaggregate(data,i,bandw.exp=bandw.exp)$dd
}
dn <- dn
dagg <- dagg
cum_average <- cumsum(dn)/seq_along(dn)
dakhir <- mean(dn)
biasd <- GPHnonaggregate(data,bandw.exp=bandw.exp)$d - cum_average
bias_dagg <- GPHnonaggregate(data,bandw.exp=bandw.exp)$d - dagg
list(dakhir=dakhir, dagg=dagg, dn=dn, cum_average=cum_average, biasd=biasd,
bias_dagg=bias_dagg)
}

```



**Lampiran F : Code Output Nilai d**

```
library(fracdiff)
skip <- function(dataku, k, bandw.exp)
{
  long <- matrix(nrow=k,ncol=1)
  d_asli <- matrix(nrow=k,ncol=1)

  for (s in 1:k)
  {
    output <- convexion(dataku,s, bandw.exp)
    long[s] <- output$dakhir
    d_asli[s] <- GPHaggregate(dataku,s,bandw.exp)$dd
  }
  data.frame(d_corrected=long, d_asli=d_asli)
}
```

### Lampiran G : Ouput ARFIMA Indosat dengan Time Series Modelling (TSM)

TSM4.31.13-03-10 Run 103 at 18:38:59 on 22-05-2013

Data file is D:\saham.csv

Dependent Variable is Indosat

3126 observations (4-3129) used for estimation with 3 pre-sample observations.

Estimation Method: Conditional ML (Time Domain)

Gaussian Likelihood

ARFIMA(3,d,3)

Strong convergence

iteration time: 1:10.82

	Estimate	Std. Err.	t Ratio	p-Value
Intercept	0.01724	0.0015	11.495	0
ARFIMA d	0.18888	0.02708	6.975	0
AR1	-0.61924	0.21828	-2.837	0.005
AR2	-0.64956	0.10966	-5.923	0
AR3	-0.66409	0.17441	-3.808	0
MA1	-0.59289	0.22182	-2.673	0.008
MA2	-0.6402	0.10318	-6.205	0
MA3	-0.68169	0.18032	-3.78	0
Error Variance^(1/2)	0.01775	0.0007	-----	-----
Log Likelihood	= 8167.22			
Schwarz Criterion	= 8131.01			
Hannan-Quinn Criterion	= 8148.46			
Akaike Criterion	= 8158.22			
Sum of Squares	= 0.9844			
R-Squared	= 0.0671			
R-Bar-Squared	= 0.0647			
Residual SD	= 0.0178			
Residual Skewness	= 2.7757			
Residual Kurtosis	= 21.7286			
Jarque-Bera Test	= 49700.7 {0}			
Box-Pierce (residuals):	Q(6)	= 13.37	{0.038}	
Box-Pierce (squared residuals):	Q(12)	= 82.9587	{0}	

### Lampiran H : Ouput ARFIMA Telkom dengan Time Series Modelling

TSM4.31.13-03-10 Run 150 at 21:38:03 on 22-05-2013

Data file is D:\saham.csv

Dependent Variable is Telkom  
 3128 observations (2-3129) used for estimation  
 with 1 pre-sample observations.  
 Estimation Method: Conditional ML (Time Domain)  
 Gaussian Likelihood  
 ARFIMA(1,d,1)

Strong convergence  
 iteration time: 1:07.78

	Estimate	Std. Err.	t Ratio	p-Value
Intercept	0.01608	0.00128	12.559	0
ARFIMA d	0.20219	0.03438	5.881	0
AR1	0.24593	0.12003	2.049	0.041
MA1	0.30794	0.11013	2.796	0.005
Error Variance <sup>(1/2)</sup>	0.01644	0.0005	-----	-----

Log Likelihood	= 8411.35			
Schwarz Criterion	= 8391.23			
Hannan-Quinn Criterion	= 8400.92			
Akaike Criterion	= 8406.35			
Sum of Squares	= 0.8455			
R-Squared	= 0.0631			
R-Bar-Squared	= 0.0619			
Residual SD	= 0.0165			
Residual Skewness	= 2.1101			
Residual Kurtosis	= 13.4061			
Jarque-Bera Test	= 16434.6	{0}		
Box-Pierce (residuals): Q(10)	= 14.3017	{0.16}		
Box-Pierce (squared residuals): Q(12)	= 158.376	{0}		

\*\*\*\*\*

TSM4.31.13-03-10 Run 161 at 21:58:43 on 22-05-2013

Data file is D:\saham.csv

Dependent Variable is Telkom  
 3127 observations (3-3129) used for estimation  
 with 2 pre-sample observations.  
 Estimation Method: Conditional ML (Time Domain)  
 Gaussian Likelihood  
 ARFIMA(2,d,2)

Strong convergence  
 iteration time: 1:29.23

	Estimate	Std. Err.	t Ratio	p-Value
Intercept	0.01617	0.00113	14.307	0
ARFIMA d	0.17493	0.02776	6.301	0
AR1	0.61813	0.16861	3.666	0

AR2	-0.37651	0.13214	-2.849	0.004
MA1	0.64879	0.1693	3.832	0
MA2	-0.39411	0.12691	-3.105	0.002
Error Variance <sup>(1/2)</sup>	0.01644	0.0005	-----	-----
Log Likelihood		= 8409.73		
Schwarz Criterion		= 8381.57		
Hannan-Quinn Criterion		= 8395.14		
Akaike Criterion		= 8402.73		
Sum of Squares		= 0.8446		
R-Squared		= 0.0616		
R-Bar-Squared		= 0.0598		
Residual SD		= 0.0165		
Residual Skewness		= 2.117		
Residual Kurtosis		= 13.4274		
Jarque-Bera Test		= 16502.5	{0}	
Box-Pierce (residuals): Q(8)		= 15.3192	{0.053}	
Box-Pierce (squared residuals): Q(12)		= 158.931	{0}	
*****				
TSM4.31.13-03-10 Run 195 at 23:42:49 on 22-05-2013				
Data file is D:\saham.csv				
-----				
Dependent Variable is Telkom				
3124 observations (6-3129) used for estimation				
with 5 pre-sample observations.				
Estimation Method: Conditional ML (Time Domain)				
Gaussian Likelihood				
ARFIMA(5,d,5)				
Strong convergence				
iteration time: 1:32.84				
	Estimate	Std. Err.	t Ratio	p-Value
Intercept	0.01615	0.00119	13.575	0
ARFIMA d	0.18712	0.02772	6.75	0
AR1	-1.23858	0.25823	-4.796	0
AR2	-1.23331	0.20673	-5.966	0
AR3	0.62215	0.25044	-2.484	0.013
AR4	-0.40282	0.09701	-4.152	0
AR5	-0.37216	0.128	-2.907	0.004
MA1	-1.19504	0.27339	-4.371	0
MA2	-1.18646	0.20706	-5.73	0
MA3	-0.55768	0.2624	-2.125	0.034
MA4	-0.3882	0.08753	-4.435	0
MA5	-0.34763	0.15145	-2.295	0.022
Error Variance <sup>(1/2)</sup>	0.01637	0.0005	-----	-----
Log Likelihood		= 8413.23		
Schwarz Criterion		= 8360.93		
Hannan-Quinn Criterion		= 8386.12		
Akaike Criterion		= 8400.23		
Sum of Squares		= 0.8376		
R-Squared		= 0.0686		

R-Bar-Squared	=	0.065	
Residual SD	=	0.0164	
Residual Skewness	=	2.0949	
Residual Kurtosis	=	13.2067	
Jarque-Bera Test	=	15845.4	{0}
Box-Pierce (residuals): Q(2)	=	5.1442	{0.076}
Box-Pierce (squared residuals): Q(12)	=	159.407	{0}

UNIVERSITAS TERBUKA

### Lampiran I : Uji White Noise dan Normality Test dengan Ljung-Box Series Indosat

```

> source("D:\\TESIS-GEDE\\Proposal Gede\\residualsindosat.csv")
Error in eval(expr, envir, enclos) : object 'Residuals' not found
>y=read.csv("D:\\TESIS-
GEDE\\ProposalGede\\residualsindosat.csv",header=TRUE)
Error: could not find function "read.csv"
>y=read.csv("D:\\TESIS-
GEDE\\ProposalGede\\residualsindosat.csv",header=TRUE)
> residuals=y$residuals
> library(ltsa)
> library(parallel)
> library(arfima)
> m=arfima(residuals,order=c(3,0,3))
Note: autoweed is ON. It is possible, but not likely,
that unique modes may be lost.
Beginning the fits with 128 starting values.
There were 43 warnings (use warnings() to see the m)
> library(fBasics)
Loading required package: MASS
Loading required package: timeDate
Loading required package: timeSeries
Attaching package: 'fBasics'
The following object(s) are masked from 'package:base':
  norm
> ksnormTest(m[[1]]$residuals)
Error in m[[1]]$residuals : $ operator is invalid for atomic vectors
> ksnormTest(m$modes[[1]]$residuals)
Title:
One-sample Kolmogorov-Smirnov test
Test Results:
STATISTIC:
D: 0.4868
P VALUE:
Alternative Two-Sided: < 2.2e-16
Alternative Less: < 2.2e-16
Alternative Greater: < 2.2e-16
Description:
Mon Jul 01 03:13:56 2013 by user: Owner
> library(stats)
> Box.test(m$modes[[1]]$residuals,lag=3,type="Ljung")
Box-Ljung test
data: m$modes[[1]]$residuals
X-squared = 0.8922, df = 3, p-value = 0.8273
> Box.test(m$modes[[1]]$residuals,lag=6,type="Ljung")
Box-Ljung test

```

```
data: m$modes[[1]]$residuals
X-squared = 5.975, df = 6, p-value = 0.426
> Box.test(m$modes[[1]]$residuals,lag=9,type="Ljung")
  Box-Ljung test
data: m$modes[[1]]$residuals
X-squared = 8.3704, df = 9, p-value = 0.4973
> Box.test(m$modes[[1]]$residuals,lag=12,type="Ljung")
  Box-Ljung test
data: m$modes[[1]]$residuals
X-squared = 13.5345, df = 12, p-value = 0.3314
```

UNIVERSITAS TERBUKA

### Lampiran J : Uji White Noise Dan Normality Test Ljung-Box Series Telkom

```

> source("H:\\restelkom1.csv")
Error in eval(expr, envir, enclos) : object 'Residuals' not found
> y=read.csv(("H:\\restelkom1.csv",header=TRUE)
Error: unexpected ',' in "y=read.csv(("H:\\restelkom1.csv", "
> y=read.csv("H:\\restelkom1.csv",header=TRUE)
> residuals=y$Residuals
> library(ltsa)
> library(parallel)
> library(arfima)
Attaching package: 'arfima'
The following object(s) are masked from 'package:stats':
  BIC
> m=arfima(residuals,order=c(1,0,1))
Note: autoweed is ON. It is possible, but not likely,
that unique modes may be lost.
Beginning the fits with 8 starting values.
Warning messages:
1: In sqrt(diag(solve(-hess))) : NaNs produced
2: In sqrt(diag(solve(-hess))) : NaNs produced
3: In sqrt(diag(solve(-hess))) : NaNs produced
4: In sqrt(diag(solve(-hess))) : NaNs produced
5: In sqrt(diag(solve(-hess))) : NaNs produced
6: In sqrt(diag(solve(-hess))) : NaNs produced
> library(fBasics)
Loading required package: MASS
Loading required package: timeDate
Loading required package: timeSeries
Attaching package: 'fBasics'
The following object(s) are masked from 'package:base':
  norm
Warning messages:
1: package 'fBasics' was built under R version 2.15.3
2: package 'MASS' was built under R version 2.15.3
3: package 'timeDate' was built under R version 2.15.3
4: package 'timeSeries' was built under R version 2.15.3
> ksnormTest(m$modes[[1]]$residuals)
Title:
  One-sample Kolmogorov-Smirnov test
Test Results:
STATISTIC:
  D: 0.4874
P VALUE:
  Alternative Two-Sided: < 2.2e-16
  Alternative Less: < 2.2e-16
  Alternative Greater: < 2.2e-16

```



```

Description:
Mon Jul 01 01:16:31 2013 by user: Gede
> library(stats)
> Box.test(m$modes[[1]]$residuals, lag = 3, type = "Ljung")
  Box-Ljung test
data: m$modes[[1]]$residuals
X-squared = 0.3224, df = 3, p-value = 0.9558
> Box.test(m$modes[[1]]$residuals, lag = 6, type = "Ljung")
  Box-Ljung test
data: m$modes[[1]]$residuals
X-squared = 1.6295, df = 6, p-value = 0.9504
> Box.test(m$modes[[1]]$residuals, lag = 9, type = "Ljung")
  Box-Ljung test
data: m$modes[[1]]$residuals
X-squared = 7.1194, df = 9, p-value = 0.6247
> Box.test(m$modes[[1]]$residuals, lag = 12, type = "Ljung")
  Box-Ljung test
data: m$modes[[1]]$residuals
X-squared = 10.1815, df = 12, p-value = 0.6
-----
> source("H:\\restelkom2.csv")
Error in eval(expr, envir, enclos) : object 'Residuals' not found
> y=read.csv("H:\\restelkom2.csv",header=TRUE)
> residual=y$Residuals
> m=arfima(residuals,order=c(2,0,2))
Note: autoweed is ON. It is possible, but not likely,
that unique modes may be lost.
Beginning the fits with 32 starting values.
Warning messages:
1: In sqrt(diag(solve(-hess))) : NaNs produced
2: In sqrt(diag(solve(-hess))) : NaNs produced
3: In sqrt(diag(solve(-hess))) : NaNs produced
4: In sqrt(diag(solve(-hess))) : NaNs produced
5: In sqrt(diag(solve(-hess))) : NaNs produced
6: In sqrt(diag(solve(-hess))) : NaNs produced
7: In sqrt(diag(solve(-hess))) : NaNs produced
8: In sqrt(diag(solve(-hess))) : NaNs produced
9: In sqrt(diag(solve(-hess))) : NaNs produced
10: In sqrt(diag(solve(-hess))) : NaNs produced
> ksnormTest(m$modes[[1]]$residuals)
Title:
One-sample Kolmogorov-Smirnov test
Test Results:
STATISTIC:
D: 0.4872
P VALUE:
Alternative Two-Sided: < 2.2e-16

```

```

Alternative Less: < 2.2e-16
Alternative Greater: < 2.2e-16
Description:
Mon Jul 01 01:52:49 2013 by user: Gede
> Box.test(m$modes[[1]]$residuals, lag = 3, type = "Ljung")
Box-Ljung test
data: m$modes[[1]]$residuals
X-squared = 0.646, df = 3, p-value = 0.8858
> Box.test(m$modes[[1]]$residuals, lag = 6, type = "Ljung")
Box-Ljung test
data: m$modes[[1]]$residuals
X-squared = 5.4231, df = 6, p-value = 0.4908
> Box.test(m$modes[[1]]$residuals, lag = 9, type = "Ljung")
Box-Ljung test
data: m$modes[[1]]$residuals
X-squared = 7.8342, df = 9, p-value = 0.5509
> Box.test(m$modes[[1]]$residuals, lag = 12, type = "Ljung")
Box-Ljung test
data: m$modes[[1]]$residuals
X-squared = 10.6018, df = 12, p-value = 0.5633
-----
> source("H:\\restelkom3.csv")
Error in eval(expr, envir, enclos) : object 'Residuals' not found
> y=read.csv("H:\\restelkom3.csv",header=TRUE)
> residuals=y$Residuals
> ksnormTest(m$modes[[1]]$residuals)
Title:
One-sample Kolmogorov-Smirnov test
Test Results:
STATISTIC:
D: 0.4872
P VALUE:
Alternative Two-Sided: < 2.2e-16
Alternative Less: < 2.2e-16
Alternative Greater: < 2.2e-16
Description:
Mon Jul 01 01:57:03 2013 by user: Gede
> Box.test(m$modes[[1]]$residuals, lag = 3, type = "Ljung")
Box-Ljung test
data: m$modes[[1]]$residuals
X-squared = 0.646, df = 3, p-value = 0.8858
> Box.test(m$modes[[1]]$residuals, lag = 6, type = "Ljung")
Box-Ljung test
data: m$modes[[1]]$residuals
X-squared = 5.4231, df = 6, p-value = 0.4908
> m=arfima(residuals,order=c(3,0,3))
Note: autoweed is ON. It is possible, but not likely,

```

that unique modes may be lost.  
Beginning the fits with 128 starting values.

```
> m=arfima(residuals,order=c(5,0,5))
```

Note: autoweed is ON. It is possible, but not likely,  
that unique modes may be lost.

Beginning the fits with 2048 starting values.

There were 50 or more warnings (use warnings() to see the first 50)

```
> ksnormTest(m$modes[[1]]$residuals)
```

Title:

One-sample Kolmogorov-Smirnov test

Test Results:

STATISTIC:

D: 0.4872

P VALUE:

Alternative Two-Sided: < 2.2e-16

Alternative Less: < 2.2e-16

Alternative Greater: < 2.2e-16

Description:

Wed Jul 03 14:45:55 2013 by user: Gede

```
> Box.test(m$modes[[1]]$residuals, lag = 3, type = "Ljung")
```

Box-Ljung test

data: m\$modes[[1]]\$residuals

X-squared = 0.0774, df = 3, p-value = 0.9944

```
> Box.test(m$modes[[1]]$residuals, lag = 6, type = "Ljung")
```

Box-Ljung test

data: m\$modes[[1]]\$residuals

X-squared = 0.2356, df = 6, p-value = 0.9998

```
> Box.test(m$modes[[1]]$residuals, lag = 9, type = "Ljung")
```

Box-Ljung test

data: m\$modes[[1]]\$residuals

X-squared = 0.6479, df = 9, p-value = 0.9999

```
> Box.test(m$modes[[1]]$residuals, lag = 12, type = "Ljung")
```

Box-Ljung test

data: m\$modes[[1]]\$residuals

X-squared = 2.3571, df = 12, p-value = 0.9986

### Lampiran K : Markov Switching Dengan TSM Series Indosat

Time Series Modelling v4.31.13-03-10 (c)James Davidson, 2002-10  
Copy licenced to Heri Kuswanto

D:\saham.csv opened.

\*\*\*\*\*

TSM4.31.13-03-10 Run 267 at 5:17:26 on 1-07-2013

Data file is D:\saham.csv

Dependent Variable is Indosat

3128 observations (2-3129) used for estimation  
with 1 pre-sample observations.

Estimation Method: Conditional ML (Time Domain)

Gaussian Likelihood

Markov-switching model with 2 regimes

Parameters for Regimes > 1 are differences from Regime 1.

ARIMA(1,0,0)

Weak convergence (no improvement in line search)

iteration time: 48.40

Markov Transition Probabilities

	P(.,1)	P(.,2)
P(1.)	0.96096	0.74292
P(2.)	0.039038	0.25708

	Estimate	Std. Err.	t Ratio	p-Value
Logistic, t(1,1)	3.2034	0.6382	-----	-----
Logistic, t(1,2)	1.06119	0.4507	-----	-----

Non-switching parameters shown as Regime 1.

Regime 1:

Intercept	0.01578	0.00037	42.66	0
AR1	0.10024	0.03868	2.592	0.01
Error Variance^(1/2)	0.01709	0.0007	-----	-----

Regime 2:

AR1	1.07575	0.3417	3.148	0.002
Log Likelihood	= 8200.54			
Schwarz Criterion	= 8176.39			
Hannan-Quinn Criterion	= 8188.03			
Akaike Criterion	= 8194.54			
Sum of Squares	= 0.8817			
R-Squared	= 0.1771			
R-Bar-Squared	= 0.1757			
Residual SD	= 0.0168			
Residual Skewness	= 2.6825			

Residual Kurtosis = 18.3686				
Jarque-Bera Test = 34535.4				{0}
Box-Pierce (residuals):	Q(11) = 63.9727			{0}
Box-Pierce (squared residuals):	Q(12) = 49.2321			{0}
<hr/>				
TSM4.31.13-03-10 Run 438 at 12:02:50 on 3-07-2013				
Data file is D:\saham.csv				
<hr/>				
Dependent Variable is Indosat				
3129 observations (1-3129) used for estimation.				
Estimation Method: Conditional ML (Time Domain)				
Gaussian Likelihood				
Markov-switching model with 2 regimes				
Parameters for Regimes > 1 are differences from Regime 1.				
ARIMA(0,0,1)				
Strong convergence				
iteration time: 30.23				
Markov Transition Probabilities				
	P(. 1)	P(. 2)		
P(1.)	0.99379	0.33340		
P(2.)	0.0062056	0.66660		
	Estimate	Std. Err.	t Ratio	p-Value
Logistic, t(1,1)	5.07608	1.9103	-----	-----
Logistic, t(1,2)	-0.69284	2.3996	-----	-----
Non-switching parameters shown as Regime 1.				
Regime 1:				
Intercept	0.01611	0.00039	41.312	0
MA1	-0.13604	0.02235	-6.087	0
Error Variance^(1/2)	0.01711	0.0008	-----	-----
Regime 2:				
MA1	-0.84443	0.02229	-37.884	0
Log Likelihood = 8206.05				
Schwarz Criterion = 8181.91				
Hannan-Quinn Criterion = 8193.54				
Akaike Criterion = 8200.05				
Sum of Squares = 0.8954				
R-Squared = 0.17				
R-Bar-Squared = 0.1686				
Residual SD = 0.0169				
Residual Skewness = 2.7				
Residual Kurtosis = 22.3226				
Jarque-Bera Test = 52478.7				{0}
Box-Pierce (residuals):	Q(11) = 147.01			{0}
Box-Pierce (squared residuals):	Q(12) = 120.966			{0}

## Lampiran L : Uji Whitenoise Dan Uji Normality Markov Switching Indosat

```

> source("D:\\residualindosatmarkov1.csv")
Error in eval(expr, envir, enclos) : object 'Residuals' not found
> residuals=read.csv(("D:\\residualindosatmarkov1.csv",header=TRUE)
Error: unexpected ',' in "residuals=read.csv(("D:\\residualindosatmarkov1.csv",
> y=read.csv(("D:\\residualindosatmarkov1.csv",header=TRUE)
Error: unexpected ',' in "y=read.csv(("D:\\residualindosatmarkov1.csv",
> residuals=read.csv("D:\\residualindosatmarkov1.csv",header=TRUE)
> library(fBasics)
Loading required package: MASS
Loading required package: timeDate
Loading required package: timeSeries
Attaching package: 'fBasics'
The following object(s) are masked from 'package:base':
  norm
> ksnormTest(residuals)
Title:
  One-sample Kolmogorov-Smirnov test
Test Results:
STATISTIC:
  D: 0.4901
P VALUE:
  Alternative Two-Sided: < 2.2e-16
  Alternative Less: < 2.2e-16
  Alternative Greater: < 2.2e-16
Description:
  Tue Jul 02 13:26:13 2013 by user: Owner
Warning messages:
1: In ks.test(x, "pnorm", alternative = "two.sided") :
  ties should not be present for the Kolmogorov-Smirnov test
2: In ks.test(x, "pnorm", alternative = "less") :
  ties should not be present for the Kolmogorov-Smirnov test
3: In ks.test(x, "pnorm", alternative = "greater") :
  ties should not be present for the Kolmogorov-Smirnov test
> library(stats)
> Box.test(residuals,lag=3,type="Ljung")
  Box-Ljung test
data: residuals
X-squared = 43.151, df = 3, p-value = 2.286e-09
> Box.test(residuals,lag=6,type="Ljung")
  Box-Ljung test
data: residuals
X-squared = 50.9111, df = 6, p-value = 3.086e-09
> Box.test(residuals,lag=9,type="Ljung")
  Box-Ljung test

```

```

data: residuals
X-squared = 60.8754, df = 9, p-value = 9.088e-10
> Box.test(residuals,lag=12,type="Ljung")
  Box-Ljung test
data: residuals
X-squared = 64.0952, df = 12, p-value = 4.005e-09
-----
> source("D:\\residualindosatmarkov2.csv")
Error in eval(expr, envir, enclos) : object 'Residuals' not found
> residuals=read.csv("D:\\residualindosatmarkov2.csv",header=TRUE)
> ksnormTest(residuals)
Title:
  One-sample Kolmogorov-Smirnov test
Test Results:
  STATISTIC:
  D: 0.4895
  P VALUE:
  Alternative Two-Sided: < 2.2e-16
  Alternative Less: < 2.2e-16
  Alternative Greater: < 2.2e-16
Description:
  Tue Jul 02 13:30:38 2013 by user: Owner
> Box.test(residuals,lag=3,type="Ljung")
  Box-Ljung test
data: residuals
X-squared = 85.5881, df = 3, p-value < 2.2e-16
> Box.test(residuals,lag=6,type="Ljung")
  Box-Ljung test
data: residuals
X-squared = 117.3843, df = 6, p-value < 2.2e-16
> Box.test(residuals,lag=9,type="Ljung")
  Box-Ljung test
data: residuals
X-squared = 140.2446, df = 9, p-value < 2.2e-16
> Box.test(residuals,lag=12,type="Ljung")
  Box-Ljung test
data: residuals
X-squared = 144.6815, df = 12, p-value < 2.2e-16

```

### Lampiran M : Markov Switching Dengan TSM Series Data Telkom

Data file is D:\saham.csv

Dependent Variable is Telkom  
3128 observations (2-3129) used for estimation  
with 1 pre-sample observations.

Estimation Method: Conditional ML (Time Domain)

Gaussian Likelihood

Markov-switching model with 2 regimes

Parameters for Regimes > 1 are differences from Regime 1.

ARIMA(1,0,0)

Strong convergence

iteration time: 16.65

Markov Transition Probabilities

	P(. 1)	P(. 2)
P(1.)	0.10782	0.044902
P(2.)	0.89218	0.95510

	Estimate	Std. Err.	t Ratio	p-Value
Logistic, t(1,1)	-2.11321	0.6323	-----	-----
Logistic, t(1,2)	-3.05733	0.4244	-----	-----

Non-switching parameters shown as Regime 1.

Regime 1:

Intercept	0.01552	0.00038	40.838	0
AR1	1.32619	0.22765	5.826	0
Error Variance <sup>(1/2)</sup>	0.0158	0.0005	-----	-----

Regime 2:

AR1	-1.23108	0.21998	-5.596	0
-----	----------	---------	--------	---

Log Likelihood = 8432.57

Schwarz Criterion = 8408.43

Hannan-Quinn Criterion = 8420.06

Akaike Criterion = 8426.57

Sum of Squares = 0.7508

R-Squared = 0.1786

R-Bar-Squared = 0.1773

Residual SD = 0.0155

Residual Skewness = 2.0719

Residual Kurtosis = 10.9398

Jarque-Bera Test = 10454.3 {0}

Box-Pierce (residuals): Q(11) = 112.008 {0}

Box-Pierce (squared residuals): Q(12) = 105.531 {0}



Dependent Variable is Telkom  
 3129 observations (1-3129) used for estimation.  
 Estimation Method: Conditional ML (Time Domain)  
 Gaussian Likelihood  
 Markov-switching model with 2 regimes  
 Parameters for Regimes > 1 are differences from Regime 1.  
 ARIMA(0,0,1)

Strong convergence  
 iteration time: 27.92

Markov Transition Probabilities

	P(.,1)	P(.,2)
P(1.)	0.99075	0.68634
P(2.)	0.0092519	0.31366

	Estimate	Std. Err.	t Ratio	p-Value
Logistic, t(1,1)	4.67363	0.2533	-----	-----
Logistic, t(1,2)	0.78308	0.3111	-----	-----

Non-switching parameters shown as Regime 1.

Regime 1:

Intercept	0.01583	0.00033	47.978	0
MA1	-0.12431	0.01869	-6.651	0
Error Variance^(1/2)	0.01579	0.0004	-----	-----

Regime 2:

MA1	-0.84802	0.02122	-39.963	0
-----	----------	---------	---------	---

Log Likelihood = 8442.77

Schwarz Criterion = 8418.63

Hannan-Quinn Criterion = 8430.26

Akaike Criterion = 8436.77

Sum of Squares = 0.7657

R-Squared = 0.1728

R-Bar-Squared = 0.1714

Residual SD = 0.0157

Residual Skewness = 1.7119

Residual Kurtosis = 7.714

Jarque-Bera Test = 4425.45 {0}

Box-Pierce (residuals): Q(11) = 183.843 {0}

Box-Pierce (squared residuals): Q(12) = 297.299 {0}

### Lampiran N : Uji White Noise Dan Normality Test Ljung-Box Series Telkom

```

> source("D:\\residualtelkommarkv1.csv")
Error in eval(expr, envir, enclos) : object 'Residuals' not found
> residuals=read.csv("D:\\residualtelkommarkv1.csv")
> residuals=read.csv("D:\\residualtelkommarkv1.csv",header=TRUE)
> ksnormTest(residuals)
Title:
One-sample Kolmogorov-Smirnov test
Test Results:
STATISTIC:
D: 0.4899
P VALUE:
Alternative Two-Sided: < 2.2e-16
Alternative Less: < 2.2e-16
Alternative Greater: < 2.2e-16
Description:
Tue Jul 02 16:00:09 2013 by user: Owner
Warning messages:
1: In ks.test(x, "pnorm", alternative = "two.sided") :
ties should not be present for the Kolmogorov-Smirnov test
2: In ks.test(x, "pnorm", alternative = "less") :
ties should not be present for the Kolmogorov-Smirnov test
3: In ks.test(x, "pnorm", alternative = "greater") :
ties should not be present for the Kolmogorov-Smirnov test
> Box.test(residuals,lag=3,type="Ljung")
Box-Ljung test
data: residuals
X-squared = 38.0533, df = 3, p-value = 2.754e-08
> Box.test(residuals,lag=6,type="Ljung")
Box-Ljung test
data: residuals
X-squared = 84.8664, df = 6, p-value = 3.331e-16
> Box.test(residuals,lag=9,type="Ljung")
Box-Ljung test
data: residuals
X-squared = 94.9467, df = 9, p-value < 2.2e-16
> Box.test(residuals,lag=12,type="Ljung")
Box-Ljung test
data: residuals
X-squared = 112.2683, df = 12, p-value < 2.2e-16

```

---

```

> source("D:\\residualtelkommarkv2.csv")
Error in eval(expr, envir, enclos) : object 'Residuals' not found
> residuals=read.csv("D:\\residualtelkommarkv2.csv",header=TRUE)
> ksnormTest(residuals)
Title:

```

One-sample Kolmogorov-Smirnov test  
 Test Results:  
 STATISTIC:  
 D: 0.4897  
 P VALUE:  
 Alternative Two-Sided:  $< 2.2e-16$   
 Alternative Less:  $< 2.2e-16$   
 Alternative Greater:  $< 2.2e-16$   
 Description:  
 Tue Jul 02 16:13:37 2013 by user: Owner  
 Warning messages:  
 1: In ks.test(x, "pnorm", alternative = "two.sided") :  
 ties should not be present for the Kolmogorov-Smirnov test  
 2: In ks.test(x, "pnorm", alternative = "less") :  
 ties should not be present for the Kolmogorov-Smirnov test  
 3: In ks.test(x, "pnorm", alternative = "greater") :  
 ties should not be present for the Kolmogorov-Smirnov test  
 > Box.test(residuals,lag=3,type="Ljung")  
 Box-Ljung test  
 data: residuals  
 X-squared = 78.3613, df = 3, p-value  $< 2.2e-16$   
 > Box.test(residuals,lag=6,type="Ljung")  
 Box-Ljung test  
 data: residuals  
 X-squared = 151.7931, df = 6, p-value  $< 2.2e-16$   
 > Box.test(residuals,lag=9,type="Ljung")  
 Box-Ljung test  
 data: residuals  
 X-squared = 171.0122, df = 9, p-value  $< 2.2e-16$   
 > Box.test(residuals,lag=12,type="Ljung")  
 Box-Ljung test  
 data: residuals  
 X-squared = 184.2247, df = 12, p-value  $< 2.2e-16$

## TENTANG PENULIS



Penulis lahir di Sidorejo Lampung pada tanggal 24 Februari 1981. Penulis merupakan anak pertama dari empat bersaudara dari pasangan Bapak Nyoman Putra dan Ibu Wayan Sasih, suami dari Putu Yusi Pramandari. Pendidikan formal yang pernah ditempuh antara lain SDN Sidorejo Lampung Timur, SMP Ganesha Metro Kodya Lampung, SMU Teladan Way Jepara Lampung Timur. Pada tahun 1999 penulis melanjutkan ke jenjang Strata Satu (S-1) di Universitas Islam Indonesia (UII) Jogjakarta pada jurusan Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam, lulus pada tahun 2003. Pada tahun 2008 penulis diterima PNS sebagai Dosen di Universitas Terbuka (UPBJJ-UT) Denpasar dan pada tahun 2011 penulis mendapatkan kesempatan melanjutkan pendidikan ke jenjang S-2 di Institut Teknologi Sepuluh Nopember Surabaya pada jurusan Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam. Harapannya semoga Tesis ini bermanfaat bagi pembaca, segala kritik dan saran yang membangun dapat dikirimkan melalui email : [isuwardika@ut.ac](mailto:isuwardika@ut.ac) / [igede\\_suwardika@yahoo.com](mailto:igede_suwardika@yahoo.com)