

**ESTIMASI PARAMETER MODEL
SEASONAL AUTOREGRESSIVE INTEGRATED MOVING AVERAGE
DENGAN METODE KALMAN FILTER**

SKRIPSI

**OLEH
DENY FATCHUR ROCHMAN
NIM. 14610082**



**JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2019**

**ESTIMASI PARAMETER MODEL
SEASONAL AUTOREGRESSIVE INTEGRATED MOVING AVERAGE
DENGAN METODE KALMAN FILTER**

SKRIPSI

**Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)**

**Oleh
Deny Fatchur Rochman
NIM. 14610082**

**JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2019**

**ESTIMASI PARAMETER MODEL
SEASONAL AUTOREGRESSIVE INTEGRATED MOVING AVERAGE
DENGAN METODE KALMAN FILTER**

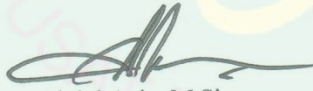
SKRIPSI

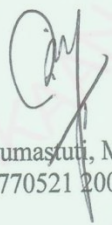
Oleh
Deny Fatchur Rochman
NIM. 14610082

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal 13 Mei 2019

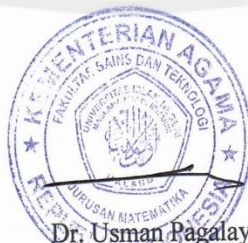
Pembimbing I,

Pembimbing II,


Abdul Aziz, M.Si
NIP. 19760318 200604 1 002


Ari Kusumastuti, M.Pd, M.Si
NIP. 19770521 200501 2 004

Mengetahui,
Ketua Jurusan Matematika



Dr. Usman Pagalay, M.Si
NIP. 19650414 200312 1 001

**ESTIMASI PARAMETER MODEL
SEASONAL AUTOREGRESSIVE INTEGRATED MOVING AVERAGE
DENGAN METODE KALMAN FILTER**

SKRIPSI

Oleh
Deny Fatchur Rochman
NIM. 14610082

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)

Tanggal 10 Juni 2019

Penguji Utama : Dr. Sri Harini, M.Si

Ketua Penguji : Angga Dwi Mulyanto, M.Si

Sekretaris Penguji : Abdul Aziz, M.Si

Anggota Penguji : Ari Kusumastuti, M.Pd, M.Si

Mengetahui,
Ketua Jurusan Matematika



Dr. Usman Pagalay, M.Si
NIP. 19650414 200312 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Deny Fatchur Rochman

NIM : 14610082

Jurusan : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Estimasi Parameter Model *Seasonal Autoregressive Integrated Moving Average* dengan Metode *Kalman Filter*

menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar rujukan. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 13 Mei 2019
Yang membuat pernyataan,



Deny Fatchur Rochman
NIM. 14610082

MOTO

“Setiap hembusan nafas yang diberikan Allah SWT kepada kita bukan hanya
berkah, tapi juga tanggung jawab”



PERSEMBAHAN

Skripsi ini penulis persembahkan untuk:

Ayahanda Zainul Arifin dan Ibunda Siti Nurul Hidayah yang selalu memberikan bimbingan dan do'anya kepada penulis untuk kesuksesan di dunia maupun akhirat. Serta Evy Nur Faizah selaku adik penulis yang senantiasa memberikan support selama ini.



KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji bagi Allah Swt atas rahmat, taufik serta hidayah-Nya, sehingga penulis mampu menyelesaikan penyusunan skripsi ini sebagai salah satu syarat untuk memperoleh gelar sarjana dalam bidang matematika di Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Dalam proses penyusunan skripsi ini, penulis banyak mendapat bimbingan dan arahan dari berbagai pihak. Untuk itu ucapan terima kasih yang sebesar-besarnya dan penghargaan yang setinggi-tingginya penulis sampaikan terutama kepada:

1. Prof. Dr. H. Abd. Haris, M.Ag, selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si, selaku dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Usman Pagalay, M.Si, selaku ketua Jurusan Matematika, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Abdul Aziz, M.Si, selaku dosen pembimbing I yang telah banyak memberikan arahan, nasihat, motivasi, dan berbagi pengalaman yang berharga kepada penulis.
5. Ari Kusumastuti, M.Pd, M.Si, selaku dosen pembimbing II yang telah banyak memberikan arahan, ilmu, dan motivasi kepada penulis.
6. Seluruh dosen Jurusan Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah ikhlas

dan sabar dalam mendidik, membimbing, dan memberikan ilmu kepada penulis.

7. Ayah dan Ibu yang selalu memberikan doa, semangat, nasihat, dan motivasi kepada penulis.
8. Teman-teman di Jurusan Matematika angkatan 2014, khususnya kelas C yang berjuang bersama-sama untuk meraih mimpi.
9. Semua pihak yang ikut membantu dalam menyelesaikan skripsi ini.

Akhirnya penulis berharap semoga skripsi ini bermanfaat bagi penulis dan bagi pembaca.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 13 Mei 2019

Penulis

DAFTAR ISI

HALAMAN JUDUL

HALAMAN PENGAJUAN

HALAMAN PERSETUJUAN

HALAMAN PENGESAHAN

HALAMAN PERNYATAAN KEASLIAN TULISAN

HALAMAN MOTO

HALAMAN PERSEMBAHAN

KATA PENGANTAR..... viii

DAFTAR ISI..... x

DAFTAR TABEL xii

DAFTAR GAMBAR..... xiii

DAFTAR SIMBOL xiv

ABSTRAK xvi

ABSTRACT xvii

ملخص..... xviii

BAB I PENDAHULUAN

1.1	Latar Belakang	1
1.2	Rumusan Masalah.....	4
1.3	Tujuan Penelitian	4
1.4	Batasan Masalah	4
1.5	Manfaat Penelitian	4
1.6	Sistematika Penulisan	5

BAB II KAJIAN TEORI

2.1	Analisis Deret Waktu Berkala	7
-----	------------------------------------	---

2.2	Fungsi Autokorelasi	8
2.3	Fungsi Autokorelasi Parsial	10
2.4	<i>White Noise</i>	14
2.5	Stasioneritas Data	16
2.5.1	Differencing	17
2.5.2	Transformasi	19
2.6	Uji Stasioneritas	22
2.7	<i>Time Series</i>	24
2.7.1	Model <i>Time Series</i> Stasioner	24
2.7.2	Model <i>Time Series</i> Nonstasioner	28
2.8	Identifikasi Model	30
2.9	Estimasi Parameter	31
2.9.1	Persamaan <i>Kalman Filter</i>	31
2.9.2	Penerapan <i>Kalman Filter</i> dalam Estimasi Parameter Model SARIMA	32
2.10	Hasil Penelitian Sebelumnya	33
2.11	Kajian Islam Tentang Estimasi	34
 BAB III METODOLOGI PENELITIAN		
3.1	Pendekatan Penelitian	37
3.2	Jenis dan Sumber Data	37
3.3	Variabel Penelitian	37
3.4	Analisis Data	37
 BAB IV PEMBAHASAN		
4.1	Implementasi Metode <i>Kalman Filter</i> pada Model ARIMA $(p, d, q)(P, D, Q)^S$	39
4.1.1	Identifikasi Data	39
4.1.2	Pengujian Asumsi <i>White Noise</i>	44
4.1.3	Estimasi Parameter Model ARIMA $(1,0,1)^{12}$	45
4.1.4	Estimasi Parameter Model ARIMA $(1,0,1)^{12}$ dengan Metode <i>Kalman Filter</i>	46
4.2	Pengaplikasian Konsep Pendugaan dalam Islam	52
 BAB V PENUTUP		
5.1	Kesimpulan	53
5.2	Saran	53
 DAFTAR RUJUKAN		54
 LAMPIRAN		

DAFTAR TABEL

Tabel 2.1	Transformasi <i>Box-Cox</i>	20
Tabel 4.1	Nilai Koefisien ACF Data Asli	41
Tabel 4.2	Ringkasan Uji Ljung-Box	45
Tabel 4.3	Nilai <i>error</i> pada setiap nilai awal	50
Tabel 4.4	Hasil estimasi parameter model ARIMA (1,0,1) ¹² menggunakan <i>Kalman Filter</i>	50

DAFTAR GAMBAR

Gambar 2.1	Contoh Plot Data Stasioner dalam Rata-rata dan Variansi	16
Gambar 2.2	Contoh Plot Data Tidak Stasioner dalam Rata-rata	17
Gambar 2.3	Contoh Pola Data Stasioner dalam Variansi	17
Gambar 4.1	Plot Data Curah Hujan.....	39
Gambar 4.2	Plot ACF Data Asli.....	41
Gambar 4.3	Plot PACF Data Asli	42
Gambar 4.4	Hasil Estimasi Parameter Model ARIMA (1,0,0)(1,0,1) ¹²	43
Gambar 4.5	Hasil Estimasi Parameter Model ARIMA (0,0,1)(1,0,1) ¹²	43
Gambar 4.6	Hasil Estimasi Parameter Model ARIMA (1,0,1) ¹²	44

DAFTAR SIMBOL

Simbol	Nama	Jenis	Keterangan
x		Skalar	Variabel x
Y		Skalar	Variabel y
x_i		Skalar	Data pengamatan x ke- i , $i = 1, 2, \dots, n$
y_i		Skalar	Data pengamatan y ke- i , $i = 1, 2, \dots, n$
n		Skalar	Banyaknya data
S_x		Skalar	Nilai simpangan baku x
ρ	ρ	Skalar	Nilai koefisien korelasi
Y_{t+k}		Skalar	Variabel Y pada waktu ke- $(t + k)$
γ_k	γ - k	Skalar	Nilai kovariansi γ pada lag ke- k
ρ_k	ρ - k	Skalar	Nilai koefisien autokorelasi pada lag ke- k
t		Skalar	Waktu pengamatan ke- t , $t = 1, 2, \dots, k$
ϕ_{ki}	ϕ - ki	Skalar	Nilai koefisien autokorelasi parsial ke- i
Y		Matriks	Variabel terikat
X		Matriks	Variabel bebas
B		Operator	Operator backward shift
a		Vektor	Error

β	<i>beta</i>	Vektor	Vektor parameter konstanta regresi
Y_t		Vektor	Data Y pada waktu ke t
μ	<i>mu</i>	Vektor	Rata-rata dari Y_t
ϕ	<i>Phi</i>	Vektor	Parameter koefisien <i>Autoregressive</i>
Φ	<i>Phi</i>	Vektor	Parameter koefisien <i>Autoregressive</i> musiman
λ	<i>lambda</i>	Vektor	Parameter transformasi <i>Box-Cox</i>
θ_j	<i>theta</i>	Vektor	Parameter koefisien <i>moving average</i>
Θ	<i>theta</i>	Vektor	Parameter koefisien <i>moving average</i> musiman
Z_t		Skalar	Selisih dari nilai variabel Y_t dengan μ
x_k		Vektor	Variabel keadaan
u_k		Vektor	Vektor masukan deterministik
z_t		Vektor	Vektor pengukuran
A		Matriks	Matriks konstan
F		Matriks	Matriks konstan
G		Matriks	Matriks konstan
H		Matriks	Matriks konstan
Q		Matriks	Matriks Kovarian
P		Matriks	Kovarian <i>error</i>

ABSTRAK

Rochman, Deny Fatchur. 2019. **Estimasi Parameter Model *Seasonal Autoregressive Integrated Moving Average* dengan Metode *Kalman Filter***. Skripsi. Jurusan Matematika Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Abdul Aziz, M.Si. (II) Ari Kusumastuti, M.Pd., M.Si.

Kata kunci: Estimasi, *Seasonal Autoregressive Integrated Moving Average*, *Kalman Filter*

Seasonal Autoregressive Integrated Moving Average (*Seasonal ARIMA*) merupakan model peramalan data yang mengandung unsur musiman. *Kalman Filter* merupakan metode estimasi yang optimal. Komponen dasar dari metode *Kalman Filter* adalah persamaan pengukuran dan persamaan transisi dengan menggunakan data pengukuran untuk memperbaiki hasil estimasi. Tujuan dari penelitian ini adalah mengetahui proses dan hasil estimasi parameter model *ARIMA* (1,0,1)¹² menggunakan metode *Kalman Filter* pada data jumlah curah hujan di Kota Semarang mulai tahun 2005 sampai dengan 2011.

Estimasi parameter model *ARIMA* (1,0,1)¹² menggunakan metode *Kalman Filter* terdiri dari beberapa tahap yaitu identifikasi data curah hujan, uji kestasioneran data, estimasi parameter dengan metode *Kalman Filter*. Hasil dari penelitian ini menunjukkan bahwa model *ARIMA* (1,0,1)¹² dengan metode *Kalman Filter* merupakan model yang sesuai ketika diterapkan pada data jumlah curah hujan di Kota Semarang. Bentuk estimasi parameternya adalah sebagai berikut

$$Y_t = 0.1Y_{t-12} - 0.1a_{t-12} + a_t$$

ABSTRACT

Rochman, Deny Fatchur. 2019. **Parameter Estimation of Seasonal Autoregressive Integrated Moving Average with Kalman Filter Method**. Thesis. Department of Mathematics, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Advisors: (I) Abdul Aziz, M.Si. (II) Ari Kusumastuti, M.Pd., M.Si.

Keyword: Estimation, Seasonal Autoregressive Integrated Moving Average, Kalman Filter

Seasonal Autoregressive Integrated Moving Average (Seasonal ARIMA) is a data forecasting model that contains seasonal elements. Kalman Filter is the optimal estimation method. The basic components of the Kalman Filter method are measurement equations and transition equations by using measurement data to improve estimation results. The purpose of this study is to know the process and the results of the ARIMA (1,0,1)¹² parameter estimation model using the Kalman Filter method on rainfall amount data in Semarang City from 2005 to 2011.

The parameter estimation of the ARIMA (1,0,1)¹² model using the Kalman Filter method consists of several stages, namely identification of rainfall data, data stationary test and parameter estimation using the Kalman Filter method. The results of this study indicate that ARIMA (1,0,1)¹² model with the Kalman Filter method is an appropriate model when applied to rainfall amount data in Semarang City. The form of parameter estimation is as follows

$$Y_t = 0.1Y_{t-12} - 0.1a_{t-12} + a_t$$

ملخص

الرحمن ، دني فتح. 2019. معيار المعلمة نموذج *Seasonal Autoregressive Integrated Moving Average* باستخدام طريقة *Kalman Filter*. البحث الجامعي .شعبة الرياضيات، كلية العلوم والتكنول وجيا، الجامعة الإسلامية احكومية مولانا مالك إبراهيم ماناج .المشرف: (1) عبد العزيز، الماجستير. (2) آري كوسوماستوتي، الماجستير

الكلمة الرئيسية: تقدير، تقدير، *Seasonal Autoregressive Integrated Moving Average. Kalman Filter*

المتوسط المتحرك للانحدار التلقائي الموسمي (*Seasonal ARIMA*) هو نموذج تنبؤ للبيانات يحتوي على عناصر موسمية. *Kalman Filter* هي طريقة التقدير المثلى. المكونات الأساسية لطريقة *Kalman Filter* هي معادلات القياس ومعادلات الانتقال باستخدام بيانات القياس لتحسين نتائج التقدير. الغرض من هذه الدراسة هو معرفة عملية ونتائج $ARIMA(1,0,1)^{12}$ نموذج تقدير المعلمة باستخدام طريقة *Kalman Filter* على بيانات كمية هطول الأمطار في مدينة سمارنج من 2005 إلى 2011.

يتكون تقدير المعلمة لنموذج $ARIMA(1,0,1)^{12}$ باستخدام طريقة *Kalman Filter* من عدة مراحل ، وهي تحديد بيانات هطول الأمطار ، واختبار محطة البيانات ، وتقدير المعلمة باستخدام طريقة *Kalman Filter*. تشير نتائج هذه الدراسة إلى أن $ARIMA(1,0,1)^{12}$ بطريقة *Kalman Filter* هو نموذج مناسب عند تطبيقه على بيانات كمية الأمطار في مدينة سمارنج. تقدير المعلمة هو

$$Y_t = 0.1Y_{t-12} - 0.1a_{t-12} + a_t$$

BAB I

PENDAHULUAN

1.1 Latar Belakang

Statistika adalah suatu ilmu yang mempelajari cara pengumpulan, pengolahan, penyajian data, penarikan kesimpulan serta pembuatan keputusan yang cukup beralasan berdasarkan data dan analisis yang dilakukan. Salah satu penerapan statistika yang biasa digunakan adalah pemodelan deret berkala (*time series*). Data *time series* merupakan sekumpulan data hasil pengamatan atau pencatatan historis dan berkala yang menggambarkan secara kronologis suatu karakteristik populasi. *Time series* adalah suatu rangkaian atau seri dari nilai-nilai suatu variabel yang dicatat dalam jangka waktu yang berurutan (Atmaja, 1997).

Analisis *time series* merupakan suatu metode analisis data yang ditujukan untuk melakukan suatu estimasi maupun peramalan pada masa yang akan datang. Dalam analisis *time series* akan diketahui bagaimana proses suatu estimasi dan hasil peramalan dapat diperoleh dengan baik. Untuk itu dalam analisis ini dibutuhkan berbagai macam informasi atau data yang cukup banyak dan diamati dalam periode waktu yang relatif cukup panjang (Atmaja, 1997). Salah satu model yang sering digunakan dalam pemodelan runtun waktu untuk peramalan adalah *Autoregressive Integrated Moving Average* (ARIMA).

Model ARIMA merupakan salah satu analisis *time series* yang didasarkan pada perilaku variabel yang diamati. Model ARIMA sama sekali mengabaikan variabel bebas karena model dapat bekerja dengan baik apabila data *time series*

yang digunakan bersifat terikat. ARIMA merupakan konsep tentang stasioner dan non stasioner, konsep autokovariansi, autokorelasi, autokorelasi parsial dan lain sebagainya. Agar model ARIMA menghasilkan ramalan yang optimal, maka model tersebut harus memenuhi asumsi-asumsi statistika (Makridakis dkk, 1999).

Model ARIMA baik digunakan untuk peramalan pada jangka waktu yang pendek dan kurang tepat jika digunakan pada data yang memiliki pola dengan jangka waktu yang panjang. Oleh karena itu, ada perluasan dari model ARIMA yang biasa dikenal dengan model SARIMA (*Seasonal ARIMA*). SARIMA merupakan model yang dapat digunakan untuk peramalan data yang mengandung unsur musiman atau memiliki jangka waktu yang panjang (Makridakis dkk, 1999).

Kalman Filter adalah sebuah metode bagian dari *state space* (ruang keadaan) yang dapat diterapkan dalam model prakiraan statistik. Metode ini menggunakan teknik rekursif dalam mengintegrasikan data pengamatan terbaru ke model untuk mengoreksi prediksi sebelumnya dan melakukan prediksi selanjutnya secara optimal berdasarkan informasi data di masa lalu maupun berdasarkan informasi data saat ini (Wei, 2006).

Menurut penelitian yang dilakukan oleh Kurniawan dkk (2014) menyimpulkan bahwa penggunaan algoritma *Kalman Filter* mempunyai pengaruh baik terhadap perbaikan hasil prediksi terbukti dari nilai *Mean Absolute Percentage Error* (MAPE) pada data suhu udara dengan menggunakan *Kalman Filter* mempunyai nilai lebih kecil dari ARIMA. Hal yang sama juga berlaku pada data kecepatan angin dimana penggunaan algoritma *Kalman Filter* mempunyai

pengaruh baik terhadap perbaikan hasil prediksi terbukti dari nilai MAPE dengan menggunakan *Kalman Filter* mempunyai nilai lebih kecil dari ARIMA.

Sedangkan Pamungkas dkk (2017) menyimpulkan bahwa *Kalman Filter* dapat meminimalkan nilai kesalahan model ARIMA. Hal ini diketahui dari nilai rata-rata MAPE ARIMA yang diestimasi menggunakan *Kalman Filter* lebih kecil daripada ARIMA. Sehingga model ARIMA yang diestimasi menggunakan *Kalman Filter* lebih akurat dibandingkan sebelum diestimasi menggunakan *Kalman Filter*.

Dalam Al-quran, estimasi telah disinggung dalam surat Al-Jasiyah ayat 24 yang artinya:

“Dan mereka berkata: "Kehidupan ini tidak lain hanyalah kehidupan di dunia saja, kita mati dan kita hidup dan tidak ada yang akan membinasakan kita selain masa", dan mereka sekali-kali tidak mempunyai pengetahuan tentang itu, mereka tidak lain hanyalah menduga-duga saja.”

Dari ayat tersebut memberikan penjelasan bahwa konteks estimasi terletak pada hubungan antara kebutuhan manusia akan ilmu pengetahuan dengan keterbatasan manusia dalam memperoleh ilmu pengetahuan itu sendiri. Suatu indikasi bahwa dengan adanya keterbatasan manusia, manusia dituntut untuk melakukan estimasi (pendugaan) terhadap segala sesuatunya sebagai fondasi fundamental dalam melakukan pencarian terhadap kebenaran ilmu pengetahuan.

Berdasarkan beberapa uraian di atas, maka penulis ingin mengembangkan penelitian yang telah dilakukan dengan memperluas model ARIMA menjadi SARIMA. Sehingga mengangkat tema dari tulisan ini dengan judul “Estimasi Parameter Model *Seasonal Autoregressive Integrated Moving Average* dengan Metode *Kalman Filter*”.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam penelitian ini yaitu bagaimana estimasi parameter model SARIMA menggunakan metode *Kalman Filter* dalam peramalan data jumlah Curah Hujan di Kota Semarang pada tahun 2005 sampai 2011?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah disebutkan, maka tujuan penelitian ini adalah mengetahui hasil estimasi parameter model SARIMA menggunakan metode *Kalman Filter* dalam peramalan data jumlah Curah Hujan di Kota Semarang pada tahun 2005 sampai 2011.

1.4 Batasan Masalah

Untuk membatasi masalah agar sesuai dengan yang dimaksudkan dan tidak menimbulkan permasalahan yang baru, maka peneliti memberikan batasan masalah sebagai berikut:

1. Estimasi yang dilakukan menggunakan data jumlah Curah Hujan di Kota Semarang pada tahun 2005 sampai 2011 yang di peroleh dari Badan Metereologi Klimatologi dan Geofisika (BMKG) Stasiun Klimatologi Semarang.
2. Model *time series* yang dikaji model ARIMA $(p, d, q)(P, D, Q)^S$

1.5 Manfaat Penelitian

Penelitian ini diharapkan memberikan tambahan wawasan dan pengetahuan mengenai estimasi parameter model SARIMA dengan metode *Kalman Filter*.

1.6 Sistematika Penulisan

Untuk memudahkan pemahaman akan penelitian ini secara menyeluruh, maka digunakan sistematika penulisan yang terdiri dari lima bab, yaitu:

Bab I Pendahuluan

Pada bab ini akan diuraikan tentang latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan.

Bab II Kajian Pustaka

Pada bab ini akan diuraikan tentang kajian teori yang mendasari pembahasan serta teori yang berhubungan dengan penelitian seperti *time series*, definisi estimasi, definisi data *time series*, definisi model SARIMA dan definisi estimasi parameter model dengan metode *Kalman Filter*.

Bab III Metode Penelitian

Pada bab ini diuraikan tentang langkah-langkah metode penelitian yang meliputi pendekatan penelitian, jenis dan sumber data, variabel penelitian dan prosedur pembentukan model peramalan.

Bab IV Pembahasan

Pada bab ini penulis menjelaskan cara mengestimasi parameter model SARIMA dengan menggunakan metode *Kalman Filter*, implementasi metode *Kalman Filter* dan kajian agama Islam mengenai model SARIMA dan metode *Kalman Filter*.

Bab V Penutup

Pada bab ini diuraikan tentang hasil pokok dan kesimpulan dari analisis terhadap data yang diolah dan berisi saran untuk pembaca dan peneliti selanjutnya.



BAB II

KAJIAN TEORI

2.1 Analisis Deret Waktu Berkala

Analisis deret waktu berkala (*time series*) adalah pendugaan suatu nilai pada masa yang akan datang dengan menggunakan nilai masa lalu dari suatu variabel atau kesalahan masa lalu. Analisis *time series* bertujuan untuk menemukan pola dalam deret data historis dan menerapkan pola tersebut ke masa yang akan datang (Makridakis dkk, 1999). Selain itu, analisis *time series* juga bisa digunakan untuk pendugaan dari data beberapa periode selanjutnya yang berguna untuk menyusun suatu perencanaan.

Salah satu langkah penting dalam memilih metode pendugaan adalah mempertimbangkan pola data sehingga metode pendugaan yang sesuai dengan data tersebut dapat bermanfaat. Berikut ini adalah pola-pola deret berkala yang telah dikenal (Hanke dan Wichern, 2005):

1. Pola Data Horizontal

Pola horizontal terjadi ketika nilai-nilai data berfluktuasi di sekitar nilai rata-rata yang konstan. Penjualan yang tidak naik ataupun turun secara signifikan dalam suatu rentang waktu tertentu.

2. Pola Data Trend

Pola data trend didefinisikan sebagai kenaikan atau penurunan pada deret waktu dalam selang periode waktu tertentu.

3. Pola Data Musiman

Pola data musiman terjadi ketika dipengaruhi faktor musiman yang signifikan sehingga data naik dan turun dengan pola yang berulang dari satu periode ke periode berikutnya.

4. Pola Data Siklis

Pola data siklis didefinisikan sebagai fluktuasi data berbentuk gelombang sepanjang periode yang tidak menentu.

2.2 Fungsi Autokorelasi

Autokorelasi merupakan korelasi atau hubungan antar data pada pengamatan data *time series*. Pada korelasi, hubungan yang terjalin merupakan dua variabel yang berbeda pada waktu yang sama, sedangkan pada autokorelasi, hubungan yang terjalin merupakan dua variabel yang sama dalam rentang waktu yang berbeda (Firdaus, 2004).

Menurut Makridakis, dkk (1999) rata-rata dan variansi dari suatu data deret berkala mungkin tidak bermanfaat apabila deret tersebut tidak stasioner, akan tetapi nilai minimum dan maksimum dapat digunakan untuk tujuan plotting. Bagaimanapun statistik kunci di dalam analisis deret berkala adalah koefisien autokorelasi (atau korelasi deret berkala dengan deret berkala itu sendiri dengan selisih waktu (*lag*) 0, 1, 2 periode atau lebih).

Koefisien korelasi sederhana antara Y_t dengan Y_{t+1} dapat dicari dengan menggunakan (2.8), dinyatakan dibawah ini (Makridakis dkk, 1999):

$$\begin{aligned}
 r_{Y_t Y_{t+1}} &= \frac{Cov_{Y_t Y_{t+1}}}{\sigma_{Y_t} \cdot \sigma_{Y_t Y_{t+k}}} \\
 &= \frac{\sum_{t=1}^n (Y_t - \bar{Y}_t)(Y_{t+1} - \bar{Y}_{t+1})}{\sqrt{\sum_{t=1}^n (Y_t - \bar{Y}_t)^2} \sqrt{\sum_{t=1}^n (Y_{t+1} - \bar{Y}_{t+1})^2}}
 \end{aligned} \tag{2.1}$$

Data Y_t diasumsikan stasioner (baik rata-rata maupun variansinya). Jadi, kedua rata-rata, Y_t dan Y_{t+1} dapat diasumsikan bernilai sama (dan kita dapat membuang subskrip dengan menggunakan $\bar{Y} = \bar{Y}_t = \bar{Y}_{t+1}$) dan dua nilai variansi dapat diukur satu kali saja dengan menggunakan seluruh data Y_t yang diketahui. dengan menggunakan asumsi-asumsi penyederhana ini, maka persamaan (2.1) menjadi:

$$r_{Y_t Y_{t+1}} = \frac{\sum_{t=1}^n (Y_t - \bar{Y})(Y_{t+1} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2} \tag{2.2}$$

Pada analisis *time series*, γ_k merupakan fungsi autokovariansi dan ρ_k merupakan *Autocorrelation Function* (ACF) karena menunjukkan keeratan antara Y_t dan Y_{t+k} dari proses yang sama namun dengan selang waktu yang berbeda. Dalam proses stasioner Y_t dengan rata-rata $E(Y_t) = \mu$, variansi $Var(Y_t) = E(Y_t - \mu)^2 = \sigma^2$ yang konstan, dan kovariansi $Cov(Y_t, Y_{t+k})$ yang berfungsi hanya pada perbedaan waktu $|t - s|$. Sehingga dalam hal ini, fungsi autokovariansi antara Y_t dan Y_{t+k} dapat ditulis sebagai berikut (Wei, 2006):

$$\begin{aligned}
\gamma_k &= \text{Cov}(Y_t, Y_{t+k}) \\
&= E[(Y_t - \mu)(Y_{t+k} - \mu)] \\
&= E[(Y_t Y_{t+k} - Y_t \mu - \mu Y_{t+k} + \mu\mu)] \\
&= E[(Y_t Y_{t+k} - Y_t \mu - Y_{t+k} \mu + \mu\mu)] \\
&= E[Y_t Y_{t+k}] - E[Y_t \mu] - E[Y_{t+k} \mu] + E[\mu\mu] \\
&= E[Y_t Y_{t+k}] - \mu E[Y_t] - \mu E[Y_{t+k}] + \mu\mu \\
&= E[Y_t Y_{t+k}] - \mu\mu - \mu\mu + \mu\mu \\
&= E[Y_t Y_{t+k}] - \mu\mu
\end{aligned} \tag{2.3}$$

Fungsi autokorelasi antara Y_t dan Y_{t+k} dapat ditulis sebagai berikut:

$$\begin{aligned}
\rho_k &= \frac{\text{Cov}(Y_t, Y_{t+k})}{\sqrt{\text{Var}(Y_t)} \sqrt{\text{Var}(Y_{t+k})}} \\
&= \frac{\gamma_k}{\sqrt{\sigma^2} \sqrt{\sigma^2}} \\
&= \frac{\gamma_k}{\sigma^2} \\
&= \frac{\gamma_k}{\gamma_0}
\end{aligned} \tag{2.4}$$

dengan:

$$\text{Var}(Y_t) = \text{Var}(Y_{t+k}) = \sigma^2 = \gamma_0$$

γ_k : nilai kovariansi γ pada lag $k, k = 1, 2, 3, \dots$

ρ_k : nilai autokorelasi pada lag k

t : observasi ke- t .

2.3 Fungsi Autokorelasi Parsial

Autokorelasi parsial digunakan untuk mengukur tingkat keeratan antara Y_t dan Y_{t+k} , apabila pengaruh dari *time lag* 1, 2, 3, . . . , dan seterusnya sampai $k-1$ dianggap terpisah (Makridakis dkk, 1999). Ada beberapa prosedur untuk menentukan bentuk PACF yang salah satunya akan dijelaskan sebagai berikut.

Autokorelasi parsial dapat diturunkan sebagai berikut, dengan variabel *dependent* Y_{t+k} dari proses stasioner rata-rata nol yang diregresikan dengan sejumlah k variabel $Y_{t+k-1}, Y_{t+k-2}, \dots, Y_t$, maka (Wei, 2006):

$$Y_{t+k} = \phi_{k1} Y_{t+k-1} + \phi_{k2} Y_{t+k-2} + \dots + \phi_{kk} Y_{t+k-k} + a_{t+k} \quad (2.5)$$

Dengan ϕ_{ki} merupakan parameter regresi dan a_{t+k} adalah nilai *error* dengan rata-rata 0, dan tidak berkorelasi dengan Y_{t+k-j} untuk $j = 1, 2, \dots, k$. Langkah pertama yang dilakukan adalah mengalikan persamaan (2.5) dengan Y_{t+k-j} pada kedua ruas sehingga diperoleh:

$$Y_{t+k} Y_{t+k-j} = \phi_{k1} Y_{t+k-1} Y_{t+k-j} + \phi_{k2} Y_{t+k-2} Y_{t+k-j} + \dots + \phi_{kk} Y_{t+k-k} Y_{t+k-j} + a_{t+k} Y_{t+k-j} \quad (2.6)$$

Selanjutnya, nilai ekspektasi dari persamaan (2.6) adalah

$$E[Y_{t+k} Y_{t+k-j}] = \phi_{k1} E[Y_{t+k-1} Y_{t+k-j}] + \phi_{k2} E[Y_{t+k-2} Y_{t+k-j}] + \dots + \phi_{kk} E[Y_{t+k-k} Y_{t+k-j}] + E[a_{t+k} Y_{t+k-j}] \quad (2.7)$$

Misalkan nilai $E[Y_{t+k} Y_{t+k-j}] = \gamma_j$, $j = 0, 1, 2, \dots, k$ dan karena $E[a_{t+k} Y_{t+k-j}] = 0$

sehingga diperoleh

$$\gamma_j = \phi_{k1} \gamma_{j-1} + \phi_{k2} \gamma_{j-2} + \dots + \phi_{kk} \gamma_{j-k} \quad (2.8)$$

dan persamaan (2.8) dibagi dengan $E[Y_{t+k}] = \gamma_0$

$$\frac{\gamma_j}{\gamma_0} = \phi_{k1} \frac{\gamma_{j-1}}{\gamma_0} + \phi_{k2} \frac{\gamma_{j-2}}{\gamma_0} + \dots + \phi_{kk} \frac{\gamma_{j-k}}{\gamma_0} \quad (2.9)$$

atau disederhanakan menjadi:

$$\rho_j = \phi_{k1} \rho_{j-1} + \phi_{k2} \rho_{j-2} + \dots + \phi_{kk} \rho_{j-k} \quad (2.10)$$

Untuk $j = 1, 2, \dots, k$, diperoleh sistem persamaan berikut:

$$\begin{aligned}
\rho_1 &= \phi_{k1}\rho_0 + \phi_{k2}\rho_1 + \dots + \phi_{kk}\rho_{j-1} \\
\rho_2 &= \phi_{k1}\rho_1 + \phi_{k2}\rho_0 + \dots + \phi_{kk}\rho_{j-2} \\
&\vdots \\
\rho_k &= \phi_{k1}\rho_{k-1} + \phi_{k2}\rho_{k-2} + \dots + \phi_{kk}\rho_0
\end{aligned} \tag{2.11}$$

Dengan menggunakan aturan Cramer, berturut-turut untuk $k = 1, 2, \dots$ diperoleh (Wei, 2006):

1. Untuk *lag* pertama ($k = 1$) diperoleh persamaan sebagai berikut:

$\rho_1 = \phi_{11}\rho_0$, karena $\rho_0 = \frac{y_0}{y_0} = 1$ sehingga $\phi_{11} = \rho_1$, yang berarti bahwa nilai fungsi autokorelasi parsial pada *lag* pertama akan sama dengan koefisien *lag* pertama.

2. Untuk *lag* kedua ($k = 2$) diperoleh persamaan sebagai berikut:

$$\begin{aligned}
\rho_1 &= \phi_{21}\rho_0 + \phi_{22}\rho_1 \\
\rho_2 &= \phi_{21}\rho_1 + \phi_{22}\rho_0
\end{aligned} \tag{2.12}$$

Persamaan (2.12) jika ditulis dalam matriks akan menjadi

$$\begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} = \begin{bmatrix} \rho_0 & \rho_1 \\ \rho_1 & \rho_0 \end{bmatrix} \begin{bmatrix} \phi_{21} \\ \phi_{22} \end{bmatrix}$$

Misalkan $A = \begin{bmatrix} \rho_0 & \rho_1 \\ \rho_1 & \rho_0 \end{bmatrix}$, $A_2 = \begin{bmatrix} \rho_0 & \rho_1 \\ \rho_1 & \rho_2 \end{bmatrix}$, dan dengan menggunakan

aturan Cramer diperoleh

$$\phi_{22} = \frac{\det(A_2)}{\det(A)} = \frac{\begin{vmatrix} 1 & \rho_1 \\ \rho_1 & \rho_2 \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{vmatrix}} \tag{2.13}$$

3. Untuk *lag* ketiga ($k = 3$) diperoleh persamaan sebagai berikut:

$$\begin{aligned}
\rho_1 &= \phi_{31}\rho_0 + \phi_{32}\rho_1 + \phi_{33}\rho_2 \\
\rho_2 &= \phi_{31}\rho_1 + \phi_{32}\rho_0 + \phi_{33}\rho_1 \\
\rho_3 &= \phi_{31}\rho_2 + \phi_{32}\rho_1 + \phi_{33}\rho_0
\end{aligned} \tag{2.14}$$

Persamaan (2.14) jika dinyatakan dalam bentuk matriks menjadi

$$\begin{bmatrix} \rho_0 & \rho_1 & \rho_2 \\ \rho_1 & \rho_0 & \rho_1 \\ \rho_2 & \rho_1 & \rho_0 \end{bmatrix} \begin{bmatrix} \phi_{31} \\ \phi_{32} \\ \phi_{33} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{bmatrix} \tag{2.15}$$

Dengan $A = \begin{bmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_1 \\ \rho_2 & \rho_1 & 1 \end{bmatrix}$, dan $A_3 = \begin{bmatrix} 1 & \rho_1 & \rho_1 \\ \rho_1 & 1 & \rho_2 \\ \rho_2 & \rho_1 & \rho_3 \end{bmatrix}$, menggunakan

aturan Cramer diperoleh koefisien autokorelasi:

$$\phi_{33} = \frac{\det(A_3)}{\det(A)} = \frac{\begin{vmatrix} 1 & \rho_1 & \rho_1 \\ \rho_1 & 1 & \rho_2 \\ \rho_2 & \rho_1 & \rho_3 \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_1 \\ \rho_2 & \rho_1 & 1 \end{vmatrix}}$$

4. Untuk lag ke- k dan $j = 1, 2, 3, \dots, k$ peroleh persamaan diperoleh:

$$\begin{aligned}
\rho_1 &= \phi_{k1}\rho_0 + \phi_{k2}\rho_1 + \dots + \phi_{kk}\rho_{j-1} \\
\rho_2 &= \phi_{k2}\rho_1 + \phi_{k2}\rho_0 + \dots + \phi_{kk}\rho_{j-2} \\
&\vdots \\
\rho_k &= \phi_{k1}\rho_{k-1} + \phi_{k2}\rho_{k-2} + \dots + \phi_{kk}\rho_0
\end{aligned} \tag{2.16}$$

$$\begin{bmatrix} \rho_0 & \rho_1 & \rho_2 & \cdots & \rho_{k-1} \\ \rho_1 & \rho_0 & \rho_1 & \cdots & \rho_{k-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \cdots & \rho_0 \end{bmatrix} \begin{bmatrix} \phi_{k1} \\ \phi_{k2} \\ \vdots \\ \phi_{kk} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_k \end{bmatrix}$$

dengan

$$A_k = \begin{bmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{k-1} \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \cdots & \rho_k \end{bmatrix}$$

diperoleh koefisien autokorelasi parsial k hasilnya adalah :

$$\phi_{kk} = \frac{\det(A_k)}{\det(A)} = \frac{\begin{vmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{k-2} & \rho_{k-1} \\ \rho_2 & 1 & \rho_1 & \cdots & \rho_{k-3} & \rho_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \cdots & \rho_1 & \rho_k \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{k-2} & \rho_{k-1} \\ \rho_2 & 1 & \rho_1 & \cdots & \rho_{k-3} & \rho_{k-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \cdots & \rho_1 & 1 \end{vmatrix}} \quad (2.17)$$

Karena ϕ_{kk} merupakan fungsi atas k , maka ϕ_{kk} disebut fungsi pasial (PACF).

2.4 White Noise

Makridakis, dkk (1999) menjelaskan bahwa suatu model bersifat *white noise* artinya residual dari model tersebut telah memenuhi asumsi identik (variansi residual homogen) dan asumsi independen (antar residual tidak berkorelasi).

1. Uji asumsi identik

Karena pada tahap identifikasi Y_t sudah stasioner terhadap rata-rata dan variansi, maka model sudah bisa dikatakan identik.

2. Uji asumsi independen

Pengujian ini dilakukan untuk mengetahui adanya autokorelasi dari nilai-nilai residual dengan menggunakan uji *Ljung Box-Pierce*.

Wei (2006) juga menjelaskan bahwa suatu proses (a_t) disebut proses *white noise* jika deretnya terdiri dari variabel random yang tidak berkorelasi dan

bedistribusi normal dengan rata-rata konstan yaitu $E(\mathbf{a}_t) = \boldsymbol{\mu}_a = 0$, variansi konstan $Var(\mathbf{a}_t) = \sigma_a^2$ dan $Cov(\mathbf{a}_t, \mathbf{a}_{t-k}) = \gamma_k = 0$ untuk $k \neq 0$. Dengan demikian fungsi akan stasioner dengan autokovariansi (γ_k)

$$\gamma_k = \begin{cases} \sigma_a^2, & \text{jika } k = 0 \\ 0, & \text{jika } k \neq 0 \end{cases} \quad (2.18)$$

fungsi autokorelasi (ρ_k)

$$\rho_k = \begin{cases} 1, & \text{jika } k = 0 \\ 0, & \text{jika } k \neq 0 \end{cases} \quad (2.19)$$

dan fungsi autokorelasi parsial (ϕ_{kk})

$$\phi_{kk} = \begin{cases} 1, & \text{jika } k = 0 \\ 0, & \text{jika } k \neq 0 \end{cases} \quad (2.20)$$

Makridakis, dkk (1999) menjelaskan bahwa pengujian asumsi white noise dilakukan dengan menggunakan uji *Ljung Box-Pierce* dengan hipotesa sebagai berikut:

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

$$H_1 : \text{minimal ada satu } \rho_i, i = 1, 2, \dots, k \text{ yang tidak sama dengan nol}$$

dengan statistik uji yaitu:

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

dengan daerah penolakan H_0 atau $H_1 : Q > \chi^2(\alpha; K - p - q)$

dimana:

K : lag maksimum

n : jumlah data

k : lag ke-k

p : orde dari AR

q : orde dari MA

$\hat{\rho}_k^2$: autokorelasi residual untuk *lag* ke- k

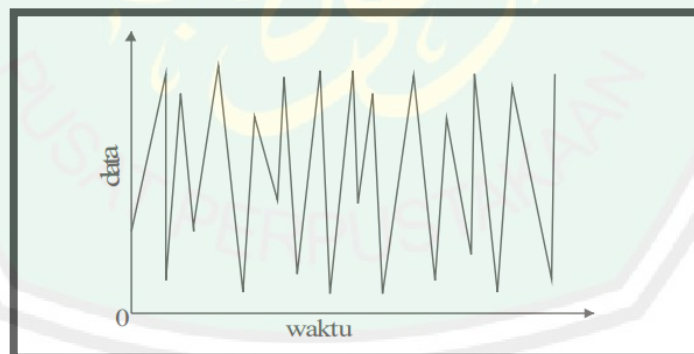
2.5 Stasioneritas Data

Stasioneritas berarti tidak terdapat perubahan yang drastis pada data. Fluktuasi data berada di sekitar suatu nilai rata-rata yang konstan, tidak tergantung pada waktu dan varians dari fluktuasi tersebut (Makridakis dkk, 1999). Bentuk visual dari plot data *time series* sering kali cukup meyakinkan para peramal bahwa data tersebut stasioner atau nonstasioner.

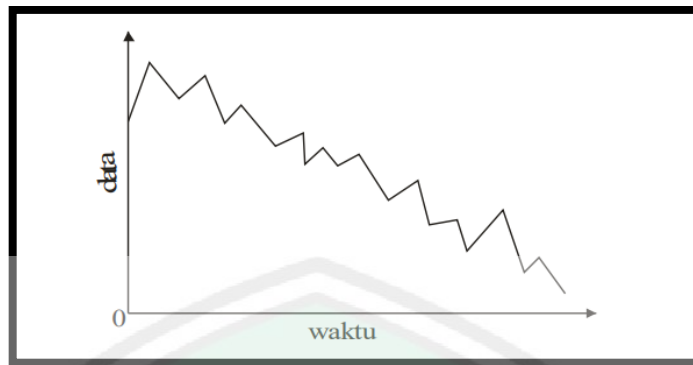
Menurut Wei (2006), stasioneritas dibagi menjadi dua, yaitu:

1. Stasioneritas dalam Rata-rata

Stasioner dalam rata-rata adalah fluktuasi data berada di sekitar suatu nilai rata-rata yang konstan (tidak terdapat pola *trend*). Untuk menstasionerkan data tidak stasioner dalam rata-rata dapat dilakukan proses pembedaan (*differencing*).



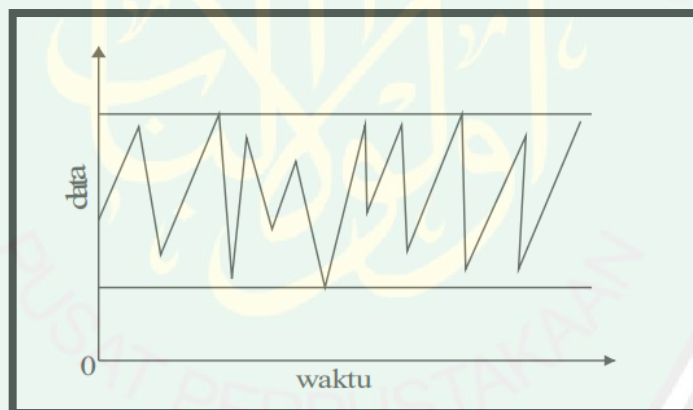
Gambar 2.1 Contoh Plot Data Stasioner dalam Rata-rata dan Variansi



Gambar 2.2 Contoh Plot Data Tidak Stasioner dalam Rata-rata

2. Stasioneritas dalam Variansi

Suatu data *time series* dikatakan stasioner dalam variansi apabila fluktuasi datanya tetap atau konstan (horizontal sepanjang sumbu waktu). Untuk menstasionerkan data yang tidak stasioner dalam variansi dapat dilakukan transformasi data.



Gambar 2.3 Contoh Pola Data Stasioner dalam Variansi

2.5.1 Differencing

Data *time series* dikatakan stasioner jika rata-rata dan variansinya konstan, tidak ada unsur *trend* dalam data, dan tidak ada unsur musiman. Apabila data tidak stasioner, maka perlu dilakukan modifikasi untuk menghasilkan data yang stasioner. Salah satu cara yang umum dipakai adalah metode *differencing* (Makridakis dkk, 1999).

Menurut Makridakis dkk (1999) notasi yang sangat bermanfaat dalam metode pembedaan (*differencing*) adalah operator langkah mundur (*backward shift*), sebagai berikut:

$$BY_t = Y_{t-1} \quad (2.22)$$

dengan:

Y_t : variabel Y pada waktu ke- t

Y_{t-1} : variabel Y pada waktu ke- $(t-1)$

B : *backward shift* (operator langkah mundur)

Notasi B yang dipasang pada Y_t mempunyai pengaruh menggeser data satu periode ke belakang. Misalkan apabila suatu *time series* tidak stasioner, maka data tersebut dapat dibuat lebih mendekati stasioner dengan melakukan *differencing* pertama.

Rumus untuk *differencing* pertama yaitu (Makridakis dkk, 1999):

$$Y_t' = Y_t - Y_{t-1} \quad (2.23)$$

dengan:

Y_t' : Variabel Y pada waktu ke- t setelah *differencing*

Y_t : Variabel Y pada waktu ke- t

Y_{t-1} : Variabel Y pada waktu ke- $(t-1)$

Menggunakan operator langkah mundur, persamaan (2.23) dapat ditulis kembali menjadi

$$Y_t' = Y_t - BY_t \quad (2.24)$$

atau

$$Y_t' = (1 - B)Y_t \quad (2.25)$$

dimana $(1 - B)$ menyatakan *differencing* pertama.

Selanjutnya untuk *differencing* kedua yang merupakan *differencing* pertama dari *differencing* pertama yang sebelumnya. Jika *differencing* kedua dihitung, maka:

$$\begin{aligned}
 Y_t' &= Y_t' - Y_{t-1}' \\
 &= (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) \\
 &= Y_t - 2Y_{t-1} + Y_{t-2} \\
 &= (1 - 2B + B^2)Y_t \\
 &= (1 - B)^2 Y_t
 \end{aligned} \tag{2.26}$$

Differencing kedua pada persamaan (2.26) dinotasikan oleh $(1 - B)^2$.

Secara umum jika terdapat *differencing* ke- d untuk mencapai stasioneritas, dapat dinotasikan dengan (Makridakis dkk, 1999):

$$(1 - \phi_1 B + \phi_2 B^2 - \phi_3 B^3 + \dots - B^d) = (1 - B)^d \tag{2.27}$$

2.5.2 Transformasi

Transformasi *Box-Cox* merupakan transformasi pangkat pada variabel respons yang dikembangkan oleh *Box-Cox*, yang bertujuan untuk menormalkan data, melinierkan model regresi dan menghomogenkan variansi. *Box* dan *Cox* mempertimbangkan kelas transformasi berparameter tunggal, yaitu λ yang dipangkatkan pada variabel respons Y , sehingga diperoleh model transformasinya Y^λ dengan λ merupakan parameter yang harus diduga. Transformasi *Box-Cox* hanya diberlakukan pada variabel respons Y yang bertanda positif (Draper & Smith, 1998).

Misalkan $T(Y_t)$ merupakan fungsi transformasi dari Y_t , maka digunakan rumus berikut (Wei, 2006):

$$T(Y_t) = Y_t^\lambda = \frac{Y_t^\lambda - 1}{\lambda} \tag{2.28}$$

dengan

Y_t : data pada waktu ke- t

λ : nilai parameter transformasi

Dalam Wei (2006) terdapat tabel Transformasi Box-Cox yang menunjukkan beberapa nilai dari λ yang umum digunakan beserta bentuk transformasinya

Tabel 2.1 Transformasi Box-Cox

λ	Transformasi
-1	$\frac{1}{Y_t}$
-0.5	$\frac{1}{\sqrt{Y_t}}$
0	$\ln Y_t$
0.5	$\sqrt{Y_t}$
1	Y_t (Tidak ada transformasi)

Untuk $\lambda = 0$ sesuai dengan transformasi logaritma, kita notastikan bahwa

$$\begin{aligned}
 \lim_{\lambda \rightarrow 0} \frac{Y_t^\lambda - 1}{\lambda} &= \lim_{\lambda \rightarrow 0} \left(\frac{Y_t^\lambda}{\lambda} - \frac{1}{\lambda} \right) \\
 &= \lim_{\lambda \rightarrow 0} \left(\frac{1}{\lambda} [Y_t^\lambda - 1] \right) \\
 &= \lim_{\lambda \rightarrow 0} \left(\frac{1}{\lambda} [e^{\ln Y_t^\lambda} - 1] \right) \\
 &= \lim_{\lambda \rightarrow 0} \left(\frac{1}{\lambda} [e^{\lambda \ln Y_t} - 1] \right) \\
 &= \lim_{\lambda \rightarrow 0} \left(\frac{1}{\lambda} \left[1 + \lambda \ln Y_t + \frac{(\lambda \ln Y_t)^2}{2!} + \dots - 1 \right] \right) \\
 &= \lim_{\lambda \rightarrow 0} \left(\ln Y_t + \frac{\lambda \ln^2 Y_t}{2!} + \dots \right) \\
 &= \ln Y_t
 \end{aligned} \tag{2.29}$$

Misal ditemukan data *time series* Y_t dengan variansi $Var(Y_t) = a_\lambda/Y_t$ artinya $Var(Y_t) = a_\lambda Y_t^{-1}$. Berarti ambil $\lambda = -1$, dengan menggunakan Transformasi *Box-Cox*, maka data Y_t dapat diubah menjadi (Wei, 2006)

$$\begin{aligned}
 Y_t &= T(Y_t) \\
 &= \frac{Y_t^\lambda - 1}{\lambda} \\
 &= \frac{Y_t^{-1} - 1}{-1} \\
 &= 1 - Y_t^{-1} \\
 &= 1 - \frac{1}{Y_t}
 \end{aligned} \tag{2.30}$$

akan tetapi untuk mengubah data Y_t menjadi $\tilde{Y}_t = 1 - 1/Y_t$ mengakibatkan variansinya menjadi (Wei, 2006)

$$\begin{aligned}
 (\tilde{Y}_t) &= \left(1 - \frac{1}{Y_t}\right) \\
 &= (1) + \left(-\frac{1}{Y_t}\right) \\
 &= 0 + \left(-\frac{1}{Y_t}\right) \\
 &= \left(-\frac{1}{Y_t}\right) \\
 &= |-1| \left(\frac{1}{Y_t}\right) \\
 &= \left(\frac{1}{Y_t}\right) \\
 &= (Y_t)
 \end{aligned} \tag{2.31}$$

Jadi transformasi $T(Y_t) = \tilde{Y}_t = 1/Y_t$ digunakan untuk mewakili transformasi $T(Y_t) = \tilde{Y}_t$ karena variansinya sama.

2.6 Uji Stasioneritas

Mengetahui suatu data stasioner atau tidak, dapat dilakukan pengujian menggunakan uji korelogram atau uji formal akar unit yaitu *Augmented Dickey Fuller* (ADF) (Makridakis dkk, 1999).

1. Uji Korelogram

Bentuk visual dari suatu plot deret berkala seringkali cukup untuk meyakinkan para penduga bahwa data tersebut adalah stasioner atau tidak stasioner, demikian pula plot autokorelasi dapat dengan mudah memperlihatkan ketidakstasioneran. Nilai-nilai autokorelasi dari data stasioner akan turun sampai nol sesudah *time-lag* kedua atau ketiga, sedangkan untuk data yang tidak stasioner, nilai-nilai tersebut berbeda signifikan dari nol untuk beberapa periode waktu (Makridakis dkk, 1999).

Uji korelogram merupakan metode pengujian yang digunakan untuk melihat kestasioneran data. Pada korelogram, suatu data dikatakan stasioner apabila plot autokorelasi dari data tidak keluar dari garis bartlett (garis putus-putus). Nilai probabilitas dari *lag* pertama hingga *lag* terakhir akan bergerak mendekati nol atau lebih kecil dari nilai taraf signifikansi α (Rosadi, 2012).

2. Uji *Augmented Dickey-Fuller*

Dickey dan Fuller mengembangkan pengujian kestasioneran untuk model runtun waktu tunggal dengan orde tinggi yang disebut dengan ADF. Menurut Gujarati (2004) uji ADF dapat digunakan pada model autoregresif berorde 2 atau lebih. Misalkan pada model AR(2) seperti berikut:

$$Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + a_t \quad (2.32)$$

Melalui operasi penjumlahan dan pengurangan pada ruas kanan persamaan

(2.32) dengan $\phi_2 Y_{t-1}$, diperoleh:

$$\begin{aligned} Y_t &= \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + (\phi_2 Y_{t-1} - \phi_2 Y_{t-1}) + a_t \\ &= \mu + (\phi_1 + \phi_2) Y_{t-1} - \phi_2 (Y_{t-1} - Y_{t-2}) + a_t \\ &= \mu + (\phi_1 + \phi_2) Y_{t-1} - \phi_2 \Delta Y_{t-1} + a_t \end{aligned} \quad (2.33)$$

Selanjutnya, kurangi ruas kiri dan kanan dengan Y_{t-1} , sehingga diperoleh:

$$Y_t - Y_{t-1} = \mu + (\phi_1 + \phi_2) Y_{t-1} - Y_{t-1} - \phi_2 \Delta Y_{t-1} + a_t \quad (2.34)$$

Menurut Gujarati (2004) uji ADF pada model autoregresif berorde p atau AR(p) adalah sebagai:

$$Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + a_t \quad (2.35)$$

dengan

- Y'_t : nilai variabel Y pada waktu ke- t setelah *differencing*
- Y_t : nilai variabel Y pada waktu ke- t
- Y_{t-1} : nilai variabel Y pada waktu ke- $(t - 1)$
- a_t : *error* pada waktu ke- t
- ϕ : koefisien regresi

Dari persamaan (2.35), hipotesis yang digunakan adalah:

$H_0 : \phi_1 = 0$ (terdapat *unit root*, variabel Y_t tidak stasioner)

$H_1 : \phi_1 < 1$ (tidak terdapat *unit root*, variabel Y_t stasioner)

Dengan kriteria keputusan tolak H_0 , jika $\phi_1 < \alpha$, dimana α adalah taraf signifikan.

Menurut Gujarati (2004) statistik uji yang digunakan adalah:

$$\tau = \frac{\sum_{k=1}^p \phi_{kk-1}}{SE\left(\sum_{k=1}^p \phi_{kk}\right)} \quad (2.36)$$

dengan:

ϕ_{kk} : autokorelasi parsial duga pada *lag-k*

$SE(\phi_{kk})$: standar *error* dari autokorelasi parsial pada *lag-k*

2.7 Time Series

Menurut Wei (2006) model *time series* adalah suatu peramalan nilai-nilai masa depan yang didasarkan pada nilai-nilai masa lampau suatu variabel dan atau kesalahan masa lampau. Model *time series* biasanya lebih sering digunakan untuk suatu peramalan atau prediksi.

2.7.1 Model Time Series Stasioner

2.7.1.1 Model Autoregressive

Model AR (p) secara umum dapat dituliskan sebagai berikut (Wei, 2006):

$$\phi_p(B)Z_t = a_t \quad (2.37)$$

atau jika diuraikan menjadi

$$\begin{aligned} (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p) Z_t &= a_t \\ Z_t - \phi_1 Z_{t-1} - \dots - \phi_p Z_{t-p} &= a_t \\ Z_t &= \phi_1 Z_{t-1} + \dots + \phi_p Z_{t-p} + a_t \end{aligned} \quad (2.38)$$

Karena $Z_t = Y_t - \mu$, maka persamaan (2.38) dapat ditulis dalam bentuk sebagai berikut

$$\begin{aligned}
Y_t - \mu &= \phi_1 (Y_{t-1} - \mu) + \dots + \phi_p (Y_{t-p} - \mu) + a_t \\
&= \phi_1 Y_{t-1} - \phi_1 \mu + \dots + \phi_p Y_{t-p} - \phi_p \mu + a_t \\
Y_t &= \mu - \phi_1 \mu - \phi_p \mu + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + a_t \\
&= \mu (1 - \phi_1 - \dots - \phi_p) + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + a_t \\
&= \mu (1 - (\phi_1 + \dots + \phi_p)) + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + a_t \\
&= \phi_0 + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + a_t
\end{aligned} \tag{2.39}$$

untuk $t = 1, 2, \dots, n$ bisa diuraikan sebagai berikut:

$$\begin{aligned}
Y_1 &= \phi_0 + \phi_1 Y_0 + \dots + \phi_p Y_{1-p} + a_1 \\
Y_2 &= \phi_0 + \phi_1 Y_1 + \dots + \phi_p Y_{2-p} + a_2 \\
&\vdots \\
Y_n &= \phi_0 + \phi_1 Y_{n-1} + \dots + \phi_p Y_{n-p} + a_n
\end{aligned}$$

atau dalam bentuk matriks (Wei, 2006)

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & Y_0 & \dots & Y_{1-p} \\ 1 & Y_1 & \dots & Y_{2-p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & Y_{n-1} & \dots & Y_{n-p} \end{bmatrix} \begin{bmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_p \end{bmatrix} + \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \tag{2.40}$$

dengan:

Y_t : Data pada periode ke- t , $t = 1, 2, \dots, n$

a_t : *Error* pada periode ke- t

μ : Suatu parameter konstanta

ϕ_i : Parameter koefisien *Autoregressive* ke- i , $i = 1, 2, \dots, p$

ϕ_0 : Konstanta rata-rata

Z_t : Selisih dari nilai variabel Y_t dengan μ

2.7.1.2 Model Moving Average

Menurut Wei (2006) model *Moving Average* MA ditemukan oleh Slutsky pada tahun 1927. Model MA merupakan proses hasil regresi dari data dengan

kesalahannya. Model MA orde q (MA(q)) secara umum dapat dituliskan sebagai berikut:

$$\begin{aligned} Z_t &= \theta_q(B) a_t \\ &= (1 - \theta_1 B - \dots - \theta_q B^q) a_t \\ &= a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \end{aligned} \quad (2.41)$$

Karena $Z_t = Y_t - \mu$ dan diasumsikan bahwa $\mu = \theta_0$ maka persamaan (2.41) dapat ditulis dalam bentuk sebagai berikut (Wei, 2006):

$$\begin{aligned} Y_t - \mu &= a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \\ Y_t - \theta_0 &= (1 - \theta_1 B - \dots - \theta_q B^q) a_t \\ Y_t &= \theta_0 + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \end{aligned} \quad (2.42)$$

untuk $t = 1, 2, \dots, n$ bisa diuraikan sebagai berikut:

$$\begin{aligned} Y_1 &= \theta_0 + a_1 - \theta_1 a_0 - \dots - \theta_q a_{1-q} \\ Y_2 &= \theta_0 + a_2 - \theta_1 a_1 - \dots - \theta_q a_{2-q} \\ &\vdots \\ Y_n &= \theta_0 + a_n - \theta_1 a_{n-1} - \dots - \theta_q a_{n-q} \end{aligned}$$

atau dalam bentuk matriks

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & a_{1-1} & \dots & a_{1-q} \\ 1 & a_{2-1} & \dots & a_{2-q} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & a_{n-1} & \dots & a_{n-q} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_q \end{bmatrix} + \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \quad (2.43)$$

dengan:

Y_t : Data pada periode ke- t , $t = 1, 2, \dots, n$

a_t : *Error* pada periode ke- t

μ : Suatu parameter konstanta

θ_j : Parameter koefisien *Moving Average* ke- j , $j = 1, 2, \dots, q$

2.7.1.3 Model Autoregressive Moving Average

Model *Autoregressive Moving Average* (ARMA) merupakan gabungan antara AR (p) yang terdapat pada persamaan (2.37) dengan MA (q) yang terdapat pada persamaan (2.41) sehingga dinyatakan sebagai ARMA (p, q) berikut (Wei, 2006):

$$\phi_p(B)Z_t = \theta_q(B)a_t \quad (2.44)$$

atau

$$\begin{aligned} (1 - \phi_1 B - \dots - \phi_p B^p)Z_t &= (1 - \theta_1 B - \dots - \theta_q B^q)a_t \\ Z_t - \phi_1 Z_{t-1} - \dots - \phi_p Z_{t-p} &= a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \\ Z_t &= \phi_1 Z_{t-1} + \dots + \phi_p Z_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \end{aligned} \quad (2.45)$$

Karena $Z_t = Y_t - \mu$, maka persamaan (2.45) dapat ditulis dalam bentuk sebagai berikut:

$$\begin{aligned} Y_t - \mu &= \phi_1 (Y_{t-1} - \mu) + \dots + \phi_p (Y_{t-p} - \mu) + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \\ &= \phi_1 Y_{t-1} - \phi_1 \mu + \dots + \phi_p Y_{t-p} - \phi_p \mu + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \\ Y_t &= \mu - \phi_1 \mu - \phi_p \mu + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \\ &= \mu(1 - \phi_1 - \dots - \phi_p) + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \end{aligned} \quad (2.46)$$

dengan memisalkan $\mu(1 - \phi_1 - \dots - \phi_p) = \phi_0$, maka diperoleh

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \quad (2.47)$$

untuk $t = 1, 2, \dots, n$ bisa diuraikan sebagai berikut:

$$\begin{aligned} Y_1 &= \phi_0 + \phi_1 Y_{1-1} + \dots + \phi_p Y_{1-p} + a_1 - \theta_1 a_{1-1} - \dots - \theta_q a_{1-q} \\ Y_2 &= \phi_0 + \phi_1 Y_{2-1} + \dots + \phi_p Y_{2-p} + a_2 - \theta_1 a_{2-1} - \dots - \theta_q a_{2-q} \\ &\vdots \\ Y_n &= \phi_0 + \phi_1 Y_{n-1} + \dots + \phi_p Y_{n-p} + a_n - \theta_1 a_{n-1} - \dots - \theta_q a_{n-q} \end{aligned}$$

atau dalam bentuk matriks

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & Y_{1-1} & \dots & Y_{1-p} \\ 1 & Y_{2-1} & \dots & Y_{2-p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & Y_{n-1} & \dots & Y_{n-p} \end{bmatrix} \begin{bmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_p \end{bmatrix} - \begin{bmatrix} 1 & a_{1-1} & \dots & a_{1-q} \\ 1 & a_{2-1} & \dots & a_{2-q} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & a_{n-1} & \dots & a_{n-q} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_q \end{bmatrix} + \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \quad (2.48)$$

dengan,

Y_t : Data pada periode ke- $t, t = 1, 2, \dots, n$

a_t : Error pada periode ke- t

μ : Suatu parameter konstanta rata-rata

ϕ_0 : Konstanta rata-rata

ϕ_i : Parameter koefisien *Autoregressive* ke- $i, i = 1, 2, \dots, p$

θ_j : Parameter koefisien *Moving Average* ke- $j, j = 1, 2, \dots, q$

2.7.2 Model Time Series Nonstasioner

2.7.2.1 Model Autoregressive Integrated Moving Average

Model ARIMA (p, d, q) yaitu jika model ARMA yang terdapat pada persamaan (2.44) dan diikuti oleh proses *differencing* orde d pada persamaan (2.23), maka diperoleh model ARIMA (p, d, q) sebagai berikut (Wei, 2006):

$$\phi_p(B)(1-B)^d Y_t = \phi_0 + \theta_q(B)a_t \quad (2.49)$$

atau diuraikan menjadi

$$\begin{aligned} (1 - \phi_1 B - \dots - \phi_p B^p)(1 - B)^d Y_t &= \phi_0 + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \\ (1 - B^d - \phi_1 B + \phi_1 B^{1+d} - \dots - \phi_p B^p + \phi_p B^{p+d}) Y_t &= \phi_0 + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \\ Y_t - Y_{t-d} - \phi_1 Y_{t-1} + \phi_1 Y_{t-1-d} - \dots - \phi_p Y_{t-p} + \phi_p Y_{t-p-d} &= \phi_0 + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \end{aligned} \quad (2.50)$$

Sehingga diperoleh model ARIMA bentuk lain sebagai berikut

$$\begin{aligned}
Y_t &= \phi_0 + Y_{t-d} + \sum_{i=1}^p \phi_i (Y_{t-1} - Y_{t-p-d}) + a_t - \sum_{j=1}^q \theta_j a_{t-j} \\
&= \phi_0 + Y_{t-d} + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} - \phi_1 Y_{t-2} - \phi_p Y_{t-p-1} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}
\end{aligned} \quad (2.51)$$

2.7.2.2 Model Seasonal Autoregressive Integrated Moving Average

Sebelumnya telah diperoleh model ARIMA(p, d, q) pada persamaan (2.49) yang juga bisa ditulis sebagai:

$$(1-B)^d \phi_p(B) Y_t = \phi_0 + \theta_q(B) a_t \quad (2.52)$$

Menurut Makridakis dkk (1999) notasi ARIMA dapat diperluas untuk menangani aspek musiman, yaitu model SARIMA,

$$\text{ARIMA}(p, d, q)(P, D, Q)^S$$

dimana

(p, d, q) : bagian yang tidak musiman dari model

(P, D, Q) : bagian musiman dari model

S : jumlah periode per musim

dengan menambahkan AR, *differencing* dan MA yang mengandung unsur musiman akan diperoleh model SARIMA sebagai berikut:

$$\phi_p(B) \Phi_P(B^S) (1-B)^d (1-B^S)^D Y_t = \phi_0 + \theta_q(B) \Theta_Q(B^S) a_t \quad (2.53)$$

dimana:

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

$$\Phi_P(B^S) = 1 - \Phi_1 B^S - \dots - \Phi_P B^{Ps}$$

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

$$\Theta_Q(B^S) = 1 - \Theta_1 B^S - \dots - \Theta_Q B^{Qs}$$

2.8 Identifikasi Model

Menurut Gaynor dan Krirk (1994) dalam memilih beberapa p dan q dapat dibantu dengan mengamati pola fungsi autokorelasi dan fungsi autokorelasi parsial dengan kriteria berikut:

1. Jika ACF tepotong (*cut off*) setelah *lag* 1 atau 2; *lag* musiman tidak signifikan dan PACF perlahan-lahan menghilang (*dying down*), maka diperoleh model non musiman MA ($q = 1$ atau $q = 2$).
2. Jika ACF tepotong (*cut off*) setelah *lag* musiman; *lag* non musiman tidak signifikan dan PACF perlahan-lahan menghilang (*dying down*), maka diperoleh model musiman MA ($Q = 1$).
3. Jika ACF terpotong setelah *lag* musiman; *lag* non musiman terpotong (*cut off*) setelah *lag* 1 atau 2, maka diperoleh model non musiman dan musiman MA ($q = 1$ atau $Q = 1$).
4. Jika ACF perlahan-lahan menghilang (*dying down*) dan PACF terpotong (*cut off*) setelah *lag* 1 atau 2; *lag* musiman tidak signifikan, maka diperoleh model non musiman AR ($p = 1$ atau $p = 2$).
5. Jika ACF perlahan-lahan menghilang (*dying down*) dan PACF terpotong atau (*cut off*) setelah *lag* musiman; *lag* non musiman tidak signifikan, maka diperoleh model musiman AR ($P = 1$).
6. Jika ACF perlahan-lahan menghilang (*dying down*) dan PACF terpotong (*cut off*) setelah *lag* musiman; dan non musiman terpotong (*cut off*) setelah *lag* 1 atau 2, maka diperoleh model non musiman dan musiman AR ($p = 1$ atau $p = 2$ dan $P = 1$).

7. Jika ACF dan PACF perlahan-lahan menghilang (*dying down*) maka diperoleh campuran (ARMA) model.

2.9 Estimasi Parameter

2.9.1 Persamaan Kalman Filter

Kalman Filter merupakan suatu metode estimasi yang optimal. Komponen dasar dari metode *Kalman Filter* adalah persamaan pengukuran dan persamaan transisi. Dengan menggunakan data pengukuran untuk memperbaiki hasil estimasi. Secara umum metode *Kalman Filter* untuk sistem dinamik linier waktu diskrit, dapat dinyatakan sebagai berikut (Lewis dkk, 2008):

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{F}_k \mathbf{u}_k + \mathbf{G}_k \mathbf{w}_k \quad (2.54)$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (2.55)$$

$$\mathbf{x}_0 \sim (\bar{\mathbf{x}}_0, \mathbf{P}_{x0}), \mathbf{w}_k \sim (0, \mathbf{Q}_k), \mathbf{v}_k \sim (0, \mathbf{R}_k)$$

Tahap Inisialisasi:

$$\mathbf{P}_0 = \mathbf{P}_{x0}, \hat{\mathbf{x}}_0 = \bar{\mathbf{x}}_0 \quad (2.56)$$

Tahap Prediksi:

$$\text{Estimasi} : \hat{\mathbf{x}}_{k+1}^- = \mathbf{A}_k \hat{\mathbf{x}}_k \quad (2.57)$$

$$\text{Kovarian error} : \mathbf{P}_{k+1}^- = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^T + \mathbf{Q}_k \quad (2.58)$$

Tahap koreksi:

$$\text{Kalman gain} : \mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T (\mathbf{H}_{k+1} \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1})^{-1} \quad (2.59)$$

$$\text{Estimasi} : \hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_{k+1} (\mathbf{z}_{k+1} - \mathbf{H}_{k+1} \hat{\mathbf{x}}_{k+1}^-) \quad (2.60)$$

$$\text{Kovarian error} \quad : \quad \mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\mathbf{P}_{k+1}^- \quad (2.61)$$

dengan:

$\mathbf{x}_k$: variabel keadaan sistem pada waktu k yang nilai estimasinya awalnya adalah $\hat{\mathbf{x}}_0$ dan kovarian awal $\mathbf{P}_{x_0}$

$\mathbf{u}_k$: variabel input deterministik pada waktu k

$\mathbf{w}_k$: *noise* pada model system

$\mathbf{z}_k$: variabel pengukuran

$\mathbf{H}$: matriks pengukuran

$\mathbf{v}_k$: *noise* pada model pengukuran

$\mathbf{R}_k$: matriks kovarian

$\mathbf{P}_k$: kovarian *error*

$\mathbf{Q}_k$: matriks kovarian pada tahap prediksi

$\mathbf{A}_k\mathbf{F}_k\mathbf{G}_k$: matriks-matriks konstan di dalam ukuran yang berkesesuaian dengan $\mathbf{A} = n \times n$, $\mathbf{F} = m \times m$, dan $\mathbf{G} = p \times 1$

$\mathbf{I}$: matriks identitas

Proses metode *Kalman Filter* terdiri dari dua tahap, yaitu *time update* dan *measurement update*, pada tahap *time update* didefinisikan estimasi state $\hat{\mathbf{x}}_k^- \in \mathbb{R}^n$, disebut juga *priori state estimate*. Sedangkan untuk tahap *measurement update* didefinisikan dengan estimasi state $\hat{\mathbf{x}}_k \in \mathbb{R}^n$, disebut juga *posteriori state estimate*.

2.9.2 Penerapan *Kalman Filter* dalam Estimasi Parameter Model SARIMA

Setelah diperoleh model SARIMA maka akan dilakukan estimasi parameter dengan menggunakan *Kalman Filter*. Dengan koefisien

$\phi_0, \phi_p, \Phi_p, \theta_q, \Theta_Q$ adalah parameter yang diestimasi menggunakan *Kalman Filter*.

Diasumsikan sebagai *state vector* yang dibentuk dari koefisien $\phi_0, \phi_p, \Phi_p, \theta_q, \Theta_Q$

yaitu $\mathbf{x}_t = [\phi_0, \phi_p, \Phi_p, \theta_q, \Theta_Q]^T$.

Berikut ini persamaan model sistem dan pengukuran pada metode *Kalman Filter* (Lewis dkk, 2008),

$$\mathbf{x}_{t+1} = \mathbf{A}_t \mathbf{x}_t + \mathbf{w}_t \quad (2.62)$$

$$\mathbf{z}_t = \mathbf{H} \mathbf{x}_t + \mathbf{v}_t \quad (2.63)$$

dengan:

$\mathbf{x}_t$: variabel keadaan sistem pada waktu t yang nilai estimasinya awalnya adalah $\hat{\mathbf{x}}_0$ dan kovarian awal $\mathbf{P}_{x_0}$

$\mathbf{w}_t$: *noise* pada model system

$\mathbf{z}_t$: variabel pengukuran

$\mathbf{H}$: matriks pengukuran

$\mathbf{v}_t$: *noise* pada model pengukuran

$\mathbf{A}$: matriks konstan di dalam ukuran yang berkesesuaian dengan $\mathbf{A} = n \times n$, dan $\mathbf{H} = p \times 1$

2.10 Hasil Penelitian Sebelumnya

Penelitian ini diambil dari penelitian terdahulu, yaitu penelitian yang dilakukan oleh Kurniawan (2014) yang memiliki kesimpulan bahwa model data adalah ARIMA (1,1,1) untuk suhu udara dan ARIMA (0,1,1) untuk kecepatan angin. Selain itu penggunaan algoritma *Kalman Filter* mempunyai pengaruh baik terhadap perbaikan hasil prediksi terbukti dari nilai MAPE pada data suhu udara

dengan ARIMA adalah 2,20288828 sedangkan nilai MAPE dengan *Kalman Filter* mempunyai nilai yang lebih kecil yaitu 1,935357143. Hal yang sama juga berlaku pada data kecepatan angin dimana penggunaan algoritma *Kalman Filter* mempunyai pengaruh baik terhadap perbaikan hasil prediksi terbukti dari nilai MAPE dengan ARIMA adalah 28,90889689 sedangkan nilai MAPE dengan *Kalman Filter* mempunyai nilai yang lebih kecil yaitu 18,73821429.

Menurut penelitian yang dilakukan oleh Pamungkas (2017) bahwa model ARIMA yang parameternya diestimasi menggunakan *Kalman Filter* hasil prediksinya lebih akurat dibandingkan sebelum diestimasi menggunakan *Kalman Filter*. Hal ini diketahui dari nilai rata-rata MAPE dari masing-masing golongan darah yang parameternya diestimasi menggunakan *Kalman Filter* mempunyai nilai lebih kecil daripada sebelum diestimasi menggunakan *Kalman Filter*. Sehingga *Kalman Filter* dapat digunakan untuk mengestimasi parameter model ARIMA dan dapat meminimalkan nilai kesalahan (*error*) model ARIMA.

2.11 Kajian Islam Tentang Estimasi

Secara umum estimasi atau pendugaan atau peramalan berarti memprediksi apa yang akan terjadi di masa mendatang dengan pendapat atau perkiraan manusia. Estimasi sendiri bersifat prediktif karena tidak ada yang bisa memastikan secara pasti tentang kondisi mendatang kecuali Allah SWT. Secara umum pendugaan terdiri dari 2 jenis yaitu:

1. Pendugaan Ilmiah

Pendugaan ilmiah adalah pendugaan yang berasal dari perkiraan yang berbasiskan ilmu pengetahuan atau keilmiah. Pendugaan seperti ini diperbolehkan dan tidak diharamkan selagi memiliki manfaat dan kemasalahatan bagi umat. Allah sendiri

memberikan perintah untuk mengikuti pengetahuan bukan hawa nafsu seperti dalam surat Ar-Rum:29 yang artinya:

“Tetapi orang-orang yang dzalim, mengikuti hawa nafsunya tanpa ilmu pengetahuan; maka siapakah yang akan menunjuki orang yang akan disesatkan Allah? Dan tiadalah bagi mereka seorang penolongpun”.

Contoh pendugaan yang berbasis kepada fakta ilmiah, data dan penelitian adalah pendugaan turunnya hujan, pendugaan turunnya bencana, pendugaan kelahiran bayi, pendugaan kondisi kesehatan dan lain-lain. Pendugaan-pendugaan ini bersifat ilmiah dan menggunakan ilmu pengetahuan yang benar. Jika digunakan sesuai sunnatulloh yang Allah berikan tentu akan memberikan manfaat yang banyak bagi umat manusia.

2. Pendugaan berasal dari Jin atau Tanpa Dasar

Pendugaan yang berasal dari ilmu hitam, jin atau orang pintar (*six sense*), adalah hal yang dilarang oleh islam. Hal ini disampaikan dalam Al-Quran surat Al-jinn: 8-10 yang artinya:

“Dan sesungguhnya kami telah mencoba mengetahui rahasia langit, maka kami mendapatinya penuh dengan penjagaan yang kuat dan penuh dengan lontaran api. Dan sesungguhnya kami dahulu dapat menduduki beberapa tempat di langit itu untuk mendengar-dengarkan beritanya. Tetapi sekarang barangsiapa yang mencoba mendengar-dengarkan seperti itu tentu akan menjumpai lontaran api yang mengintai untuk membakarnya. Dan sungguh dengan adanya penjagaan tersebut kami tidak mengetahui apakah keburukam yang dikehendaki bagi orang yang di bumi ataukah Rabb mereka menghendaki kebaikan bagi mereka”.

Dalam ayat di atas menjelaskan bahwa jin bisa mencuri informasi untuk masa depan. Untuk itu, banyak sekali peramal atau orang-orang yang bisa membaca masa depan tanpa dasar apapun karena bisikan setan atau berkawan dengan jin. Hal inilah yang mendekatkan kepada kesyirikan. Dalam hal lain, ada juga yang merasa memiliki kemampuan untuk membaca masa depan, padahal dirinya hanya

manusia yang memiliki keterbatasan. Hal ini juga di sampaikan dalam sebuah hadist,

“Barangsiapa yang mendatangi seorang peramal lalu menanyakan tentang suatu ramalan, maka tidak akan diterima sholatnya selama 40 malam” (HR. Muslim).

Contoh pendugaan atau peramalan yang diharamkan dalam islam adalah: meminta ramalan jodoh, meminta ramalan rezeki, meminta ramalan kematian, meminta ramalan nasib dan sebagainya. Hal ini tentu dilarang, karena semua bergantung kepada doa dan ikhtiar manusia bukan pada apa yang diramalkan. Nasib manusia bisa berubah ketika manusia bisa berusaha dengan ikhtiar yang kuat dan ketawakalan kepada Alloh SWT.



BAB III

METODOLOGI PENELITIAN

3.1 Pendekatan Penelitian

Pendekatan yang digunakan dalam penelitian ini adalah dengan pendekatan kuantitatif dengan bantuan studi literatur yang dilakukan dengan cara mengkaji buku-buku yang berkaitan dengan penelitian kuantitatif, dimana data yang digunakan dalam penelitian ini berupa angka atau data numerik.

3.2 Jenis dan Sumber Data

Jenis data yang digunakan dalam penelitian ini berupa data sekunder yang diperoleh dari Badan Meteorologi Klimatologi dan Geofisika (BMKG) Stasiun Klimatologi Semarang berupa data jumlah Curah Hujan di Kota Semarang pada tahun 2005 sampai 2011.

3.3 Variabel Penelitian

Variabel dalam penelitian ini adalah variabel terikat berupa data jumlah curah hujan (Y_t)

3.4 Analisis Data

Implementasi Metode *Kalman Filter* pada Model ARIMA $(p, d, q)(P, D, Q)^S$.

Langkah-langkah yang harus dilakukan dalam implementasi metode *Kalman Filter* adalah sebagai berikut:

1. Mengidentifikasi data jumlah curah hujan
2. Menguji kestasioneran data

3. Mengestimasi parameter model ARIMA $(p, d, q)(P, D, Q)^S$ dengan metode *Kalman Filter*.
 - a. Menentukan model ARIMA $(p, d, q)(P, D, Q)^S$
 - b. Mengestimasi parameter model ARIMA $(p, d, q)(P, D, Q)^S$ dengan metode *Kalman Filter*.



BAB IV

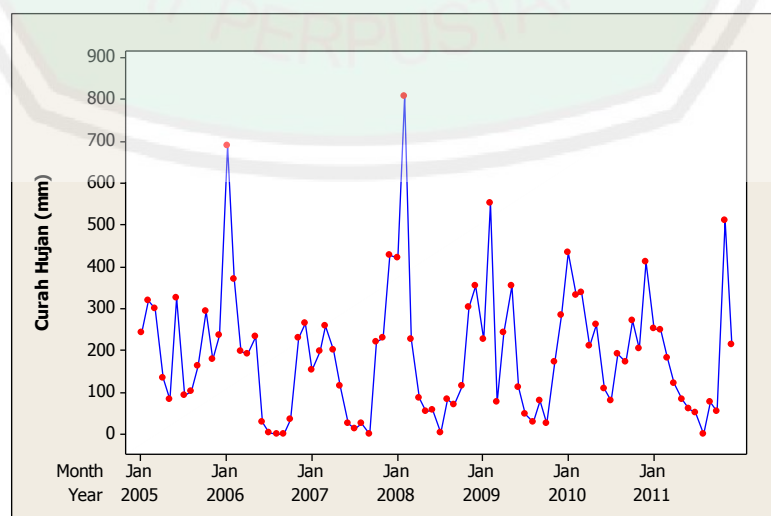
PEMBAHASAN

4.1 Implementasi Metode Kalman Filter pada Model ARIMA $(p, d, q)(P, D, Q)^S$

4.1.1 Identifikasi Data

Bab ini berisi tentang penerapan model SARIMA untuk pendugaan parameter pada data jumlah curah hujan di Kota Semarang. Data yang digunakan dalam penelitian ini adalah data pada tahun 2005 sampai dengan tahun 2011 seperti yang terlampir pada lampiran I. Adapun tahap-tahap pembentukan model yaitu identifikasi model, estimasi parameter model, dan diagnosis model. Tahap identifikasi data yaitu melihat kestasioneran data dan mencari model sementara yang sesuai dengan membuat plot data aktual, grafik autokorelasi, dan grafik autokorelasi parsial.

Langkah awal dalam analisis data deret berkala adalah membuat plot data secara grafis, pada data jumlah curah hujan dapat digambar grafiknya sebagai berikut:



Gambar 4.1 Plot Data Curah Hujan

Berdasarkan Gambar 4.1 grafik menunjukkan bahwa data jumlah curah hujan menunjukkan adanya plot berulang. Hal ini mengindikasikan adanya unsur musiman pada data tersebut. Gambar 4.1 juga menunjukkan bahwa nilai data tidak berfluktuasi di sekitar nilai rata-rata yang konstan. Fluktuasi data tidak konstan (horizontal sepanjang sumbu waktu) yang berarti data tersebut belum stasioner terhadap rata-rata dan variansi, sehingga perlu dilakukan proses transformasi dan *differencing* pada data tersebut.

Perhitungan ACF adalah sebagai berikut:

$$r_{Y_t Y_{t+1}} = \frac{\sum_{t=1}^n (Y_t - \bar{Y})(Y_{t+1} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2}$$

dengan

$$\bar{Y} = \frac{\sum_{t=1}^n Y_t}{n} = \frac{16196}{84} = 192.8095$$

misal

$$a = (241 - 192.8095)(319 - 192.8095) + \dots + (512 - 192.8095)(214 - 192.8095)$$

$$b = (241 - 192.8095)^2 + \dots + (214 - 192.8095)^2$$

sehingga diperoleh

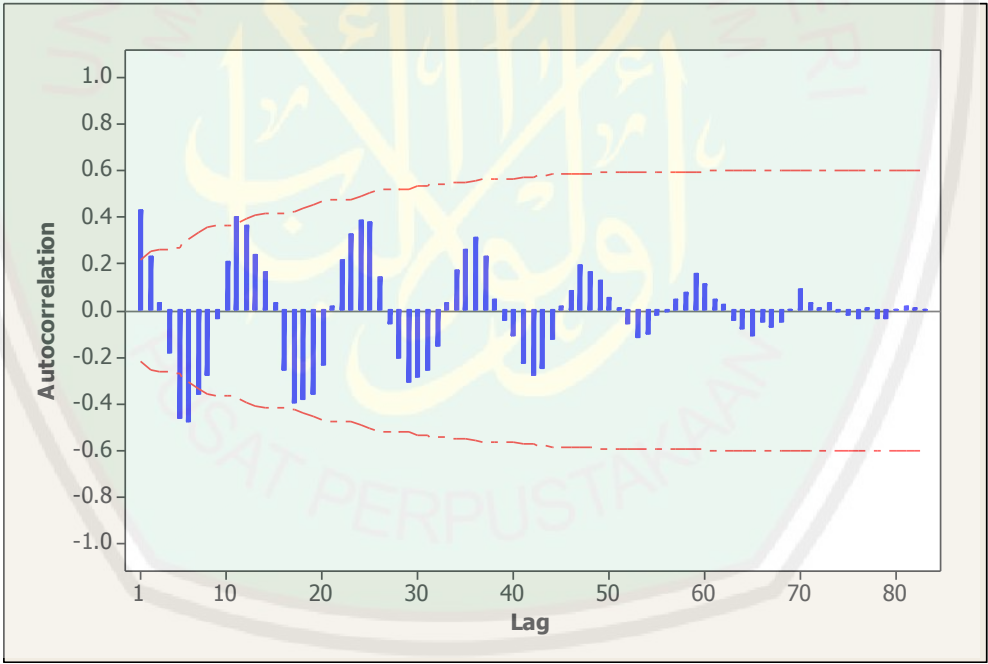
$$\hat{\rho}_1 = \frac{a}{b} = 0.435492$$

dengan cara serupa diperoleh nilai ACF seperti pada Tabel 4.1 berikut, dan lebih lengkapnya pada lampiran II.

Tabel 4.1 Nilai Koefisien ACF Data Asli

Lag	ACF	T	LBQ				
1	0.435492	3.99	16.51	16	-0.255566	-1.22	134.18
2	0.232320	1.81	21.26	17	-0.394284	-1.84	150.94
3	0.032593	0.24	21.36	18	-0.379563	-1.71	166.71
4	-0.183840	-1.38	24.41	19	-0.358909	-1.56	181.03
5	-0.463761	-3.41	44.08	20	-0.229895	-0.97	186.99
6	-0.478163	-3.11	65.25	21	0.019482	0.08	187.04
7	-0.355454	-2.08	77.10	22	0.216332	0.90	192.49
8	-0.277581	-1.55	84.43	23	0.328749	1.36	205.29
9	-0.036010	-0.20	84.55	24	0.385140	1.56	223.15
10	0.212609	1.15	88.97	25	0.381924	1.51	241.01
11	0.400199	2.14	104.82	26	0.140845	0.54	243.48
12	0.366675	1.86	118.31	27	-0.053227	-0.20	243.84
13	0.241466	1.18	124.24	28	-0.203385	-0.78	249.18
14	0.167519	0.80	127.13	29	-0.307866	-1.17	261.62
15	0.032400	0.15	127.24	30	-0.286245	-1.07	272.59

Untuk plot ACF dari data asli adalah sebagai berikut:



Gambar 4.2 Plot ACF Data Asli

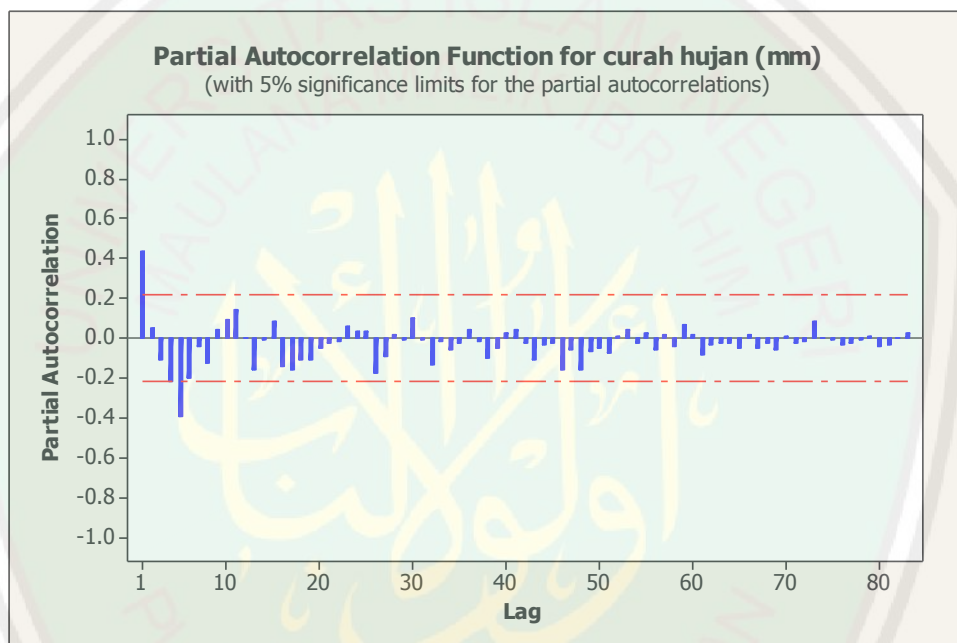
Berdasarkan Gambar 4.2 dan Tabel 4.1 dapat dilihat bahwa nilai koefisien ACF menurun menuju nol sehingga dapat dikatakan bahwa data telah stasioner terhadap rata-rata.

Sedangkan hasil dari perhitungan PACF dari data jumlah curah hujan sesuai persamaan (2.13) adalah sebagai berikut:

$$\begin{aligned}\hat{\phi}_{11} &= \hat{\rho}_1 = 0.435492 \\ \hat{\phi}_{22} &= \frac{\hat{\rho}_2 - \hat{\rho}_1^2}{1 - \hat{\rho}_1^2} = \frac{0.232320 - (0.435492)^2}{1 - (0.435492)^2} \\ &= 0.0526524227\end{aligned}$$

Nilai koefisien PACF yang selanjutnya didapatkan dengan menggunakan langkah yang serupa dan lebih lengkap terdapat pada Lampiran III.

Untuk plot PACF dari data asli adalah sebagai berikut:



Gambar 4.3 Plot PACF Data Asli

Berdasarkan plot ACF yang terdapat pada Gambar 4.2 dapat diketahui bahwa data memiliki pola *dying down* atau perlahan-lahan menuju nol, sedangkan plot PACF pada Gambar 4.3 memiliki pola *dying down*. Berikut merupakan beberapa model sementara yang memungkinkan yaitu sebagai berikut:

1. ARIMA (1,0,0)(1,0,1)¹²

Type	Coef	SE Coef	T	P
AR 1	-0.0746	0.1221	-0.61	0.543
SAR 12	-0.4880	0.1257	-3.88	0.000
SMA 12	0.8745	0.1074	8.14	0.000
Constant	6.702	2.206	3.04	0.003
Mean	4.191	1.379		

Number of observations: 72
 Residuals: SS = 836222 (backforecasts excluded)
 MS = 12297 DF = 68

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

Lag	12	24	36	48
Chi-Square	11.8	28.6	35.6	45.8
DF	8	20	32	44
P-Value	0.158	0.097	0.305	0.399

Gambar 4.4 Hasil Estimasi Parameter Model ARIMA (1,0,0)(1,0,1)¹²

Gambar 4.4 menunjukkan bahwa nilai *p – value* untuk parameter model AR melebihi taraf signifikansi, sehingga model tersebut kurang signifikan. Jadi model ARIMA(1,0,0)(1,0,1)¹² kurang tepat untuk digunakan.

2. ARIMA (0,0,1)(1,0,1)¹²

Type	Coef	SE Coef	T	P
SAR 12	-0.4837	0.1262	-3.83	0.000
MA 1	0.0590	0.1221	0.48	0.631
SMA 12	0.8749	0.1075	8.14	0.000
Constant	6.161	2.078	2.96	0.004
Mean	4.153	1.401		

Number of observations: 72
 Residuals: SS = 837344 (backforecasts excluded)
 MS = 12314 DF = 68

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

Lag	12	24	36	48
Chi-Square	11.4	27.8	34.9	45.3
DF	8	20	32	44
P-Value	0.179	0.114	0.331	0.418

Gambar 4.5 Hasil Estimasi Parameter Model ARIMA (0,0,1)(1,0,1)¹²

Gambar 4.5 menunjukkan bahwa nilai p – $value$ untuk parameter model MA melebihi taraf signifikansi, sehingga model tersebut kurang signifikan. Jadi model ARIMA(0,0,1)(1,0,1)¹² kurang tepat untuk digunakan.

3. ARIMA (1,0,1)¹²

Final Estimates of Parameters				
Type	Coef	SE Coef	T	P
SAR 12	-0.4737	0.1264	-3.75	0.000
SMA 12	0.8750	0.1072	8.16	0.000
Constant	6.152	2.176	2.83	0.006
Mean	4.174	1.477		
Number of observations: 72				
Residuals: SS = 840242 (backforecasts excluded)				
MS = 12177 DF = 69				
Modified Box-Pierce (Ljung-Box) Chi-Square statistic				
Lag	12	24	36	48
Chi-Square	9.8	25.8	33.5	44.6
DF	9	21	33	45
P-Value	0.364	0.215	0.442	0.491

Gambar 4.6 Hasil Estimasi Parameter Model ARIMA (1,0,1)¹²

Gambar 4.5 menunjukkan bahwa nilai p – $value$ untuk parameter model SAR dan SMA tidak melebihi taraf signifikansi, sehingga model tersebut signifikan. Jadi model ARIMA(1,0,1)¹² tepat untuk digunakan.

4.1.2 Pengujian Asumsi White Noise

Pengujian asumsi *white noise* dilakukan dengan menggunakan uji *Ljung-Box* dengan hipotesis sebagai berikut:

$$H_0 : \hat{\rho}_k = 0 \quad (\text{Error memenuhi asumsi white noise})$$

$$H_1 : \hat{\rho}_k \neq 0 \quad (\text{Error belum memenuhi asumsi white noise})$$

dengan statistik uji *Ljung-Box* adalah

$$\chi_{hitung}^2 = Q = n(n+2) \sum_{k=1}^m \frac{r_k^2}{n-k}$$

untuk taraf signifikansi 5%. Kriteria keputusan tolak H_0 jika $\chi^2_{hitung} > \chi^2_{(\alpha, df)}$ atau $p - value < \alpha$. Hasil dari uji diagnostik model dapat dilihat pada tabel 4.2 berikut

Tabel 4.2 Ringkasan Uji Ljung-Box

Lag	df	Statistik Ljung-Box (χ^2_{hitung})	$\chi^2_{(0,05, df)}$	P-value
12	9	9.8	16.9	0,364
24	21	25.8	32.7	0,215
36	33	33.5	47.4	0,442
48	45	44.6	61.7	0,491

Ringkasan pada Tabel 4.2 menunjukkan bahwa lag 12, 24, 36 dan 48 memiliki nilai statistik Ljung-Box tidak lebih dari nilai tabel ($\chi^2_{(0,05, df)}$). Sehingga dapat disimpulkan bahwa H_0 diterima yang berarti bahwa model memenuhi asumsi residual *white noise* dan berdistribusi normal sehingga model layak digunakan.

4.1.3 Estimasi Parameter Model ARIMA (1, 0, 1)¹²

Berdasarkan uji asumsi *White Noise* dan pemilihan model pada identifikasi data, diperoleh model yang sesuai untuk data *Inflow* Waduk Saguling adalah ARIMA(1,0,1)¹². Dari model umum pada persamaan (2.48) diperoleh:

$$\phi_0(B)\Phi_1(B^{12})(1-B)^0(1-B^{12})^0 Y_t = \phi_0 + \theta_0(B)\Theta_1(B^{12})a_t$$

atau dapat ditulis

$$\begin{aligned}\Phi_1(B^{12})Y_t &= \Theta_1(B^{12})a_t \\ (1 - \Phi_1 B^{12})Y_t &= (1 - \Theta_1 B^{12})a_t \\ Y_t - \Phi_1 B^{12}Y_t &= a_t - \Theta_1 B^{12}a_t \\ Y_t - \Phi_1 Y_{t-12} &= a_t - \Theta_1 a_{t-12}\end{aligned}$$

sehingga diperoleh bentuk model sebagai berikut:

$$Y_t = \Phi_1 Y_{t-12} - \Theta_1 a_{t-12} + a_t \quad (4.1)$$

4.1.4 Estimasi Parameter Model ARIMA (1,0,1)¹² dengan Metode Kalman Filter

Algoritma *Kalman Filter* yang digunakan pada penelitian ini adalah persamaan (4.1) yang diubah dalam bentuk *state space* berdasarkan model sistem pada persamaan (2.62) yaitu:

$$\begin{bmatrix} \Phi_1 \\ \Theta_1 \\ Y_t \end{bmatrix}_{t+1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Y_{t-12} & -(a_{t-12}) & 0 \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Theta_1 \\ Y_t \end{bmatrix}_t + w_t$$

dimana

$$x = \begin{bmatrix} \Phi_1 \\ \Theta_1 \\ Y_t \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Y_{t-12} & -(a_{t-12}) & 0 \end{bmatrix}$$

Model pengukuran dari penelitian ini menggunakan persamaan (2.63) atau dapat ditulis

$$z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Theta_1 \\ Y_t \end{bmatrix}_t + v_t$$

dimana

$$H = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

$$x = \begin{bmatrix} \Phi_1 \\ \Theta_1 \\ Y_t \end{bmatrix}$$

Setelah diperoleh model sistem dan pengukuran, selanjutnya dilakukan inisialisasi. Untuk nilai awal Y_t diambil dari data pertama jumlah curah hujan. Untuk nilai awal diberikan sebagai berikut:

Kovarian error model sistem: $R = 10^{-6}$

Kovarian error model pengukuran:

$$Q_0 = \begin{bmatrix} 10^{-6} & 0 & 0 \\ 0 & 10^{-6} & 0 \\ 0 & 0 & 10^{-6} \end{bmatrix}$$

Nilai estimasi awal:

$$\hat{x}_0 = \begin{bmatrix} 0.5 \\ 0.5 \\ 241 \end{bmatrix} \quad (4.2)$$

Nilai awal kovarian error:

$$P_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Selanjutnya masuk ke dalam tahap prediksi untuk iterasi pertama:

$$\begin{aligned}
 \hat{\mathbf{x}}_1^- &= \mathbf{A}_0 \hat{\mathbf{x}}_0 + \mathbf{w}_0 \\
 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \\ 241 \end{bmatrix} + \begin{bmatrix} 0.141 \times 10^{-4} \\ 0.084 \times 10^{-4} \\ 0.089 \times 10^{-4} \end{bmatrix} \\
 &= \begin{bmatrix} 0.5000141 \\ 0.5000084 \\ 0.0000089 \end{bmatrix}
 \end{aligned}$$

$$\begin{aligned}
 \mathbf{P}_1^- &= \mathbf{A}_0 \mathbf{P}_0 \mathbf{A}_0^T + \mathbf{Q}_0 \\
 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 10^{-6} & 0 & 0 \\ 0 & 10^{-6} & 0 \\ 0 & 0 & 10^{-6} \end{bmatrix} \\
 &= \begin{bmatrix} 1.000001 & 0 & 0 \\ 0 & 1.000001 & 0 \\ 0 & 0 & 1.000001 \end{bmatrix}
 \end{aligned}$$

Tahap koreksi:

Pada tahap koreksi melibatkan *Kalman gain*:

$$\begin{aligned}
 \mathbf{K}_1 &= \mathbf{P}_1^- \mathbf{H}_1^T (\mathbf{H}_1 \mathbf{P}_1^- \mathbf{H}_1^T + \mathbf{R}_1)^{-1} \\
 &= \begin{bmatrix} 1.000001 & 0 & 0 \\ 0 & 1.000001 & 0 \\ 0 & 0 & 1.000001 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \\
 &\quad \left(\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1.000001 & 0 & 0 \\ 0 & 1.000001 & 0 \\ 0 & 0 & 1.000001 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + 10^{-6} \right)^{-1} \\
 &= \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix}
 \end{aligned}$$

Lalu nilai $\mathbf{P}_{t+1}$ dicari menggunakan nilai $\mathbf{P}_{t+1}^-$ yang diperoleh dari tahap prediksi.

$$\begin{aligned}
P_1 &= (I - K_1 H_1) P_1^- \\
&= \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} 1.000001 & 0 & 0 \\ 0 & 1.000001 & 0 \\ 0 & 0 & 1.000001 \end{bmatrix} \\
&= \begin{bmatrix} 1.000001 & 0 & 0 \\ 0 & 1.000001 & 0 \\ 0 & 0 & 5 \times 10^{-6} \end{bmatrix}
\end{aligned}$$

Kemudian nilai $\hat{x}_{t+1}$ diestimasi dengan menggunakan nilai $\hat{x}_{t+1}^-$ yang diperoleh dari tahap prediksi.

$$\begin{aligned}
\hat{x}_1 &= \hat{x}_1^- + K_1 (z_1 - H_1 \hat{x}_1^-) \\
&= \begin{bmatrix} 0.5000141 \\ 0.5000084 \\ 0.0000089 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix} \left(\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.1 \\ 0.1 \\ 241 \end{bmatrix} + \begin{bmatrix} 0.141 \times 10^{-4} \\ 0.084 \times 10^{-4} \\ 0.089 \times 10^{-4} \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5000141 \\ 0.5000084 \\ 0.0000089 \end{bmatrix} \right) \\
&= \begin{bmatrix} 0.500014169 \\ 0.500008431 \\ 0.000008957 \end{bmatrix}
\end{aligned}$$

Untuk iterasi selengkapnya bisa dilihat di lampiran VII. Dari tahap-tahap di atas diperoleh nilai *error* yaitu

$$Error = Y_{tt} - Y_{tu} = \begin{bmatrix} 1.86 \times 10^{-2} \\ 1.32 \times 10^{-2} \\ \vdots \\ 0.37 \times 10^{-2} \end{bmatrix} \quad \text{atau}$$

$$\begin{aligned}
Norm &= \sqrt{|x_1|^2 + |x_2|^2 + \dots + |x_n|^2} \\
&= \sqrt{|1.86 \times 10^{-2}|^2 + |1.32 \times 10^{-2}|^2 + \dots + |0.37 \times 10^{-2}|^2} \\
&= 0.143
\end{aligned}$$

dimana

Y_{tt} : Data hasil estimasi

Y_{tu} : Data asli

$x_1, x_2, \dots, x_n$: Error ke-1,2, ..., n

Selanjutnya jika persamaan (4.2) diubah nilai awalnya maka menghasilkan *error* seperti pada tabel sebagai berikut

Tabel 4.3 Nilai *error* pada setiap nilai awal

No	Nilai Awal	Nilai <i>Error</i>
1.	$\hat{x}_0 = \begin{bmatrix} 1 \\ 1 \\ 241 \end{bmatrix}$	0.146
2.	$\hat{x}_0 = \begin{bmatrix} 1.5 \\ 1.5 \\ 241 \end{bmatrix}$	0.151
3.	$\hat{x}_0 = \begin{bmatrix} 0.7 \\ 0.7 \\ 241 \end{bmatrix}$	0.144
4.	$\hat{x}_0 = \begin{bmatrix} 0.1 \\ 0.1 \\ 241 \end{bmatrix}$	0.142

Iterasi dilakukan sebanyak jumlah data observasi yaitu 84. Dari beberapa percobaan nilai awal diatas dapat disimpulkan bahwa tabel pada nomor 4 memiliki nilai *error* yang paling kecil. Sehingga hasil estimasi parameter model ARIMA menggunakan *Kalman Filter* dapat dilihat pada tabel 4.5.

Tabel 4.4 Hasil estimasi parameter model ARIMA (1,0,1)¹² menggunakan *Kalman Filter*

Model	Parameter	Koefisian
ARIMA(1,0,1) ¹²	AR(1) = Φ_1	0.1
	MA(1) = Θ_1	0.1

Hasil parameter model ARIMA (1,0,1)¹² menggunakan *Kalman Filter* yang diperoleh pada Tabel 4.5, disubstitusikan pada persamaan (4.1) sehingga diperoleh persamaan model sebagai berikut:

$$Y_t = 0.1Y_{t-12} - 0.1a_{t-12} + a_t$$



4.2 Pengaplikasian Konsep Pendugaan dalam Islam

Pendugaan merupakan kemampuan untuk menentukan sesuatu tanpa melakukan perhitungan eksak. Didalam matematika terdapat beberapa contoh pendugaan, diantaranya pendugaan jumlah (banyak) yaitu pendugaan terhadap jumlah banyaknya suatu objek yang tidak bisa dihitung jumlah sebenarnya, seperti contoh banyaknya mobil diparkiran suatu tempat umum, tentunya tidak bisa dihitung secara rinci berapa jumlahnya mobil yang parkir ditempat tersebut, tapi setidaknya bisa diduga berapa kisaran mobil yang ada, bisa berjumlah puluhan, ratusan, bahkan ribuan.

Pada penjelasan tulisan diatas terdapat penduga θ_1 dll merupakan contoh beberapa parameter yang belum diketahui nilainya, seperti halnya pada surat As-Shaffat مِائَةِ أَلْفٍ أَوْ يَزِيدُونَ yang bertujuan untuk menjelaskan bahwa Nabi Yunus diutus kepada umatnya yang berjumlah 100.000 atau lebih, pada ayat tersebut tidak disebutkan secara rinci jumlah objeknya, tetapi hanya menyebutkan perkiraan umat Nabi Yunus dikisaran angka 100.000.

Dari penjelasan diatas dapat dikatakan bahwa pendugaan itu bisa dihitung, tetapi tidak menggunakan perhitungan secara eksak, hal ini juga sesuai dengan penjelasan pada Al-Qur'an surat As-Shaffat ayat 147 bahwa ayat tersebut menjelaskan tentang pendugaan.

Dengan demikian pendugaan pada penelitian ini dapat dicari dengan memperoleh nilai galat/kesalahan yang minimum.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pembahasan pada penelitian ini, maka dapat diambil kesimpulan bahwa proses dan hasil estimasi parameter model ARIMA (1,0,1)¹² menggunakan metode *Kalman Filter* pada data jumlah Curah Hujan di Kota Semarang pada tahun 2005 sampai 2011 adalah

$$Y_t = 0.1Y_{t-12} - 0.1a_{t-12} + a_t$$

5.2 Saran

Berdasarkan hasil penelitian ini, maka saran untuk penelitian selanjutnya adalah melakukan pengembangan terhadap model ARIMA (1,0,1)¹² dengan metode *Kalman Filter* yakni melakukan peramalan data untuk periode selanjutnya.

DAFTAR RUJUKAN

- Abdullah, bin Muhammad. 2004. *Tafsir Ibnu Katsir Jilid 1*. Bogor: Pustaka Imam Asy-Syafi'i
- Abdurrahman, Syaikh bin Nashir as-Sa'di. 2006. *Tafsir As-Sa'di*. Alih bahasa oleh Muhammad Iqbal dkk. Jakarta: Pustaka Sahifa
- Atmaja, L. S. 1997. *Statistika untuk Bisnis dan Ekonomi*. Yogyakarta: Penerbit Andi.
- Aziz, A. 2010. *Ekonometrika Teori dan Praktik Eksperimen dengan Matlab*. Malang: UIN Mailiki Press
- Box, G.E.P., Jenkins, G.M., dan Reinsel, G.C. 1976. *Time Series Analysis. Forecasting and Control Third Edition*. San Fransisco: Golden Day.
- Draper, N. R. dan Smith, H. 1998. *Applied Regression Analysis Third ed*. New York: John Wiley.
- Firdaus, Muhammad. 2004. *Ekonometrika Suatu Pendekatan Aplikatif*. Jakarta: PT Bumi Aksara.
- Gaynor, P. E., dan Krirk, Patrick, R. C. 1994. *Time Series Modelling and Forecasting in Bussines and Economic*. Newyork, McGraw Hill.
- Gudono. 2011. *Analisis Data Multivariat*. Yogyakarta: BPFE.
- Gujarati, N. Damodar. 2004. *Dasar-dasar Ekonometrika*. Jakarta: Erlangga
- Hanke, J.E. dan Wichern, D.W. 2005. *Business Forecasting Eight Edition*. New Jersey: Pearson Prenticehall.
- Hasan, M.I. 2002. *Pokok-pokok Materi Metodologi Penelitian dan Aplikasinya*. Jakarta: Ghalia Indonesia.
- Heizer, Jay, dan Barry Render. 2009. *Manajemen Operasi Buku 1 Edisi 9*. Jakarta: Salemba 4.
- Kurniawan, T., Lukman Hanafi., dan Erna Apriliani. 2014. *Penerapan Metode Kalman Filter Dalam Perbaikan Hasil Prediksi Cuaca Dengan Metode ARIMA*. *Jurnal Sains dan Seni ITS*, Vol.3, No.2.
- Lains, A. 2003. *Ekonometrika: Teori dan Aplikasi*. Jakarta: LP3ES Indonesia.
- Lewis, F.L., Xie, L., dan Popa. 2008. *Optimal and Robust Estimation with an Introduction to Stochastic Control Theory*. Second Edition. London: CRC Press.

- Makridakis, S., Wheelwright, S.C., dan McGee. 1999. *Metode dan Aplikasi Peramalan Edisi Kedua*. Jakarta: Penerbit Erlangga.
- Nawari. 2010. *Analisis Regresi dengan MS Excel 2007 dan SPSS 17*. Jakarta: PT Elex Media Komputindo.
- Pamungkas, M.H., Erna Apriliani dan Nuri Wahyuningsih. 2017. *Estimasi Parameter Model ARIMA Menggunakan Kalman Filter untuk Peramalan Permintaan Darah*. *Jurnal Sains dan Seni ITS*, Vol.2, No.2.
- Pankratz, A. 1983. *Forecasting With Univariate Box-Jenkins Models*. Canada: John Willey & Sons, Inc.
- Purnomo, M. H. dan Muntasa, A. 2015. *Konsep Pengolahan Citra Digital dan Ekstraksi Fitur*. Yogyakarta: Graha Ilmu.
- Rahayu, S. dan Tarno. 2006. *Prediksi Produksi Jagung di Jawa Tengah dengan Arima dan Bootstrap*. Prosiding SPMIPA. Semarang: Universitas Diponegoro.
- Rosadi, D. 2012. *Pengantar Analisis Runtun Waktu*. Yogyakarta: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada.
- Sembiring, R.K. 1995. *Analisis Regresi*. Bandung: Penerbit ITB.
- Subagyo, P. 1986. *Forecasting Konsep dan Aplikasi*. Yogyakarta: BPFE.
- Supangat, Andi. 2010. *Statistik dalam Kajian Deskriptif, Inferensi dan Nonparametrik*. Jakarta: Kencana Prenada Media Group.
- Wei, W.W.S. 2006. *Time Series Analysis: Univariate and Multivariate Methods Second Edition*. New Jersey: Pearson Prentice Hall.
- Wibisono, Yusuf. 2009. *Metode Statistik*. Yogyakarta: Gadjah Mada University Press.
- Zulaikha, R. 2014. *Estimasi Parameter pada Model Statistik Nonlinier secara Least Square*. Skripsi tidak dipublikasikan. Malang: UIN Maulana Malik Ibrahim Malang.

LAMPIRAN I

DATA JUMLAH CURAH HUJAN DI KOTA SEMARANG PADA TAHUN 2005 SAMPAI 2011

BULAN	TAHUN						
	2005	2006	2007	2008	2009	2010	2011
Januari	241	689	154	421	225	433	253
Februari	319	369	198	806	552	332	248
Maret	299	197	257	227	75	337	183
April	133	190	202	85	243	210	121
Mei	81	232	115	55	354	263	81
Juni	324	27	25	58	110	107	59
Juli	91	2	12	2	46	79	52
Agustus	102	0	25	81	27	190	0
September	164	0	1	70	78	171	75
Oktober	294	36	220	114	25	272	53
November	179	228	228	304	171	205	512
Desember	237	266	429	353	285	413	214

LAMPIRAN II

NILAI ACF DATA JUMLAH CURAH HUJAN

Lag	ACF	T	LBQ
1	0.435492	3.99	16.51
2	0.232320	1.81	21.26
3	0.032593	0.24	21.36
4	-0.183840	-1.38	24.41
5	-0.463761	-3.41	44.08
6	-0.478163	-3.11	65.25
7	-0.355454	-2.08	77.10
8	-0.277581	-1.55	84.43
9	-0.036010	-0.20	84.55
10	0.212609	1.15	88.97
11	0.400199	2.14	104.82
12	0.366675	1.86	118.31
13	0.241466	1.18	124.24
14	0.167519	0.80	127.13
15	0.032400	0.15	127.24
16	-0.255566	-1.22	134.18
17	-0.394284	-1.84	150.94
18	-0.379563	-1.71	166.71
19	-0.358909	-1.56	181.03
20	-0.229895	-0.97	186.99
21	0.019482	0.08	187.04
22	0.216332	0.90	192.49
23	0.328749	1.36	205.29
24	0.385140	1.56	223.15
25	0.381924	1.51	241.01
26	0.140845	0.54	243.48
27	-0.053227	-0.20	243.84
28	-0.203385	-0.78	249.18
29	-0.307866	-1.17	261.62
30	-0.286245	-1.07	272.59
31	-0.252043	-0.93	281.24
32	-0.153425	-0.56	284.51
33	0.034787	0.13	284.69
34	0.175491	0.64	289.14
35	0.264992	0.96	299.49
36	0.312713	1.12	314.21
37	0.236360	0.83	322.79
38	0.047537	0.17	323.15
39	-0.041829	-0.15	323.43
40	-0.106925	-0.37	325.31
41	-0.227462	-0.79	334.00
42	-0.276459	-0.96	347.14
43	-0.248528	-0.85	358.03
44	-0.123645	-0.42	360.79
45	0.018484	0.06	360.85
46	0.084456	0.29	362.21
47	0.192607	0.65	369.45
48	0.166422	0.56	375.01
49	0.125957	0.42	378.28
50	0.052266	0.18	378.86
51	0.011089	0.04	378.89
52	-0.058970	-0.20	379.67
53	-0.111825	-0.37	382.59
54	-0.103459	-0.35	385.17
55	-0.020662	-0.07	385.27
56	-0.004532	-0.02	385.28
57	0.045614	0.15	385.83
58	0.077175	0.26	387.49
59	0.161813	0.54	395.05
60	0.115707	0.38	399.08
61	0.049062	0.16	399.84
62	0.022824	0.08	400.01
63	-0.043755	-0.15	400.67
64	-0.075601	-0.25	402.73
65	-0.105745	-0.35	406.99
66	-0.050969	-0.17	408.03
67	-0.073057	-0.24	410.30
68	-0.046409	-0.15	411.27
69	0.004542	0.01	411.28
70	0.093595	0.31	415.80
71	0.032032	0.11	416.37
72	0.012734	0.04	416.47
73	0.031400	0.10	417.11
74	-0.002636	-0.01	417.12
75	-0.016171	-0.05	417.33
76	-0.032367	-0.11	418.28
77	0.008777	0.03	418.35
78	-0.033799	-0.11	419.73
79	-0.029977	-0.10	421.03
80	0.004615	0.02	421.07
81	0.017745	0.06	421.83
82	0.008952	0.03	422.11
83	0.000506	0.00	422.12

LAMPIRAN III

NILAI PACF DATA JUMLAH CURAH HUJAN

Lag	PACF	T			
1	0.435492	3.99	35	-0.026929	-0.25
2	0.052652	0.48	36	0.041724	0.38
3	-0.106649	-0.98	37	-0.015047	-0.14
4	-0.217145	-1.99	38	-0.097603	-0.89
5	-0.389798	-3.57	39	-0.049172	-0.45
6	-0.201805	-1.85	40	0.027324	0.25
7	-0.043646	-0.40	41	0.044834	0.41
8	-0.121652	-1.11	42	-0.023697	-0.22
9	0.045224	0.41	43	-0.106352	-0.97
10	0.092446	0.85	44	-0.033246	-0.30
11	0.145874	1.34	45	-0.026817	-0.25
12	-0.003671	-0.03	46	-0.156593	-1.44
13	-0.159653	-1.46	47	-0.057290	-0.53
14	-0.005239	-0.05	48	-0.162858	-1.49
15	0.082731	0.76	49	-0.069990	-0.64
16	-0.143137	-1.31	50	-0.054171	-0.50
17	-0.160289	-1.47	51	-0.077011	-0.71
18	-0.106102	-0.97	52	0.008729	0.08
19	-0.111365	-1.02	53	0.039350	0.36
20	-0.051710	-0.47	54	-0.023267	-0.21
21	-0.024656	-0.23	55	0.026213	0.24
22	-0.013255	-0.12	56	-0.057416	-0.53
23	0.058937	0.54	57	0.019738	0.18
24	0.032485	0.30	58	-0.044510	-0.41
25	0.035257	0.32	59	0.066072	0.61
26	-0.175425	-1.61	60	0.015493	0.14
27	-0.089386	-0.82	61	-0.084579	-0.78
28	0.018309	0.17	62	-0.031209	-0.29
29	-0.012036	-0.11	63	-0.029070	-0.27
30	0.103039	0.94	64	-0.025745	-0.24
31	-0.006996	-0.06	65	-0.046326	-0.42
32	-0.130387	-1.20	66	0.017063	0.16
33	-0.019843	-0.18	67	-0.047154	-0.43
34	-0.062037	-0.57	68	-0.023721	-0.22
			69	-0.056478	-0.52
			70	0.008933	0.08
			71	-0.024794	-0.23
			72	-0.014682	-0.13
			73	0.087413	0.80
			74	-0.003220	-0.03
			75	-0.006903	-0.06
			76	-0.033407	-0.31
			77	-0.023406	-0.21
			78	-0.009561	-0.09
			79	0.008775	0.08
			80	-0.040244	-0.37
			81	-0.036664	-0.34
			82	-0.000766	-0.01
			83	0.026426	0.24

LAMPIRAN IV

PROGRAM ESTIMASI PARAMETER DENGAN METODE KALMAN
FILTER

```

%Kalman Filter SARIMA

clc, clear

%Initial Data
%-----//
Data1=xlsread('nilaiawal.xlsx');
Data2=xlsread('dataurutX.xlsx');
sData2=size(Data2);
display(Data2)

%Matrices
%-----//
xhneg(:,1)=Data1(1:3,1);
Pneg(:, :, 1)=Data1(1:3,2:4);
Qj=Data1(1:3,5:7)*10^(-6);
Hj=[0;0;1];
Rj=10^(-6);
Y=Data2;

%Error Random Normal
%-----//
error=abs((5*randn(84,1)+5)*10^(-6));
serror= size(error);

%Vektor Galat a
%-----//
for t_i=1:84
    t_i2= 85-t_i;
    if t_i<serror(1)
        a(t_i2)=error(serror(1)-t_i+1);
    end
end
display(a)

%Matrix A keseluruhan (Balok)
%-----//
for t_i=1:85
    t=t_i-1;
    if t_i<=13
        t_ii=t_i+12;
        A(:, :, t_i)=[1, 0, 0; 0, 1, 0; 0, -a(t_ii), 0];
    elseif t_i>13 && t_i<=25
        A(:, :, t_i)=[1, 0, 0; 0, 1, 0; Y(t-12), -a(t-12), 0];
    elseif t_i>25
        A(:, :, t_i)=[1, 0, 0; 0, 1, 0; Y(t-12)-Y(t-24), -a(t-12),
0];

```

```

        end
        disp(t_i)
    end

    %Prediksi
    %-----//
    for t_i=1:84
        t=t_i-1;
        xhneg(:,t_i+1)=A(:, :, t_i)*xhneg(:,t_i)+a(t_i);
        Pneg(:, :, t_i+1)=A(:, :, t_i)*Pneg(:, :, 1)*A(:, :, t_i)'+Qj;
    end
    display(xhneg)
    display(Pneg)
    display(a(1))

    %Koreksi
    %-----//
    K=[];
    H=Hj';
    for t=1:85
        t_i=t-1;
        disp(H*Pneg(:, :, t)*H') %Evaluator
        K(:, :, t)=Pneg(:, :, t)*H'*inv(H*Pneg(:, :, t)*H'+Rj);
    end

    %Estimasi xh
    %-----//
    for t=1:85
        z(:, :, t)=H*xhneg(:, t);
        xh(:, t)=xhneg(:, t)+K(:, :, t)*(z(:, :, t)-H*xhneg(:, t));
    end

    %Estimasi Nilai P
    %-----//
    for t=1:85
        sP=size(Pneg(:, :, t));
        P(:, :, t)=(eye(sP(1), sP(2))-K(:, :, t)*H)*Pneg(:, :, t);
        %"Iterasi" terhadap t
    end

    %Displaying P
    %-----//
    format long
    Phasil=P(:, :, end);
    display(Phasil)

    xhasil=xh(:, end);
    display(xhasil)
    xhasil2=xh(3, :);

    jwb=input('tampilkan data model Y hasil? ');
    if jwb==1
        for t=25:84
            Ytt(t,1)=0.5*(xhasil2(t-12)+a(t)-0.5*a(t-12)/100);
        end
    end

```

```

        for t=1:24
            Ytt(t,1)=xhasil2(t);
        end
        Ytt=Ytt-mean(Ytt)*ones(size(Ytt))+mean(Data2);
        display(Ytt)
    else
        break
    end

    %Error Kalkman Filter
    %-----//
    Er=((Ytt-Y)/10^4);
    display(Er)

    %Display P0
    %-----//
    Phasil1=Pneg(:, :, 2);
    display(Phasil1);
    xhasil1=xhneg(:, 2);
    display(xhasil1);
    Khasil0=K(:, :, 2);
    display(Khasil0);
    xhhasil1=xh(:, 2);
    display(xhhasil1);
    P1=P(:, :, 2);
    display(P1);
    norm(Er);
    display(norm(Er));

```

LAMPIRAN V

HASIL PROGRAM KALMAN FILTER DARI ITERASI 1 SAMPAI 84

```

Pneg(:,1) =
    1    0    0
    0    1    0
    0    0    1
Pneg(:,2) =
    1.0000010000000000    0    0
    0    1.0000010000000000 -0.000007379302936
    0 -0.000007379302936    0.000001000054454
Pneg(:,3) =
    1.0000010000000000    0    0
    0    1.0000010000000000 -0.000012061163432
    0 -0.000012061163432    0.000001000145472
Pneg(:,4) =
    1.0000010000000000    0    0
    0    1.0000010000000000 -0.000005113042422
    0 -0.000005113042422    0.000001000026143
Pneg(:,5) =
    1.0000010000000000    0    0
    0    1.0000010000000000 -0.000004760652949
    0 -0.000004760652949    0.000001000022664
Pneg(:,6) =
    1.0000010000000000    0    0
    0    1.0000010000000000 -0.000013506673271
    0 -0.000013506673271    0.000001000182430
Pneg(:,7) =
    1.0000010000000000    0    0
    0    1.0000010000000000 -0.000002451441436
    0 -0.000002451441436    0.000001000006010
Pneg(:,8) =
    1.0000010000000000    0    0
    0    1.0000010000000000 -0.000004985725199
    0 -0.000004985725199    0.000001000024857
Pneg(:,9) =
    1.0000010000000000    0    0
    0    1.0000010000000000 -0.000009599335399
    0 -0.000009599335399    0.000001000092147
Pneg(:,10) =
    1.0000010000000000    0    0
    0    1.0000010000000000 -0.000005749043663
    0 -0.000005749043663    0.000001000033052
Pneg(:,11) =
    1.0000010000000000    0    0
    0    1.0000010000000000 -0.000012024667228
    0 -0.000012024667228    0.000001000144593

```

```

Pneg(:,12) =
    1.0000010000000000      0      0
      0  1.0000010000000000 -0.000010170607698
      0 -0.000010170607698  0.000001000103441
Pneg(:,13) =
    1.0000010000000000      0      0
      0  1.0000010000000000 -0.000006457851444
      0 -0.000006457851444  0.000001000041704
Pneg(:,14) =
    1.0000010000000000      0      0
      0  1.0000010000000000 -0.000001111507308
      0 -0.000001111507308  0.000001000001235
Pneg(:,15) =
    1.0e+004 *
    0.000100000100000      0  0.024100000000000
      0  0.000100000100000      0
    0.024100000000000      0  5.808100000100000
Pneg(:,16) =
    1.0e+005 *
    0.000010000010000      0  0.003190000000000
      0  0.000010000010000 -0.000000000049524
    0.003190000000000 -0.000000000049524  1.017610000010000
Pneg(:,17) =
    1.0e+004 *
    0.000100000100000      0  0.029900000000000
      0  0.000100000100000 -0.000000000155095
    0.029900000000000 -0.000000000155095  8.940100000099999
Pneg(:,18) =
    1.0e+004 *
    0.000100000100000      0  0.013300000000000
      0  0.000100000100000 -0.000000000166650
    0.013300000000000 -0.000000000166650  1.768900000100000
Pneg(:,19) =
    1.0e+003 *
    0.001000001000000      0  0.081000000000000
      0  0.001000001000000 -0.000000009320747
    0.081000000000000 -0.000000009320747  6.561000001000088
Pneg(:,20) =
    1.0e+005 *
    0.000010000010000      0  0.003240000000000
      0  0.000010000010000 -0.000000000055671
    0.003240000000000 -0.000000000055671  1.049760000010000
Pneg(:,21) =
    1.0e+003 *
    0.001000001000000      0  0.091000000000000
      0  0.001000001000000 -0.000000006991814
    0.091000000000000 -0.000000006991814  8.281000001000049
Pneg(:,22) =
    1.0e+004 *
    0.000100000100000      0  0.010200000000000

```

```

0 0.000100000100000 -0.000000000941985
0.010200000000000 -0.000000000941985 1.040400000100009
Pneg(:,23) =
1.0e+004 *
0.000100000100000 0 0.016400000000000
0 0.000100000100000 -0.000000000590129
0.016400000000000 -0.000000000590129 2.689600000100004
Pneg(:,24) =
1.0e+004 *
0.000100000100000 0 0.029400000000000
0 0.000100000100000 -0.000000000775427
0.029400000000000 -0.000000000775427 8.643600000100005
Pneg(:,25) =
1.0e+004 *
0.000100000100000 0 0.017900000000000
0 0.000100000100000 -0.000000000841482
0.017900000000000 -0.000000000841482 3.204100000100007
Pneg(:,26) =
1.0e+004 *
0.000100000100000 0 0.023700000000000
0 0.000100000100000 -0.000000001085304
0.023700000000000 -0.000000001085304 5.616900000100012
Pneg(:,27) =
1.0e+005 *
0.000010000010000 0 0.004480000000000
0 0.000010000010000 -0.00000000073793
0.004480000000000 -0.00000000073793 2.007040000010001
Pneg(:,28) =
1.0e+003 *
0.001000001000000 0 0.050000000000000
0 0.001000001000000 -0.000000012061163
0.050000000000000 -0.000000012061163 2.500000001000145
Pneg(:,29) =
1.0e+004 *
0.000100000100000 0 -0.010200000000000
0 0.000100000100000 -0.000000000511304
-0.010200000000000 -0.000000000511304 1.040400000100003
Pneg(:,30) =
1.0e+003 *
0.001000001000000 0 0.057000000000000
0 0.001000001000000 -0.000000004760653
0.057000000000000 -0.000000004760653 3.249000001000023
Pneg(:,31) =
1.0e+004 *
0.000100000100000 0 0.015100000000000
0 0.000100000100000 -0.000000001350667
0.015100000000000 -0.000000001350667 2.280100000100018
Pneg(:,32) =
1.0e+004 *

```

```

0.000100000100000      0 -0.029700000000000
      0 0.000100000100000 -0.000000000245144
-0.029700000000000 -0.000000000245144 8.820900000100000
Pneg(:,33) =
1.0e+003 *
0.001000001000000      0 -0.089000000000000
      0 0.001000001000000 -0.0000000004985725
-0.089000000000000 -0.0000000004985725 7.921000001000025
Pneg(:,34) =
1.0e+004 *
0.000100000100000      0 -0.010200000000000
      0 0.000100000100000 -0.000000000959934
-0.010200000000000 -0.000000000959934 1.040400000100009
Pneg(:,35) =
1.0e+004 *
0.000100000100000      0 -0.016400000000000
      0 0.000100000100000 -0.000000000574904
-0.016400000000000 -0.000000000574904 2.689600000100003
Pneg(:,36) =
1.0e+004 *
0.000100000100000      0 -0.025800000000000
      0 0.000100000100000 -0.0000000001202467
-0.025800000000000 -0.0000000001202467 6.656400000100014
Pneg(:,37) =
1.0e+003 *
0.001000001000000      0 0.049000000000000
      0 0.001000001000000 -0.0000000010170608
0.049000000000000 -0.0000000010170608 2.401000001000103
Pneg(:,38) =
1.0e+002 *
0.010000010000000      0 0.290000000000000
      0 0.010000010000000 -0.0000000064578514
0.290000000000000 -0.0000000064578514 8.410000010000417
Pneg(:,39) =
1.0e+005 *
0.000010000010000      0 -0.005350000000000
      0 0.000010000010000 -0.000000000011115
-0.005350000000000 -0.000000000011115 2.862250000010000
Pneg(:,40) =
1.0e+004 *
0.000100000100000      0 -0.017100000000000
      0 0.000100000100000 -0.000000000783348
-0.017100000000000 -0.000000000783348 2.924100000100006
Pneg(:,41) =
1.0e+003 *
0.001000001000000      0 0.060000000000000
      0 0.001000001000000 -0.0000000001913106
0.060000000000000 -0.0000000001913106 3.600000000100004
Pneg(:,42) =
1.0e+002 *

```

```

0.010000010000000      0 0.120000000000000
      0 0.010000010000000 -0.000000062223734
0.120000000000000 -0.000000062223734 1.440000010000387
Pneg(:,43) =
1.0e+004 *
0.000100000100000      0 -0.011700000000000
      0 0.000100000100000 -0.000000000904219
-0.011700000000000 -0.000000000904219 1.368900000100008
Pneg(:,44) =
1.000001000000000      0 -2.000000000000000
      0 1.000001000000000 -0.000006065208492
-2.000000000000000 -0.000006065208492 4.000001000036787
Pneg(:,45) =
1.0e+002 *
0.010000010000000      0 0.100000000000000
      0 0.010000010000000 -0.000000093983858
0.100000000000000 -0.000000093983858 1.000000010000883
Pneg(:,46) =
1.0e+002 *
0.010000010000000      0 0.250000000000000
      0 0.010000010000000 -0.000000151943813
0.250000000000000 -0.000000151943813 6.250000010002309
Pneg(:,47) =
1.000001000000000      0 1.000000000000000
      0 1.000001000000000 -0.000009619662243
1.000000000000000 -0.000009619662243 1.000001000092538
Pneg(:,48) =
1.0e+004 *
0.000100000100000      0 0.018400000000000
      0 0.000100000100000 -0.000000000633459
0.018400000000000 -0.000000000633459 3.385600000100005
Pneg(:,49) =
1.000001000000000      0      0
      0 1.000001000000000 -0.000008208307532
      0 -0.000008208307532 0.000001000067376
Pneg(:,50) =
1.0e+004 *
0.000100000100000      0 0.016300000000000
      0 0.000100000100000 -0.000000000712743
0.016300000000000 -0.000000000712743 2.656900000100005
Pneg(:,51) =
1.0e+004 *
0.000100000100000      0 0.026700000000000
      0 0.000100000100000 -0.000000000157362
0.026700000000000 -0.000000000157362 7.128900000099999
Pneg(:,52) =
1.0e+005 *
0.000010000010000      0 0.006080000000000
      0 0.000010000010000 -0.000000000029179
0.006080000000000 -0.000000000029179 3.696640000010000

```

```

Pneg(:,53) =
1.0e+002 *
0.010000010000000      0 -0.300000000000000
      0 0.010000010000000 -0.00000011234391
-0.300000000000000 -0.00000011234391 9.000000010001237
Pneg(:,54) =
1.0e+004 *
0.000100000100000      0 -0.011700000000000
      0 0.000100000100000 -0.000000000478208
-0.011700000000000 -0.000000000478208 1.368900000100003
Pneg(:,55) =
1.0e+003 *
0.001000001000000      0 -0.060000000000000
      0 0.001000001000000 -0.0000000007912116
-0.060000000000000 -0.0000000007912116 3.600000001000063
Pneg(:,56) =
1.0e+003 *
0.001000001000000      0 0.033000000000000
      0 0.001000001000000 -0.000000000032500
0.033000000000000 -0.000000000032500 1.089000001000000
Pneg(:,57) =
1.0e+002 *
0.010000010000000      0 -0.100000000000000
      0 0.010000010000000 -0.0000000053225837
-0.100000000000000 -0.0000000053225837 1.000000010000283
Pneg(:,58) =
1.0e+003 *
0.001000001000000      0 0.056000000000000
      0 0.001000001000000 -0.0000000008001460
0.056000000000000 -0.0000000008001460 3.136000001000064
Pneg(:,59) =
1.0e+003 *
0.001000001000000      0 0.069000000000000
      0 0.001000001000000 -0.0000000001807575
0.069000000000000 -0.0000000001807575 4.761000001000004
Pneg(:,60) =
1.0e+004 *
0.000100000100000      0 -0.010600000000000
      0 0.000100000100000 -0.0000000000673796
-0.010600000000000 -0.0000000000673796 1.123600000100005
Pneg(:,61) =
1.0e+003 *
0.001000001000000      0 0.076000000000000
      0 0.001000001000000 -0.0000000004090784
0.076000000000000 -0.0000000004090784 5.776000001000017
Pneg(:,62) =
1.0e+003 *
0.001000001000000      0 -0.076000000000000
      0 0.001000001000000 -0.0000000000302326
-0.076000000000000 -0.0000000000302326 5.776000001000001

```

```

Pneg(:,63) =
1.0e+004 *
0.000100000100000      0 -0.019600000000000
      0 0.000100000100000 -0.000000000481233
-0.019600000000000 -0.000000000481233 3.841600000100002
Pneg(:,64) =
1.0e+004 *
0.000100000100000      0 -0.025400000000000
      0 0.000100000100000 -0.000000000448152
-0.025400000000000 -0.000000000448152 6.451600000100002
Pneg(:,65) =
1.0e+004 *
0.000100000100000      0 -0.015200000000000
      0 0.000100000100000 -0.000000000563988
-0.015200000000000 -0.000000000563988 2.310400000100003
Pneg(:,66) =
1.0e+004 *
0.000100000100000      0 0.015800000000000
      0 0.000100000100000 -0.000000000088462
0.015800000000000 -0.000000000088462 2.496400000100000
Pneg(:,67) =
1.0e+004 *
0.000100000100000      0 0.029900000000000
      0 0.000100000100000 -0.000000000004734
0.029900000000000 -0.000000000004734 8.940100000099999
Pneg(:,68) =
1.0e+003 *
0.001000001000000      0 0.052000000000000
      0 0.001000001000000 -0.000000000865162
0.052000000000000 -0.000000000865162 2.704000001000001
Pneg(:,69) =
1.0e+003 *
0.001000001000000      0 0.044000000000000
      0 0.001000001000000 -0.0000000003627139
0.044000000000000 -0.0000000003627139 1.936000001000013
-0.054000000000000 -0.0000000006441140 2.916000001000041
Pneg(:,71) =
1.000001000000000      0 8.000000000000000
      0 1.000001000000000 -0.000002970918601
8.000000000000000 -0.000002970918601 64.000001000008822
Pneg(:,72) =
1.0e+003 *
0.001000001000000      0 -0.089000000000000
      0 0.001000001000000 -0.0000000005551094
-0.089000000000000 -0.0000000005551094 7.921000001000031
Pneg(:,73) =
1.0e+004 *
0.000100000100000      0 -0.013300000000000
      0 0.000100000100000 -0.000000000893533
-0.013300000000000 -0.000000000893533 1.768900000100008

```

```

Pneg(:,75) =
1.0e+004 *
0.000100000100000      0 0.0208000000000000
      0 0.000100000100000 -0.0000000000546554
0.020800000000000 -0.0000000000546554 4.326400000100003
Pneg(:,76) =
1.0e+004 *
0.000100000100000      0 -0.0220000000000000
      0 0.000100000100000 -0.0000000000310921
-0.022000000000000 -0.0000000000310921 4.840000000100001
Pneg(:,77) =
1.0e+004 *
0.000100000100000      0 0.0262000000000000
      0 0.000100000100000 -0.0000000000241338
0.026200000000000 -0.0000000000241338 6.864400000099999
Pneg(:,78) =
1.0e+003 *
0.001000001000000      0 -0.0330000000000000
      0 0.001000001000000 -0.0000000004780907
-0.033000000000000 -0.0000000004780907 1.089000001000023
Pneg(:,79) =
1.0e+003 *
0.001000001000000      0 -0.0910000000000000
      0 0.001000001000000 -0.0000000009804126
-0.091000000000000 -0.0000000009804126 8.281000001000097
Pneg(:,80) =
1.000001000000000      0 -3.000000000000000
      0 1.000001000000000 -0.000013691224663
-3.000000000000000 -0.000013691224663 9.000001000187449
Pneg(:,81) =
1.0e+003 *
0.001000001000000      0 0.0330000000000000
      0 0.001000001000000 -0.0000000002848969
0.033000000000000 -0.0000000002848969 1.089000001000008
Pneg(:,82) =
1.0e+004 *
0.000100000100000      0 0.0163000000000000
      0 0.000100000100000 -0.0000000000313661
0.016300000000000 -0.0000000000313661 2.656900000100001
Pneg(:,83) =
1.0e+003 *
0.001000001000000      0 0.0930000000000000
      0 0.001000001000000 -0.0000000005831737
0.093000000000000 -0.0000000005831737 8.649000001000035
Pneg(:,84) =
1.0e+004 *
0.000100000100000      0 0.0247000000000000
      0 0.000100000100000 -0.0000000000688133
0.024700000000000 -0.0000000000688133 6.100900000100006
Pneg(:,85) =

```

1.0e+003 *
0.001000001000000 0 0.0340000000000000
0 0.001000001000000 -0.000000003865248
0.034000000000000 -0.000000003865248 1.156000001000015

ans =

0	2.862250000010000e+005
1	2.924100000100006e+004
1.000054454111827e-006	3.600000001000004e+003
1.000145471663339e-006	1.440000010000387e+002
1.000026143202805e-006	1.368900000100008e+004
1.000022663816500e-006	4.000001000036787
1.000182430222860e-006	1.000000010000883e+002
1.000006009565115e-006	6.250000010002309e+002
1.000024857455763e-006	1.000001000092538
1.000092147240103e-006	3.385600000100004e+004
1.000033051503041e-006	1.000067376312542e-006
1.000144592621953e-006	2.656900000100005e+004
1.000103441260944e-006	7.128900000099999e+004
1.000041703845271e-006	3.696640000010000e+005
1.000001235448497e-006	9.000000010001237e+002
5.808100000100000e+004	1.368900000100002e+004
1.017610000010000e+00	3.600000001000063e+003
8.940100000099999e+004	1.089000001000000e+003
1.768900000100000e+004	1.000000010000283e+002
6.561000001000088e+003	3.136000001000064e+003
1.049760000010000e+005	4.761000001000004e+003
8.281000001000050e+003	1.123600000100005e+004
1.040400000100009e+004	5.776000001000017e+003
2.689600000100004e+004	5.776000001000000e+003
8.643600000100005e+004	3.841600000100002e+004
3.204100000100007e+004	6.451600000100002e+004
5.616900000100012e+004	2.310400000100003e+004
2.007040000010001e+005	2.496400000100000e+004
2.500000001000145e+003	8.940100000099999e+004
1.040400000100003e+004	2.704000001000001e+003
3.249000001000023e+003	1.936000001000013e+003
2.280100000100018e+004	2.916000001000041e+003
8.820900000099999e+004	64.000001000008822
7.921000001000025e+003	7.921000001000031e+003
1.040400000100009e+004	1.768900000100008e+004
2.689600000100003e+004	4.624000001000025e+003
6.656400000100014e+004	4.326400000100003e+004
2.401000001000103e+003	4.840000000100001e+004
8.410000010000417e+002	6.864400000099999e+004

Phasil =

0.000001001730117 0.000000113683755 0.000000029411765
0.000000113683755 1.000000999999987 -0.000000000000003
0.000000029411769 -0.000000000000003 0.000001000000148

xhasil =

0.100490773942475
0.100490773942475
3.416591154860994

tampilkan data model Y hasil? =1

Ytt =

1.0e+002 *

4.259572932980635	2.007883893976522
1.849572925601333	1.489785359852356
1.849572970443226	1.742358067446440
1.849572943376813	1.726908903224663
1.849572944884716	1.652329958898930
1.849573012680330	1.535649508772801
1.849572986199737	1.885677820254802
1.849572997911903	1.857814745055431
1.849573017576910	1.214689538926427
1.849572986242208	1.655976687816164
1.849572998493239	1.925969499387884
1.849573006952839	1.857098856274183
1.849573035049243	1.705379578294069
1.849573005661353	1.877125860195539
2.090765304265617	1.869523279652540
2.168865927148464	1.887377273244896
2.148862848380470	1.867114576768402
1.982708336586004	2.078451962210563
1.930666271176084	1.844660838806834
2.173954211984573	2.026520240578189
1.940684634167329	2.197832742893219
1.951707876138458	2.537648292190046
2.013799365909774	1.810449797430956
2.144014096918178	1.719251293113764
1.608573048991874	1.795091446895067
1.849573079940996	1.886191918285545
2.114884551665872	1.837533896994940
2.200795282988710	1.908872663962712
2.178791915645312	1.925630092669951
1.996021927177215	1.714115054882534
1.938775688671852	1.933468725637073
2.206392475736317	1.749322866476886
1.949795886273949	1.606404238065498
1.961921416449288	1.508153429094924
2.030222074563895	1.684764810197388
2.173458270731128	2.035761740346755
2.046788421969426	2.185756104619550
2.110707409788940	1.903689773924749
2.319080623094451	1.899170269461110
1.872740261406418	1.784314368380008
1.707244907705947	1.851485730394156

1.899074681522372

1.761901733392123

Error: 0e-002

1.849572932980635
 -1.340427074398668
 -1.140427029556774
 0.519572943376813
 1.039572944884716
 -1.390426987319670
 0.939572986199737
 0.829572997911903
 0.209573017576910
 -1.090427013757793
 0.059572998493239
 -0.520426993047161
 -5.040426964950757
 -1.840426994338647
 0.120765304265617
 0.268865927148464
 -0.171137151619530
 1.712708336586004
 1.910666271176084
 2.173954211984573
 1.940684634167329
 1.591707876138458
 -0.266200634090226
 -0.515985903081822
 0.068573048991874
 -0.130426920059004
 -0.455115448334128
 0.180795282988710
 1.028791915645312
 1.746021927177215
 1.818775688671852
 1.956392475736317
 1.939795886273949
 -0.238078583550712
 -0.249777925436105
 -2.116541729268872
 -2.163211578030574
 -5.949292590211059
 0.049080623094451
 1.022740261406418
 1.157244907705947
 1.319074681522372

1.987883893976522
 0.679785359852356
 1.042358067446440
 0.586908903224663
 -1.387670041101070
 -1.994350491227199
 -0.364322179745198
 -3.662185254944569
 0.464689538926427
 -0.774023312183836
 -1.614030500612116
 0.757098856274183
 1.245379578294069
 1.607125860195539
 1.089523279652540
 1.637377273244896
 0.157114576768402
 -0.771548037789437
 -2.485339161193167
 -1.293479759421811
 -1.172167257106782
 0.437648292190046
 -0.819550202569044
 0.649251293113764
 1.005091446895067
 -0.013808081714455
 0.127533896994941
 -0.811127336037288
 -0.124369907330049
 -2.415884945117466
 -0.596531274362927
 -0.730677133523114
 -0.223595761934502
 0.298153429094924
 0.874764810197388
 1.445761740346756
 1.665756104619550
 1.903689773924749
 1.149170269461110
 1.254314368380008
 -3.268514269605844
 -0.378098266607877

Norm err:

0.143193924833538



KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI (UIN)
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No. 50 Dinoyo Malang (0341)551345 Fax. (0341) 572533

BUKTI KONSULTASI SKRIPSI

Nama : Deny Fatchur Rochman
NIM : 14610082
Fakultas/ Jurusan : Sains dan Teknologi/ Matematika
Judul Skripsi : Estimasi Parameter Model *Seasonal Autoregressive Integrated Moving Average* dengan Metode Kalman Filter
Pembimbing I : Abdul Aziz, M.Si
Pembimbing II : Ari Kusumastuti, M.Pd, M.Si

No	Tanggal	Hal	Tanda Tangan	
1	13 Juli 2018	Revisi Bab I	1.	
2	18 Juli 2018	Revisi Bab I & II		2.
3	20 Juli 2018	Revisi Bab II	3.	
4	26 Juli 2018	Revisi Bab III		4.
5	31 Juli 2018	Revisi Bab IV	5.	
6	05 Agustus 2018	Revisi Kajian Agama Bab I & II		6.
7	13 Agustus 2018	Revisi Bab IV	7.	
8	20 Agustus 2018	Revisi Bab IV		8.
9	04 September 2018	Revisi Bab IV	9.	
10	17 Oktober 2018	ACC untuk diseminarkan		10.
11	18 Oktober 2018	ACC untuk diseminarkan	11.	
12	26 Oktober 2018	Revisi Bab IV		12.
13	18 Januari 2019	Revisi Bab IV	13.	
14	08 Maret 2019	Revisi Bab IV & V		14.
15	05 April 2019	Revisi Kajian Agama Bab II	15.	
16	06 April 2019	Revisi Kajian Agama Bab IV		16.
17	08 April 2019	ACC Kajian Agama Keseluruhan	17.	
18	03 Mei 2019	Revisi Bab IV & V		18.
19	03 Mei 2019	Revisi Abstrak	19.	
20	06 Mei 2019	ACC Keseluruhan		20.

Malang, 06 Mei 2019
Mengetahui,
Ketua Jurusan Matematika

Dr. Usman Pagalay, M.Si
NIP. 19650414 200312 1 001