

**PEMODELAN HARGA MINYAK SAWIT MENTAH
DENGAN METODE ARIMA-GARCH**

PROPOSAL TUGAS AKHIR

OLEH:

name student

NO. BP.



PROGRAM STUDI MAGISTER MATEMATIKA

JURUSAN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS of ...

city

2022

DAFTAR ISI

DAFTAR ISI	1
DAFTAR GAMBAR	4
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	1
1.3 Tujuan Penelitian	1
1.4 Sistematika Penulisan	1
BAB II LANDASAN TEORI	2
2.1 Minyak Sawit Mentah.....	2
2.2 Faktor yang Mempengaruhi Harga Minyak Sawit Men- tah	3
2.3 Data Deret Waktu.....	3
2.3.1 Pola Data	4
2.3.2 Stasioneritas Data	6
2.3.3 Uji Kestasioneran.....	10
2.4 Autocorrelation Function (ACF) dan Partial Autocor- relation Function (PACF)	12
2.5 Proses White Noise	19

2.6	Model-Model Deret Waktu.....	21
2.6.1	<i>Auto-regressive</i> (AR)	21
2.6.2	<i>Moving Average</i> (MA).....	22
2.6.3	<i>Auto-regressive Moving Average</i> (ARMA)	23
2.6.4	<i>Auto-regressive Integrated Moving Average</i> (ARIMA)	
	24	
2.6.5	<i>Autoregressive Condittional heteroskedasticity</i> (ARCH)	
	dan <i>General Auto Regressive Conditional Heteroskedeas-</i>	
	<i>ticity</i> (GARCH)	25
2.7	Estimasi Parameter dari Model Regresi GARCH	28
2.7.1	Uji Q-Ljung Box	32
2.8	Fuzzy Time Series	35
2.8.1	Himpunan Fuzzy	35
2.8.2	Rantai Markov (<i>Markov Chain</i>).	37
2.8.3	Proses awal Pemodelan FTS	39
2.8.4	FTS Chen	43
2.8.5	FTS Lee	45
2.8.6	FTS Tsaur	46
2.9	Keakuratan Model.....	48
BAB III	METODE PENELITIAN	50
3.1	Jenis dan Sumber Data.....	50
3.2	Teknik Pengolahan dan Analisis Data	50

DAFTAR PUSTAKA	51
----------------------	----

DAFTAR GAMBAR

Gambar 2.3.1	Plot Data Berpola Horizontal	4
Gambar 2.3.2	Plot Data Berpola Trend	5
Gambar 2.3.3	Plot Data Berpola Musiman	5
Gambar 2.3.4	Plot Data Berpola Siklis	6

BAB I

PENDAHULUAN

- 1.1 Latar Belakang**
- 1.2 Rumusan Masalah**
- 1.3 Tujuan Penelitian**
- 1.4 Sistematika Penulisan**

BAB II

LANDASAN TEORI

Bab ini berisikan tentang teori-teori yang terkait dengan pembahasan tentang model ARIMA-GARCH, model hybrid ARIMA-FTS Chen, ARIMA-FTS Lee dan ARIMA-FTS Tsaur dalam memodelkan harga minyak sawit mentah (CPO).

2.1 Minyak Sawit Mentah

Minyak Sawit Mentah (CPO) adalah minyak mentah yang berwarna kemerah-merahan yang diperoleh dari hasil ekstraksi atau dari proses pengempaan daging buah kelapa sawit. Produk CPO ini mencakupi banyak bidang, diantaranya pada industri sabun sebagai bahan penghasil busa, industri baja sebagai bahan pelumas, industri pangan sebagai minyak goreng, margarin, shortening dan vegetable ghee, industri oleokimia sebagai fatty acids, fatty alcohol dan glycesrin serta biodisel, dan juga pada industri lainnya.

Dengan banyaknya manfaat CPO tadi, CPO bisa dikatakan salah satu faktor penggerak perekonomian dunia, terutama Indonesia. Hal ini dikarenakan Indonesia merupakan negara penghasil minyak sawit mentah tertinggi di dunia.

Sebagai catatan, 85

2.2 Faktor yang Mempengaruhi Harga Minyak Sawit Mentah

Faktor-faktor yang dapat mempengaruhi harga minyak mentah adalah sebagai berikut:

1. Harga Minyak mentah lainnya, Harga minyak bumi mentah, minyak kedelai mentah berbanding lurus dengan harga minyak sawit mentah. Jika minyak mentah lainnya naik, maka minyak sawit mentah juga akan naik, begitupun sebaliknya.
2. Harga minyak mentah dihargai dalam dolar Amerika Serikat. Ketika nilai dolar AS turun, harga minyak mentah akan mengalami kenaikan dan begitupun sebaliknya.
3. Permintaan pasar, Semakin tinggi permintaan pasar, semakin tinggi juga harga minyak sawit mentah, begitupun sebaliknya

2.3 Data Deret Waktu

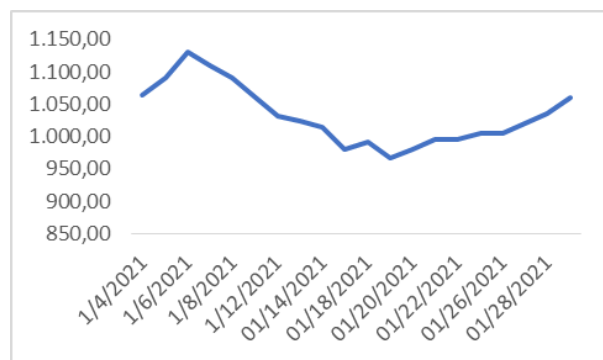
Data deret waktu merupakan sekumpulan nilai suatu variabel yang dikumpulkan secara berkala pada waktu tertentu. Waktu tertentu yang dimaksud bisa dalam harian, mingguan, bulanan, tahunan, dan lain-lain. Data deret waktu menggambarkan aktivitas data dari waktu ke waktu.

2.3.1 Pola Data

Data deret waktu dapat diklasifikasi berdasarkan pola datanya. Pola data ini bertujuan untuk memberikan gambaran kepada peneliti-peneliti tentang model yang akan digunakan. Berikut dipaparkan beberapa klasifikasi pola data, yaitu:

1. Pola Data Horizontal

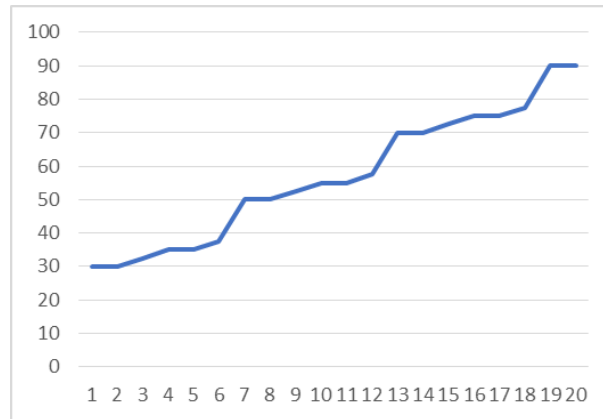
Pola data horizontal terjadi apabila data pengamatan bergerak secara acak, namun menyerupai bentuk garis horizontal. Data berpola horizontal disebut juga dengan data stasioner. Bentuk plot data horizontal dapat dilihat pada Gambar 2.3.1 berikut:



Gambar 2.3.1: Plot Data Berpola Horizontal

2. Pola Data Trend

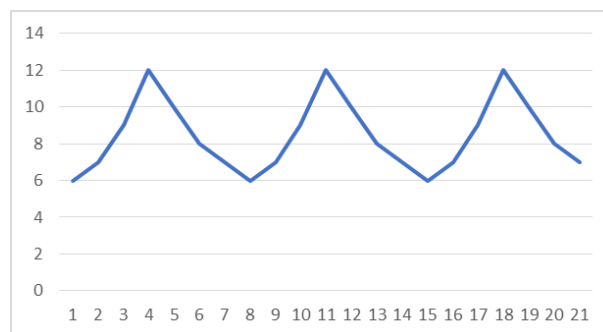
Pola data trend terjadi apabila data pengamatan bergerak secara acak, namun menyerupai bentuk garis miring selama periode yang lama. Pola data trend disebut dengan data tidak stasioner. Bentuk plot data trend dapat dilihat pada Gambar 2.3.2 berikut



Gambar 2.3.2: Plot Data Berpola Trend

3. Pola Data Musiman

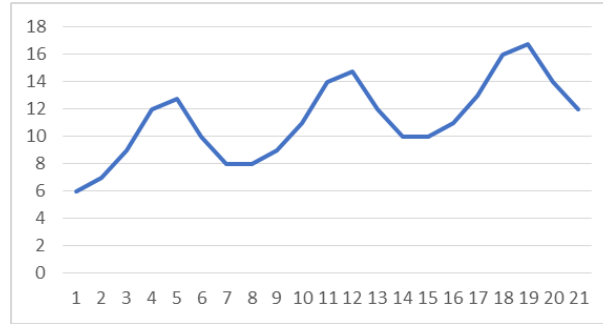
Pola data musiman terjadi apabila data pengamatan bergerak secara acak namun ada pergerakan yang selalu sama dan berulang dalam selang waktu berbeda. Artinya apabila data tersebut dikelompokkan menjadi beberapa bagian, maka pergerakan data tersebut sama. Hal ini biasanya dipengaruhi oleh faktor-faktor musim yang berulang, dari satu periode ke periode berikutnya. Bentuk pola data musiman adalah sebagai berikut:



Gambar 2.3.3: Plot Data Berpola Musiman

4. Pola Data Siklis

Pola data siklis terjadi apabila data pengamatan bergerak di sekitar garis trend. Hal ini biasanya dipengaruhi oleh fluktuasi ekonomi jangka panjang. Pola data siklis terlihat pada Gambar 2.3.4 berikut:



Gambar 2.3.4: Plot Data Berpola Siklis

2.3.2 Stasioneritas Data

Setelah tahap awal yakni pola data diketahui berdasarkan plotnya, tahap berikutnya adalah memeriksa stasioneritas data. Data deret waktu dinyatakan stasioner apabila data menggambarkan pola yang tetap dari waktu ke waktu. Dengan kata lain, tidak terdapat kenaikan atau penurunan pada data yang terlalu signifikan. Stasioneritas data, diklasifikasikan menjadi 2 bentuk, yaitu:

1. Stasioner terhadap Nilai Tengah

Data deret waktu dikatakan stasioner terhadap nilai tengah, apabila data bergerak di sekitar nilai tengah selama waktu pengamatan. Misalkan X_t adalah data deret waktu pada saat t . Data deret waktu X_t dikatakan stasioner pada nilai tengah jika nilai $E(X_t) = \mu$, dimana nilai μ merupakan nilai tengah yang tidak bergantung terhadap t . Apabila data deret waktu tidak stasioner terhadap nilai tengah maka dilakukan *differencing* agar data menjadi stasioner terhadap nilai tengah.

Differencing merupakan proses menghitung nilai selisih dari data pengamatan sampai data stasioner. Artinya setelah melakukan differencing pertama, nilai selisih dari data pengamatan yang didapatkan diperiksa apakah stasioner atau tidak. Jika belum stasioner, maka dilakukan *differ-*

encing kembali hingga stasioner.

Misalkan X_t adalah data deret waktu pada saat t yang tidak stasioner.

Data deret waktu X_t dilakukan differencing orde pertama dinyatakan sebagai berikut:

$$\nabla^1 X_t = X_t - X_{t-1} \quad (2.3.1)$$

dimana

$\nabla^1 X_t$: *differencing* orde pertama pada nilai X ke t

X_t : nilai ke t

X_{t-1} : nilai ke $t - 1$

Metode *differencing* yang digunakan adalah metode operator *shift* mundur (*backward shift*), yang dinotasikan dengan B . Penggunaan operator B adalah sebagai berikut:

$$BX_t = X_{t-1} \quad (2.3.2)$$

Operator B pada X_t , mempunyai pengaruh dalam menggeser data satu periode ke belakang. Penerapan dua operator B pada X_t , akan menggeser data tersebut dua periode ke belakang seperti berikut:

$$B(BX_t) = B^2 X_t = X_{t-2} \quad (2.3.3)$$

Persamaan (2.3.1) dapat ditulis menggunakan operator *shift* mundur yaitu:

$$\begin{aligned} \nabla^1 X_t &= X_t - X_{t-1} \\ &= X_t - BX_t \\ &= (1 - B)X_t \end{aligned} \quad (2.3.4)$$

Berdasarkan Persamaan (2.3.4), diperoleh *differencing* orde pertama dapat dinyatakan oleh $(1 - B)$. Jika data belum stasioner, maka dilakukan *differencing* orde kedua yang dapat dihitung sebagai berikut:

$$\begin{aligned}
\nabla^2 X_t &= \nabla^1 X_t - \nabla^1 X_{t-1} \\
&= (X_t - X_{t-1}) - (X_{t-1} - X_{t-2}) \\
&= X_t - X_{t-1} - X_{t-1} + X_{t-2} \\
&= X_t - 2X_{t-1} + X_{t-2} \tag{2.3.5} \\
&= X_t - 2BX_t + B^2X_t \\
&= (1 - 2B + B^2)X_t \\
&= (1 - B)^2X_t
\end{aligned}$$

Pada Persamaan (2.3.5) *differencing* orde kedua dapat dinyatakan oleh $(1 - B)^2$. Jika data masih belum stasioner, maka dilakukan *differencing* orde ketiga yang dapat dihitung sebagai berikut:

$$\begin{aligned}
\nabla^3 X_t &= \nabla^2 X_t - \nabla^2 X_{t-1} \\
&= \nabla^1 X_t - \nabla^1 X_{t-1} - (\nabla^1 X_{t-1} - \nabla^1 X_{t-2}) \\
&= \nabla^1 X_t - \nabla^1 X_{t-1} - \nabla^1 X_{t-1} + \nabla^1 X_{t-2} \\
&= \nabla^1 X_t - 2\nabla^1 X_{t-1} + \nabla^1 X_{t-2} \\
&= (X_t - X_{t-1}) - 2(X_{t-1} - X_{t-2}) + (X_{t-2} - X_{t-3}) \\
&= X_t - X_{t-1} - 2X_{t-1} + 2X_{t-2} + X_{t-2} - X_{t-3} \\
&= X_t - 3X_{t-1} + 3X_{t-2} - X_{t-3} \\
&= X_t - 3BX_t + 3B^2X_t - B^3X_t \\
&= (1 - 3B + 3B^2 - B^3)X_t \\
&= (1 - B)^3X_t
\end{aligned} \tag{2.3.6}$$

Operator $(1 - B)^3$ menyatakan *differencing* orde ketiga pada data. Tujuan melakukan *differencing* untuk mencapai stasioneritas terhadap nilai tengah. Secara umum, apabila terdapat *differencing* orde ke d untuk mencapai stasioneritas maka dapat dinyatakan sebagai berikut:

$$\nabla^d X_t = (1 - B)^d X_t \tag{2.3.7}$$

2. Stasioner terhadap Ragam

Data deret waktu dikatakan stasioner terhadap ragam, apabila data berfluktuasi dengan ragam yang konstan selama waktu pengamatan. Misalkan X_t adalah data deret waktu pada saat t . Data deret waktu X_t dikatakan stasioner terhadap ragam, jika $Var(X_t) = Var(X_{t+k}) = \sigma^2; k = 1, 2, \dots$, dengan nilai ragam tidak bergantung pada t . Jika pada data asli agar

stasioner terhadap ragam.

Transformasi *Box-Cox* adalah salah satu metode yang digunakan untuk menstasionerkan data terhadap ragam. Misalkan $T(X_t)$ adalah fungsi transformasi dari X_t . Jika X_t belum stasioner terhadap ragam, maka dapat ditransformasi dengan formula berikut:

$$T(X_t) = \begin{cases} \frac{X_t^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ \ln(X_t), & \lambda = 0 \end{cases} \quad (2.3.8)$$

dengan λ disebut parameter transformasi. Berdasarkan Persamaan (2.3.8), diperoleh hasil transformasi nilai X_t dari beberapa nilai λ yang umum digunakan seperti pada tabel berikut:

Tabel 2.3.1 : Bentuk Transformasi

λ	Bentuk Transformasi
-1	$\frac{1}{X_t}$
-0,5	$\frac{1}{\sqrt{X_t}}$
0	$\ln(X_t)$
0,5	$\sqrt{X_t}$
1	X_t (tidak ditransformasikan)

2.3.3 Uji Kestasioneran

Langkah-langkah yang dilakukan untuk pengujian stasioneritas suatu data adalah sebagai berikut:

1. Analisis Grafik

Langkah awal yang dilakukan untuk melihat kestasioneran suatu data deret waktu adalah analisis grafik. Pada grafik tersebut dibuat plot antara data pengamatan dengan waktu. Apabila data tersebut berfluktuasi di sekitar nilai tengah dan ragam yang konstan maka dapat disimpulkan data tersebut stasioner. Akan tetapi, langkah analisis grafik ini memiliki kelemahan yaitu bersifat subjektif dalam mengambil keputusan apakah data tersebut stasioner atau tidak. Hal ini mengakibatkan terjadinya perbedaan dalam pengambilan keputusan.

2. Uji *Augmented Dickey-Fuller* (ADF)

Uji akar unit merupakan salah satu uji formal untuk menguji kestasioneran terhadap nilai tengah pada deret waktu. Uji akar unit yang sering digunakan adalah uji *Augmented Dickey-Fuller* (ADF). Uji ADF ini dikenalkan oleh David Dickey dan Wayne Fuller. Pengujian dengan *Augmented Dickey-Fuller* (ADF) disajikan dalam bentuk persamaan regresi sebagai berikut:

$$\nabla X_t = \mu + \delta X_{t-1} + \sum_{i=1}^k \phi_i \nabla X_{t-i} + \varepsilon_t \quad (2.3.9)$$

dengan $\delta = \rho - 1$, $\nabla X_t = X_t - X_{t-1}$, k adalah jumlah *lag*, ϕ dan μ adalah parameter model, serta t adalah waktu pengamatan. Hipotesis yang digunakan dalam uji ADF yaitu:

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

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

Statistik uji *Augmented Dickey-Fuller* (ADF) yang digunakan adalah sebagai berikut :

$$ADF = \frac{\delta}{SE(\delta)} \quad (2.3.10)$$

Dimana $SE(\delta)$ merupakan standard error untuk δ . Klasifikasi pengambilan keputusan sebagai berikut :

- Jika nilai kritis tabel-t ADF \geq statistik uji ADF, maka tolak H_0 , hal ini menandakan bahwa data stationer
- Jika nilai kritis tabel-t ADF $<$ statistik uji ADF, maka tolak H_0 , hal ini menandakan bahwa data belum stationer

2.4 Autocorrelation Function (ACF) dan Partial Autocorrelation Function (PACF)

Autocorrelation Function (ACF) adalah fungsi yang menunjukkan besarnya korelasi antara pengamatan pada waktu ke- t , dengan pengamatan waktu sebelumnya. Partial Autocorrelation Function (PACF) adalah fungsi yang menunjukkan besarnya korelasi parsial antara pengamatan pada waktu ke- t , dengan pengamatan waktu sebelumnya. Fungsi ACF dan PACF ini digunakan untuk mengidentifikasi model dari data deret waktu. Berikut ini diberikan definisi dari fungsi autokovarian dan fungsi autokorelasi pada data deret waktu stationer.

Definisi 2.4.1 Misalkan X_t adalah data deret waktu stationer, maka

fungsi autokovarian (ACVF) dari X_t pada lag k adalah

$$\gamma_k = Cov(X_t, X_{t+k}) = E[(X_t - \mu)(X_{t+k} - \mu)] \quad (2.4.1)$$

Fungsi autokorelasi (ACF) dari X_t adalah

$$\rho_k = Corr(X_t, X_{t+k}) = \frac{\gamma_k}{\gamma_0} \quad (2.4.2)$$

Fungsi autokovarian dan fungsi autokorelasi biasanya tidak diketahui pada data populasi. Hal ini, dapat diduga menggunakan fungsi autokovarian dan autokorelasi sampel. Fungsi autokovarian untuk sampel X_1, X_2, \dots, X_n yaitu sebagai berikut:

$$\hat{\gamma}_k = \frac{1}{n} \sum_{t=1}^{n-k} (X_t - \hat{X})(X_{t+k} - \hat{X}) \quad (2.4.3)$$

dan fungsi autokorelasi untuk sampel X_1, X_2, \dots, X_n yaitu :

$$\hat{\rho}_k = \frac{\sum_{t=1}^{n-k} (X_t - \hat{X})(X_{t+k} - \hat{X})}{\sum_{t=1}^n (X_t - \hat{X})^2} \quad (2.4.4)$$

dimana

$\hat{\gamma}_k$: koefisien autokovarian lag ke- k untuk $k = 1, 2, 3, \dots$

$\hat{\rho}_k$: koefisien autokorelasi lag ke- k untuk $k = 1, 2, 3, \dots$

n : jumlah data pengamatan

X_t : nilai X ke- t

\bar{X} : nilai tengah X .

Fungsi autokorelasi parsial digunakan untuk mengukur korelasi antara X_t dan X_{t+k} , setelah data $X_{t+1}, X_{t+2}, \dots, X_{t+k-1}$ dihilangkan. Fungsi autokorelasi parsial antara X_t dengan X_{t+k} didefinisikan sebagai berikut :

$$\phi_{kk} = \text{Corr}(X_t, X_{t+k} | X_{t+k-1}, X_{t+k-2}, \dots, X_{t+1}; k \geq 1 \quad (2.4.5)$$

Fungsi autokorelasi parsial biasanya tidak diketahui pada data populasi. Hal ini, dapat diduga dengan menggunakan fungsi autokorelasi parsial sampel yang dinyatakan sebagai berikut:

$$\hat{\phi}_{k+1,k+1} = \frac{\hat{\rho}_{k+1} - \sum_{j=1}^k \hat{\phi}_k, j \hat{\rho}_{k+1-j}}{1 - \sum_{j=1}^k \hat{\phi}_k, j \hat{\rho}_j} \quad (2.4.6)$$

dan

$$\hat{\phi}_{k+1,j} = \hat{\phi}_{kj} - \hat{\phi}_{k+1,k+1} + \hat{\phi}_{k,k+1-j}; j = 1, 2, \dots, k$$

Fungsi autokorelasi parsial antara data X_t dan X_{t+k} pada Persamaan (2.4.5) diuraikan sebagai berikut:

Misalkan data X_t adalah data yang stasioner dengan $E(X_t) = 0$. Data X_{t+k} dapat dinyatakan sebagai model linier yaitu:

$$X_{t+k} = \hat{\phi}_{kj} X_{t+k-1} + \hat{\phi}_{k2} X_{t+k-2} + \dots + \hat{\phi}_{kk} X_{t+\varepsilon t+k} \quad (2.4.7)$$

dengan $\hat{\phi}_{ki}$ adalah parameter model ke- i dan ε_{t+k} adalah residu yang tidak berkorelasi dengan X_{t+k-j} untuk $j = 1, 2, \dots, k$. Langkah awal yang dilakukan yaitu mengalikan Persamaan (2.4.7), dengan X_{t+k-j} pada kedua ruas sehingga diperoleh

$$X_{t+k} X_{t+k-j} = \hat{\phi}_{k1} X_{t+k-1} X_{t+k-j} + \hat{\phi}_{k2} X_{t+k-2} X_{t+k-j} + \dots + \hat{\phi}_{kk} X_t X_{t+k-j} + \varepsilon_{t+k} X_{t+k-j} \quad (2.4.8)$$

nilai harapan dari $X_{t+k}X_{t+k-j}$ adalah sebagai berikut:

$$E(X_{t+k}X_{t+k-j}) = \hat{\phi}_{k1}E(X_{t+k-1}X_{t+k-j}) + \hat{\phi}_{k2}E(X_{t+k-2}X_{t+k-j}) + \dots + \hat{\phi}_{kk}E(X_tX_{t+k-j}) + E(\varepsilon_{t+k}X_{t+k-j}) \quad (2.4.9)$$

dengan memisalkan nilai $E(X_{t+k}X_{t+k-j}) = \gamma_j$ untuk $j = 0, 1, \dots, k$ dan diketahui

$E(\varepsilon_{t+k}X_{t+k-j}) = 0$, diperoleh

$$\gamma_j = \phi_{k1}\gamma_{j-1} + \phi_{k2}\gamma_{j-2} + \dots + \phi_{kk}\gamma_{j-k} \quad (2.4.10)$$

diketahui $\rho_j = \frac{\gamma_j}{\gamma_0}$ untuk $j = 0, 1, \dots, k$. Persamaan (2.4.10) dibagi dengan γ_0

sehingga diperoleh

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

untuk $j = 1, 2, \dots, k$, dengan mensubstitusikan $\rho_k = \rho_{-k}$ diperoleh sistem

persamaan linear sebagai berikut:

$$\begin{aligned} \rho_1 &= \rho_0\phi_{k1} + \rho_1\phi_{k2} + \rho_2\phi_{k3} + \dots + \rho_{k-1}\phi_{kk} \\ \rho_2 &= \rho_1\phi_{k1} + \rho_0\phi_{k2} + \rho_1\phi_{k3} + \dots + \rho_{k-2}\phi_{kk} \\ \rho_3 &= \rho_2\phi_{k1} + \rho_1\phi_{k2} + \rho_0\phi_{k3} + \dots + \rho_{k-3}\phi_{kk} \\ &\vdots \\ \rho_k &= \rho_{k-1}\phi_{k1} + \rho_{k-2}\phi_{k2} + \rho_{k-3}\phi_{k3} + \dots + \rho_0\phi_{kk} \end{aligned} \quad (2.4.11)$$

Diketahui $\rho_0 = 1$ sehingga Persamaan (2.4.11) dapat dituliskan dalam notasi matriks, yaitu

$$\begin{bmatrix} 1 & \rho_1 & \cdots & \rho_{k-1} \\ \rho_1 & 1 & \cdots & \rho_{k-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} \phi_{k1} \\ \phi_{k2} \\ \vdots \\ \phi_{kk} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_k \end{bmatrix} \quad (2.4.12)$$

Persamaan 2.4.12 dapat diselesaikan dengan menggunakan aturan *Cramer*.

Penjelasan tentang aturan *Cramer* pada sistem persamaan linier diberikan oleh Teorema 2.4.1 berikut.

Teorema 2.4.1. *Jika $Ax = b$ adalah suatu sistem dari n persamaan linier dengan n faktor yang tidak diketahui sedemikian rupa sehingga $\det(A) \neq 0$, maka sistem ini memiliki solusi yang unik. Solusinya adalah*

$$x_1 = \frac{\det(A_1)}{\det(A)}, x_2 = \frac{\det(A_2)}{\det(A)}, \dots, x_n = \frac{\det(A_n)}{\det(A)}$$

dimana A_j adalah matriks yang diperoleh dengan mengganti entri-entri pada kolom ke- j dari A dengan entri-entri pada matriks

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Persamaan (2.4.12) dapat diselesaikan menggunakan Teorema 2.4.1 tentang aturan *Cramer*, dengan

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

untuk $k = 1, 2, \dots$ digunakan untuk mencari nilai fungsi autokorelasi par-

sial lag k yaitu $\phi_{k1}, \phi_{k2}, \dots, \phi_{kk}$.

1. Lag pertama dengan $k = 1$

. Fungsi autokorelasi parsial pada lag pertama, akan sama dengan fungsi autokorelasi pada lag pertama. Hal ini dikarenakan, sistem persamaan

$\rho_1 = \phi_{11}\rho_0$ dengan $k = 1$ dan $\rho_0 = 1$ sehingga diperoleh

$$\phi_{11} = \rho_1 \quad (2.4.13)$$

2. Lag kedua dengan $k = 2$.

Fungsi autokorelasi parsial pada lag kedua diperoleh sistem persamaan yaitu:

$$\begin{aligned} \rho_1 &= \phi_{11}\rho_0 + \phi_{22}\rho_0 \\ \rho_2 &= \phi_{11}\rho_1 + \phi_{22}\rho_1 \end{aligned} \quad (2.4.14)$$

dan dapat ditulis dalam bentuk matriks, yaitu:

$$\begin{bmatrix} \rho_0 & \rho_1 \\ \rho_1 & \rho_0 \end{bmatrix} \begin{bmatrix} \phi_{11} \\ \phi_{12} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix}$$

dengan menggunakan aturan Cramer diperoleh,

$$\phi_{22} = \frac{\det(A_2)}{\det(A)} = \frac{\begin{vmatrix} 1 & \rho_1 \\ \rho_1 & \rho_2 \end{vmatrix}}{\begin{vmatrix} \rho_0 & \rho_1 \\ \rho_1 & \rho_0 \end{vmatrix}} = \frac{\rho_2 - \rho_1^2}{1 - \rho_1^2} \quad (2.4.15)$$

3. Lag ketiga dengan $k = 3$

Fungsi autokorelasi parsial pada lag ketiga diperoleh sistem persamaan yaitu:

$$\begin{aligned} \rho_1 &= \phi_{11}\rho_0 + \phi_{22}\rho_1 + \phi_{33}\rho_2 \\ \rho_2 &= \phi_{11}\rho_1 + \phi_{22}\rho_0 + \phi_{33}\rho_1 \end{aligned}$$

$$\rho_3 = \phi_{11}\rho_2 + \phi_{22}\rho_1 + \phi_{33}\rho_0 \quad (2.4.16)$$

dapat ditulis dalam bentuk matriks, yaitu:

$$\begin{bmatrix} \rho_0 & \rho_1 & \rho_2 \\ \rho_1 & \rho_0 & \rho_1 \\ \rho_2 & \rho_1 & \rho_0 \end{bmatrix} \begin{bmatrix} \phi_{11} \\ \phi_{12} \\ \phi_{13} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{bmatrix}$$

dengan menggunakan aturan Cramer diperoleh,

$$\begin{aligned} \phi &= \frac{\det(A_3)}{\det(A)} = \frac{\begin{vmatrix} 1 & \rho_1 & \rho_1 \\ \rho_1 & 1 & \rho_2 \\ \rho_2 & \rho_1 & \rho_3 \end{vmatrix}}{\begin{vmatrix} & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_1 \\ \rho_2 & \rho_1 & 1 \end{vmatrix}} \\ &= \frac{\rho_3 + \rho_2^2\rho_1 + \rho_1^3 - \rho_1\rho_2 - \rho_2\rho_1 - \rho_1^2\rho_3}{1 + \rho_1^2\rho_2 + \rho_1^2\rho_2 - \rho_2^2 - \rho_1^2 - \rho_1^2} \\ &= \frac{\rho_3 - \rho_1^2\rho_3 - \rho_1\rho_2 + \rho_2^2\rho_1 - \rho_2\rho_1 + \rho_1^3}{1 + \rho_1^2\rho_2 + \rho_1^2\rho_2 - \rho_2^2 - \rho_1^2 - \rho_1^2} \\ &= \frac{\rho_3 - \rho_1^2\rho_3 - \rho_1\rho_2(1 - \rho_2) - \rho_2\rho_1 + \rho_1^3}{1 + \rho_1^2\rho_2 + \rho_1^2\rho_2 - \rho_2^2 - \rho_1^2 - \rho_1^2} \\ &= \frac{\rho_3 - \rho_1^2\rho_3 - \rho_1\rho_2(1 - \rho_1^2 - \rho_2 + \rho_1^2) - \rho_2\rho_1 + \rho_1^3}{1 - \rho_1^2 - \rho_1^2 + \rho_1^4 + 2\rho_1^2\rho_2 - \rho_1^4 - \rho_2^2} \\ &= \frac{\rho_3(1 - \rho_1^2) - \rho_1\rho_2(1 - \rho_1^2) + (\rho_2 - \rho_1^2)\rho_1\rho_2 - \rho_1(\rho_2 - \rho_1^2)}{(1 - \rho_1^2) - \rho_1^2(1 - \rho_1^2) + \rho_1^2(\rho_2 - \rho_1^2) - \rho_2(\rho_2 - \rho_1^2)} \\ &= \frac{\rho_3(1 - \rho_1^2) - \rho_1\rho_2(1 - \rho_1^2) + (\rho_2 - \rho_1^2)\rho_1\rho_2 - \rho_1(\rho_2 - \rho_1^2)}{(1 - \rho_1^2)} \\ &= \frac{(1 - \rho_1^2) - \rho_1^2(1 - \rho_1^2) + \rho_1^2(\rho_2 - \rho_1^2) - \rho_2(\rho_2 - \rho_1^2)}{(1 - \rho_1^2)} \\ &= \frac{\rho_3 - \rho_1\rho_2 + (\frac{\rho_2 - \rho_1^2}{1 - \rho_1^2})\rho_1\rho_2 - (\frac{\rho_2 - \rho_1^2}{1 - \rho_1^2})\rho_1}{1 - \rho_1^2 + (\frac{\rho_2 - \rho_1^2}{1 - \rho_1^2})\rho_1^2 - (\frac{\rho_2 - \rho_1^2}{1 - \rho_1^2})\rho_2} \\ &= \frac{\rho_3 - (\rho_1 - (\frac{\rho_2 - \rho_1^2}{1 - \rho_1^2})\rho_1)\rho_2 - (\frac{\rho_2 - \rho_1^2}{1 - \rho_1^2})\rho_1}{1 - (\rho_1 - (\frac{\rho_2 - \rho_1^2}{1 - \rho_1^2})\rho_1)\rho_1 - (\frac{\rho_2 - \rho_1^2}{1 - \rho_1^2})\rho_2} \end{aligned}$$

$$\begin{aligned}
&= \frac{\rho_3 - (\phi_{11} - \phi_{22}\phi_{11})\rho_2 - \phi_{22}\rho_1}{1 - (\phi_{11} - \phi_{22}\phi_{11})\rho_1 - \phi_{22}\rho_1} \\
&= \frac{\rho_3 - \phi_{21}\rho_2 - \phi_{22}\rho_1}{1 - \phi_{21}\rho_1 - \phi_{22}\rho_2} \\
&= \frac{\rho_3 - \sum_{j=1}^2 \phi_{2,j}\rho_{3-j}}{1 - \sum_{j=1}^2 \phi_{2,j}\phi_j}
\end{aligned}$$

4. Lag ke- k , dengan menggunakan aturan Cramer diperoleh fungsi autokorelasi parsial lag ke- k yaitu:

$$\begin{aligned}
\phi_{kk} &= \frac{\begin{vmatrix} 1 & \rho_1 & \cdots & \rho_{k-1} & \rho_1 \\ \rho_1 & 1 & \cdots & \rho_{k-2} & \rho_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \cdots & \rho_1 & \rho_k \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 & \cdots & \rho_{k-2} & \rho_{k-1} \\ \rho_1 & 1 & \cdots & \rho_{k-3} & \rho_{k-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \cdots & \rho_1 & 1 \end{vmatrix}} \\
&= \frac{\rho_k \sum_{j=1}^k \phi_{k-1,j}\rho_j}{1 - \sum_{j=1}^k \sum_{j=1}^k \phi_{k-1,j}\rho_j} \tag{2.4.17}
\end{aligned}$$

dengan ϕ_{kk} disebut *Partial Autocorrelation Function* (PACF) antara X_t dan X_{t+k} . Fungsi ϕ_{kk} menjadi notasi standar untuk PACF antara pengamatan X_t dan X_{t+k} dalam analisis data deret waktu.

2.5 Proses White Noise

Proses *white noise* merupakan proses stasioner yang dianggap sebagai faktor pembangun bagi proses deret waktu lainnya. Proses *white noise* digu-

nakan untuk menjelaskan perilaku data yang acak dan stasioner. Proses *white noise* dinotasikan sebagai berikut :

$$X_t \sim WN(0, \sigma^2) \quad (2.5.1)$$

Suatu data deret waktu X_t disebut *white noise* apabila mempunyai sifat-sifat sebagai berikut:

1. Deretnya terdiri dari peubah acak yang tidak saling berkorelasi.
2. $E(X_t) = 0$ untuk setiap t
3. $Var(X_t) = \sigma^2$ untuk setiap t .
4. $\gamma_k = Cov(X_{t+k}, X_t) = 0$ untuk $k \neq 0$

Karena variabel X_t tidak berkorelasi, maka fungsi autokovariannya yaitu sebagai berikut:

$$\gamma_k = \begin{cases} \sigma^2, & k = 0 \\ 0, & k \neq 0 \end{cases} \quad (2.5.2)$$

Fungsi autokorelasinya yaitu:

$$\rho_k = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases} \quad (2.5.3)$$

Fungsi autokorelasi parsialnya adalah sebagai berikut:

$$\phi_k = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases} \quad (2.5.4)$$

2.6 Model-Model Deret Waktu

Beberapa model deret waktu yang dapat digunakan pada pemodelan data deret waktu adalah sebagai berikut:

2.6.1 *Auto-regressive*(AR)

Model *Auto-regressive* (AR) merupakan model stasioner dari data deret waktu, dimana nilai pengamatan waktu ke- t dipengaruhi oleh nilai pengamatan sebelumnya. Model AR dengan orde p dinotasikan dengan $AR(p)$. Bentuk umum dari model $AR(p)$ adalah:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + \varepsilon_t \quad (2.6.1)$$

dimana :

X_t : data pada periode ke- t , $t = 1, 2, \dots, n$

ϕ_i : parameter AR ke- i , $i = 1, 2, \dots, p$

P : orde dari model AR

ε_t : residu saat t dengan $\varepsilon_t \sim WN(0, \sigma^2)$

dengan menggunakan operator *shift* mundur yaitu $BX_t = X_{t-1}$, Persamaan (2.6.1) dapat ditulis yaitu:

$$\begin{aligned} X_t &= \phi_1 BX_t + \phi_2 B^2 X_t + \dots + \phi_p B^p X_t + \varepsilon_t \\ X_t - \phi_1 BX_t - \phi_2 B^2 X_t - \dots - \phi_p B^p X_t &= \varepsilon_t \\ (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p) X_t &= \varepsilon_t \\ \phi_p (B^p) X_t &= \varepsilon_t \end{aligned} \quad (2.6.2)$$

dengan $\phi_p(B) = (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)$. Model AR(p) memenuhi kondisi stasioner jika total koefisien $\sum_{i=1}^p \phi_i < 1, i = 1, 2, \dots, p$

2.6.2 *Moving Average* (MA)

Model *Moving Average* (MA) merupakan model stasioner dari data deret waktu, dimana nilai pengamatan waktu ke- t dipengaruhi oleh resid pada orde q waktu sebelumnya. Bentuk umum model MA dengan orde q yang dinotasikan dengan MA(q) sebagai berikut:

$$X_t = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} \quad (2.6.3)$$

dimana :

X_t : data pada periode ke- t , $t = 1, 2, \dots, q$

θ_i : parameter MA ke- i , $i = 1, 2, \dots, q$

q : orde dari model MA

ε_t : residu saat t , $\varepsilon_t \sim WN(0, \sigma^2)$

dengan menggunakan operator *shift* mundur yaitu $BX_t = X_{t-1}$, model MA bisa dituliskan sebagai berikut:

$$\begin{aligned} X_t &= \varepsilon_t - \theta_1 B \varepsilon_t - \theta_2 B^2 \varepsilon_t - \dots - \theta_q B^q \varepsilon_t \\ &= (1 - \theta_1 B - \theta_2 B^2 + \dots + \theta_q B^q) \varepsilon_t \\ &= \theta_q(B) \varepsilon_t \end{aligned} \quad (2.6.4)$$

dengan $\theta_q(B) = (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q)$. Model MA(q) akan memenuhi kondisi stasioner jika total koefisien $\sum_{i=1}^q \theta_i < 1$. $i = 1, 2, \dots, q$.

2.6.3 *Auto-regressive Moving Average (ARMA)*

Ketika mengidentifikasi suatu model, adakalanya ditemukan bahwa data mengikuti proses AR sekaligus proses MA. Hal ini berarti, data mengikuti proses *Auto-regressive Moving Average* (ARMA), yaitu model gabungan dari model AR orde p dan model MA orde q . Model umum ARMA yang mempunyai orde (p, q) dapat dinyatakan sebagai berikut:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} \quad (2.6.5)$$

dimana:

X_t : data pada periode ke- $t, t = 1, 2, \dots, n$

θ_i : parameter MA ke- $i, i = 1, 2, \dots, q$

ϕ_i : parameter AR ke- $i, i = 1, 2, \dots, p$

ε_t : residu saat $t, \varepsilon_t \sim WN(0, \sigma^2)$

dengan menggunakan operator shift mundur yaitu $BX_t = X_{t-1}$, Persamaan (2.6.5) dapat ditulis yaitu:

$$(1 - \phi_1 B - \phi_2 B^2 + \dots + \phi_p B^p)X_t = (1 - \theta_1 B - \theta_2 B^2 + \dots + \theta_q B^q)\varepsilon_t \quad (2.6.6)$$

atau dapat ditulis dalam bentuk seperti dibawah ini:

$$\phi_p(B)X_t = \theta_q(B)\varepsilon_t \quad (2.6.7)$$

pada saat memilih orde p dan q pada model ARMA, dapat diidentifikasi dengan melihat plot ACF dan PACF yang disajikan pada tabel berikut:

Tabel 2.6.3: Pola Autokorelasi (ACF) dan Autokorelasi Parsial (PACF)

Model	ACF	ACF
ARMA (p, 0)	Menurun secara bertahap menuju nol	Menuju nol setelah lag ke-p
ARMA (0, q)	Menuju nol setelah lag ke-q	Menurun secara bertahap menuju nol
ARMA (p, q)	Menurun secara bertahap menuju nol	Menurun secara bertahap menuju nol

2.6.4 *Auto-regressive Integrated Moving Average (ARIMA)*

Model *Auto-regressive Integrated Moving Average* (ARIMA) merupakan gabungan dari model AR dan MA setelah dilakukan *differencing*. Dengan kata lain, model ARIMA merupakan model ARMA yang distasionerkan dengan melakukan *differencing* pertama atau lebih (orde d). Bentuk umum model ARIMA (p, d, q) adalah sebagai berikut:

$$(1 - B)^d X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (2.6.8)$$

atau dengan menggunakan operator shift mundur yaitu $BX_t = X_{t-1}$

Persamaan (2.6.39) dapat ditulis sebagai berikut:

$$\phi_p(B)(1 - B)^d X_t = \theta_q(B)\varepsilon_t \quad (2.6.9)$$

dengan

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

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

B : operator shift mundur

$(1 - B)^d X_t$: deret waktu yang stasioner pada *differencing* ke- d .

θ_i : parameter MA ke i , $i = 1, 2, \dots, q$

ϕ_i : parameter AR ke i , $i = 1, 2, \dots, p$

ε_t : residu saat t , $\varepsilon_t \sim WN(0, \sigma^2)$

2.6.5 *Autoregressive Conditional heteroskedasticity* (ARCH) dan *General Auto Regressive Conditional Heteroskedasticity* (GARCH)

Enders menyatakan bahwa banyak data runtun waktu di bidang ekonomi tidak memiliki mean yang konstan dan kebanyakan di antaranya memperlihatkan adanya periode-periode yang relatif tenang (*tranquility*) kemudian diikuti oleh periode-periode yang penuh gejolak (*volatility*) [5]. Kondisi demikian disebut heteroskedastik bersyarat, yaitu jika variansi tak bersyarat / variansi jangka panjang dari data adalah konstan, namun terdapat periode-periode tertentu dimana variansi relatif besar atau sebaliknya. Adanya heteroskedastisitas dalam data mengakibatkan variansi dari estimator menjadi lebih besar, yang selanjutnya akan berakibat pada semakin lebarnya interval konfidensi estimator.

Salah satu model generasi baru yang memfokuskan pemodelan pada variansi adalah model Autoregressive Conditional Heteroscedasticity (ARCH) yang dikenalkan oleh Engle [6].

Heteroskedastisitas dipandang bukan sebagai suatu masalah, tetapi justru memanfaatkan kondisi tersebut untuk membuat model. Bahkan dengan memanfaatkan heteroskedastisitas dalam residu yang tepat, maka akan diperoleh varians yang lebih efisien. Model ini dikenal dengan nama *Autoregressive Conditional heteroskedasticity* (ARCH) dan *General Auto Regressive Conditional Heteroskedasticity* (GARCH). Misalkan terdapat persamaan sebagai berikut:

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + \varepsilon_t \quad (2.6.10)$$

Pada model deret waktu biasa ε_t diasumsikan selalu homoskedastisitas yaitu $E[\varepsilon_t^2] = E[\varepsilon_t^2 | \varepsilon_{t-1}^2, \varepsilon_{t-2}^2, \dots] = \sigma^2$ untuk setiap t . Sedangkan pada model ARCH/GARCH, varian ε_t selalu berubah setiap waktu sehingga $E[\varepsilon_t^2] = E[\varepsilon_t^2 | \varepsilon_{t-1}^2, \varepsilon_{t-2}^2, \dots] = \sigma_t^2$. Varian ε_t heteroskedastisitas dan mengikuti persamaan berikut:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 \quad (2.6.11)$$

dengan α_0 adalah komponen konstanta dan $\alpha_1 \varepsilon_{t-1}^2$ adalah komponen ARCH.

Persamaan (2.6.5.1) merupakan model ARCH dimana σ_t^2 bergantung kepada volatilitas satu periode yang lalu atau disebut juga model ARCH (1). Apabila σ_t^2 bergantung kepada beberapa periode yang lalu, misalkan sebanyak p periode, maka persamaan ARCH(p) menjadi:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \dots + \alpha_p \varepsilon_{t-p}^2 \quad (2.6.12)$$

atau dapat ditulis:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^p \alpha_i \varepsilon_{t-1}^2$$

Namun pada model ARCH seringkali dijumpai jumlah p yang lebih besar sehingga parameter yang akan diestimasi menjadi sangat banyak. Hal ini sering terjadi saat menganalisis data dengan periode waktu harian. Untuk mengatasi masalah tersebut model ARCH dikembangkan menjadi model GARCH dengan model umum GARCH(p,q) yaitu:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^p \alpha_i \varepsilon_{t-1}^2 + \sum_{j=1}^q \beta_j \sigma_{t-1}^2 \quad (2.6.13)$$

dengan $\alpha > 0$; $\alpha_i, \beta_j \geq 0$; dan $\sum_{i=1}^p \alpha_i + \sum_{j=1}^q \beta_j < 1$.

Pendugaan parameter ARCH

Parameter model ARCH diduga dengan menggunakan metode penduga kemungkinan maksimum (*Maximum Likelihood Estimation*). Asumsikan bahwa residual kuadrat menyebar menurut sebaran normal, dan t merupakan banyaknya pengamatan. Maka fungsi likelihood untuk adalah:

$$L = \prod_{t=1}^T \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left(-\frac{1\varepsilon_t^2}{2\sigma_t^2}\right) \quad (2.6.14)$$

Kemudian fungsi (2.6.5.1.1) dapat dituliskan menjadi

$$\begin{aligned} l = \ln(L) &= \ln\left(\prod_{t=1}^T \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left(-\frac{1\varepsilon_t^2}{2\sigma_t^2}\right)\right) \\ &= \sum_{t=1}^T \ln\left(\frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left(-\frac{1\varepsilon_t^2}{2\sigma_t^2}\right)\right) \end{aligned}$$

$$= \sum_{t=1}^T \ln\left(\frac{1}{\sqrt{2\pi\sigma_t^2}}\right) + \sum_{t=1}^T \ln(\exp(-\frac{1\varepsilon_t^2}{2\sigma_t^2}))$$

$$= -\frac{T}{2}\ln 2\pi - \frac{1}{2} \sum_{t=1}^T \ln \sigma_t^2 - \sum_{t=1}^T \frac{\varepsilon_t^2}{2\sigma_t^2}$$

dengan $\sigma_t^2 = \alpha_0 + \sum_{i=1}^p \alpha_i \varepsilon_{t-i}^2$. Parameter $\alpha = (\alpha_0, \alpha_1, \dots, \alpha_p)$ diestimasi

dengan memaksimumkan fungsi likelihood (2.6.5.1.2) yaitu

$$\frac{\partial(-\frac{T}{2}\ln 2\pi - \frac{1}{2} \sum_{t=1}^T \ln \sigma_t^2 - \sum_{t=1}^T (\frac{\varepsilon_t^2}{2\sigma_t^2}))}{\partial \alpha} = 0$$

Misalkan persamaan model ARCH (1) adalah $\sigma_t^2 = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2$ maka fungsi likelihood untuk ε_t^2 adalah

$$l = -\frac{T}{2}\ln 2\pi - \frac{1}{2} \sum_{t=1}^T \ln(\alpha_0 + \alpha_1 \varepsilon_{t-1}^2) - \sum_{t=1}^T \left(\frac{\varepsilon_t^2}{2(\alpha_0 + \alpha_1 \varepsilon_{t-1}^2)}\right) \quad (2.6.15)$$

Selanjutnya persamaan (2.6.5.1.3) diturunkan terhadap masing-masing parameter yaitu:

$$\frac{\partial l}{\partial \alpha_0} = -\frac{1}{2} \sum_{t=1}^T (\alpha_0 + \alpha_1 \varepsilon_{t-1}^2)^{(-1)} + \frac{1}{2} \sum_{t=1}^T \frac{\varepsilon_t^2}{2(\alpha_0 + \alpha_1 \varepsilon_{t-1}^2)}$$

$$\frac{\partial l}{\partial \alpha_1} = -\frac{1}{2} \sum_{t=1}^T \varepsilon_{t-1}^2 \sum_{t=1}^T (\alpha_0 + \alpha_1 \varepsilon_{t-1}^2)^{(-1)} + \frac{1}{2} \sum_{t=1}^T \left(\frac{\varepsilon_{t-1}^2 \varepsilon_t^2}{(\alpha_0 + \alpha_1 \varepsilon_{t-1}^2)}\right)$$

Kemudian estimasi parameter ini dilanjutkan dengan proses iterasi Newton Raphson.

2.7 Estimasi Parameter dari Model Regresi GARCH

Parameter model GARCH diduga dengan menggunakan metode penduga kemungkinan maksimum (Maximum Likelihood Estimation). Karena hasilnya

sangat mirip dengan regresi model ARCH, maka estimasi parameternya akan sangat skematis.

Berdasarkan persamaan (2.6.5.3), model umum GARCH(p,q) adalah

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

dengan

$$y_t = x_t \beta_t + \varepsilon_t$$

atau

$$\varepsilon_t = y_t - x_t \beta_t$$

dimana

y_t : variabel dependen

x_t : vektor variabel penjelas

b : vektor parameter

Dari persamaan diatas, model GARCH terkait dengan residual kuadrat dengan variansi bersyarat. Variansi bersyarat data tersebut diambil pada saat t-1 (dinotasikan sebagai ψ_{t-1}), distribusi error diasumsikan sebagai

$$\varepsilon | \psi_{t-1} \sim N(0, h_t)$$

dimana variansi bersyaratnya adalah

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

dengan kita misalkan $\sigma_t^2 = h_t$

$$h_t = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \alpha_2 \varepsilon_{t-2}^2 + \dots + \alpha_q \varepsilon_{t-q}^2 + \beta_1 h_{t-1} + \beta_2 h_{t-2} + \dots + \beta_p h_{t-p}$$

Kita definisikan :

$$z_t = [1, \varepsilon_{t-1}^2, \varepsilon_{t-2}^2, \dots, \varepsilon_{t-q}^2, h_{t-1}, h_{t-2}, \dots, h_{t-p}]$$

$$\sigma = [\alpha_0, \alpha_1, \dots, \alpha_q, \beta_1, \beta_2, \dots, \beta_p]$$

maka $h_t = z_t \sigma$

Dalam pembahasan estimasi GARCH ini, dimisalkan kita memiliki model regresi GARCH(1,1) sedemikian hingga

$$y_t = \beta_0 + \beta_1 x_t + \varepsilon_t, t = 1, 2, \dots, T$$

dan

$$h_t = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 h_{t-1}$$

dengan u_t adalah proses white noise dengan mean 0 dan variansi 1. Variabel u_t dan ε_{t-1} independen satu dengan yang lain. Maka kita akan memiliki vektor parameter θ sebagai berikut :

$$\theta = (\beta_0, \beta_1, \alpha_0, \gamma_1, \gamma_1)$$

dengan $\beta = (\beta_0, \beta_1)$ maka persamaan diatas dapat ditulis

$$\theta = (\beta, \sigma)$$

$$\text{dengan } \sigma = \begin{bmatrix} \alpha_0 & \gamma_1 & \alpha_1 \end{bmatrix} \text{ dan } \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

Bollerslev (1986) menggunakan algoritma Berndt, Hall, Hall and Hausman (BHHH) dalam mengestimasi model regresi GARCH. Dengan estimator BHHH,

hanya perlu turunan pertama dari fungsi likelihood, sehingga tidak perlu lagi mencari turunan keduanya.

Misalkan L menyatakan rata-rata dari fungsi likelihood dan L_t menyatakan fungsi log likelihood untuk observasi ke- t , maka L dan L_t dinyatakan sebagai berikut

$$L = T^{-1} \sum_{t=1}^T L_t$$

$$L_t = -\frac{1}{2} \log 2\pi - \frac{1}{2} \log h_t - \frac{1}{2} \frac{\varepsilon_t^2}{h_t}$$

Untuk mengestimasi parameter dari variansi σ terlebih dahulu kita cari vektor gradien dari fungsi likelihood, dengan cara menurunkan fungsi likelihood terhadap parameter σ .

$$\frac{\partial L_t}{\partial \sigma} = \frac{1}{2h_t} \frac{\partial h_t}{\partial \sigma} \left(\frac{\varepsilon_t^2}{h_t} - 1 \right)$$

dimana

$$\frac{\partial h_t}{\partial \sigma} = z_t + \sum_{i=1}^p \beta_i \frac{\partial h_{t-1}}{\partial \sigma}$$

Kemudian estimasi parameter ini dilanjutkan dengan metode Newton - Raphson [3].

Untuk mengestimasi parameter dari mean yaitu β , terlebih dahulu kita cari vektor gradien dari fungsi likelihood, dengan cara menurunkan fungsi likelihood terhadap parameter β . Hasilnya seperti yang kita peroleh pada persamaan berikut yaitu

$$\frac{\partial h_t}{\partial \beta} = \frac{\varepsilon_t x_t}{h_t} + \frac{1}{2} \left(\frac{1}{h_t} \right) \frac{\partial h_t}{\partial \beta} \left(\frac{\varepsilon_t^2}{h_t} - 1 \right)$$

dimana

$$\frac{\partial h_t}{\partial \beta} = -2 \sum_{j=1}^q \alpha_j x_{t-j} \varepsilon_{t-j} + \sum_{j=1}^p j \frac{\partial_{t-1}}{\partial \beta}$$

Kemudian estimasi parameter ini dilanjutkan dengan metode Newton - Raphson [3]

2.7.1 Uji Q-Ljung Box

Uji Q-Ljung Box merupakan uji yang digunakan untuk melihat ada atau tidaknya autokorelasi pada data. Hipotesis yang akan diuji adalah

$H_0 : \rho_k = 0$ (tidak ada autokorelasi antar residu)

$H_1 : \rho_k \neq 0$ (ada autokorelasi antar residu)

Statistik uji yang digunakan yaitu

$$Q_{LB} = (n)(n+2) \sum_{k=1}^K \frac{r_k^2}{n-k}$$

Q_{LB} = nilai Q Ljung-Box

n = banyak data pengamatan

r_k^2 = koefisien autokorelasi sisaan pada lag k dengan $k = 1, 2, \dots, K$

K = lag maksimum

Jika nilai $Q_{LB} > \lambda_{(k-p-q)^2}(\alpha)$ maka hipotesis nol ditolak atau jika p-value lebih kecil dari λ maka disimpulkan untuk menolak H_0 . Hal ini berarti masih terdapat autokorelasi pada residual model.

Uji Lagrange Multiplier

Uji Lagrange Multiplier (LM) adalah uji untuk mendeteksi apakah masih terdapat atau tidak efek heteroskedastisitas/ARCH pada residual kuadrat. Uji ini dilakukan dengan meregresikan residual kuadrat kemudian menduga nilai koefisiennya. Misalkan model regresi residual kuadrat sebagai berikut:

$$\varepsilon_t^2 = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \dots + \alpha_k \varepsilon_{t-k}^2$$

hipotesis dari uji ini adalah

$$H_0 : \alpha_0 = \alpha_1 = \dots = \alpha_k = 0 \text{ (tidak ada efek ARCH)}$$

$$H_1 : \alpha_i \neq 0, i = 0, 1, 2, \dots, k \text{ (terdapat efek ARCH)}$$

Statistik ujinya adalah:

$$LM = nR^2$$

dengan

n = banyak data pengamatan

R^2 = koefisien determinasi model regresi residu kuadrat

Statistik uji LM menyebar mengikuti distribusi χ^2 dengan q adalah orde dari model ARCH. Hipotesis nol ditolak jika $LM > \chi_q^2(\alpha)$.

Uji Jarque Bera

Uji Jarque Bera digunakan untuk menguji kenormalan residual model. Ukuran yang digunakan pada uji ini adalah skewness (kemenjuluran) dan kurtosis (keruncingan). Hipotesis yang akan diuji adalah:

H_0 : sisaan menyebar normal

H_1 : sisaan tidak meenyebar normal

Statistik uji pada Jarque Bera adalah:

$$JB = \frac{n}{6} \left(S^2 + \frac{(K - 3)^2}{4} \right) \quad (2.7.1)$$

dengan

n = banyak data pengamatan

S = *skewness* (kemenjuluran)

K = *kurtosis* (keruncingan)

Statistik uji Jarque Berra menyebar mengikuti sebaran χ^2 dengan derajat bebas 2. Hipotesis nol ditolak jika $JB > \chi^2_{(2)}(\alpha)$.

Akaike Information Criterion (AIC) dan Bayesian Information Criterion (BIC)

Jika terdapat dua atau lebih pada satu model time series, maka dilakukan pemilihan model terbaik. Kriteria pemilihan model yang paling umum digunakan pada model time series adalah *Akaike Information Criterion* (AIC) dan *Bayesian Information Criterion* (BIC). AIC dan BIC digunakan untuk melihat kecocokan model terhadap data. AIC pertama kali dikemukakan oleh Hirotugu Akaike pada tahun 1974, sedangkan BIC atau disebut juga Schwarz Information Criterion dipublikasikan pada tahun 1978. Dengan AIC dan BIC dipilih model dengan jumlah lag yang paling cocok dengan model.

Berikut perumusan AIC dan BIC :

$$AIC = n\ln(RSS) + 2k$$

$$BIC = n\ln(RSS) + k\ln(n) \quad (2.7.2)$$

dengan

n = banyak data pengamatan

RSS = jumlah kuadrat residual

K = Jumlah parameter yang diestimasi

AIC digunakan untuk menentukan model yang dapat menjelaskan data dengan jumlah parameter yang baik untuk menduga data. Sedangkan BIC digunakan untuk memperbaiki sifat pendugaan yang terlalu tinggi dari AIC. Semakin kecil nilai AIC dan BIC sebuah model maka semakin ideal hasil pe-modelannya. Berlaku juga sebaliknya.

2.8 Fuzzy Time Series

2.8.1 Himpunan Fuzzy

Himpunan tegas (crisp) merupakan himpunan yang nilai derajat keanggotaan dari elemen hanya mempunyai dua kemungkinan yaitu 0 dan 1. Nilai 0 menunjukkan bahwa suatu elemen tidak terdapat di dalam himpunan, sedangkan nilai 1 menunjukkan elemen tersebut terdapat dalam himpunan. Jika pada himpunan tegas nilai derajat keanggotaan hanya 0 dan 1, maka terdapat sebuah himpunan yang dikenal dengan himpunan fuzzy yang dapat digunakan untuk

mengatasi kelemahan dari himpunan tegas. Nilai derajat keanggotaan himpunan fuzzy berada pada interval $[0,1]$. Dapat diartikan bahwa nilai kebenaran suatu elemen tidak hanya benar atau salah. Nilai 0 menunjukkan salah, nilai 1 menunjukkan benar, dan masih ada nilai-nilai yang terletak antara benar dan salah tersebut, sebagai contoh 0,1; 0,4; 0,75; ::: dst Beberapa hal yang berkaitan dengan fuzzy yaitu:

1. Variabel Fuzzy

Variabel fuzzy adalah variabel yang akan dibahas dalam suatu sistem fuzzy.

2. Himpunan Fuzzy

Himpunan fuzzy adalah himpunan dengan nilai derajat keanggotaan untuk setiap elemennya berada pada interval $[0,1]$. Ada 2 jenis himpunan fuzzy yaitu:

- a Linguistik, yaitu penamaan kelompok berdasarkan suatu keadaan, atau kondisi tertentu dengan menggunakan bahasa alami, seperti: sangat kurus, kurus, ideal, gemuk dan sangat gemuk
- b Numerik, yaitu suatu nilai yang menunjukkan ukuran dari suatu variabel berupa sebuah bilangan seperti: 4, 27, 50, 98 dll.

3. Himpunan Semesta

Himpunan semesta adalah keseluruhan nilai yang dapat dioperasikan dalam suatu variabel fuzzy. Nilai dari himpunan semesta dapat berupa bilangan positif maupun negatif.

4. Domain Himpunan Fuzzy

Domain himpunan fuzzy merupakan keseluruhan himpunan nilai yang memenuhi persyaratan dalam himpunan semesta, serta akan dioperasikan dalam suatu himpunan fuzzy. Nilai domain berupa bilangan positif maupun negatif.

5. Fungsi Keanggotaan

Fungsi keanggotaan merupakan suatu kurva yang menggambarkan pemetaan titik-titik data ke dalam nilai derajat keanggotaannya, yang biasanya terdapat pada interval antara 0 sampai 1.

2.8.2 Rantai Markov (*Markov Chain*)

Rantai markov pertama kali diperkenalkan pada tahun 1906. Awalnya, penerapan rantai Markov digunakan untuk menganalisis dan memperkirakan perilaku partikel-partikel gas dalam suatu wadah tertutup. Setelah itu, rantai Markov digunakan untuk meramalkan keadaan cuaca. Sifat dari rantai Markov, apabila suatu proses berada dalam suatu keadaan tertentu, maka peluang berkembangnya proses dimasa yang akan datang, hanya bergantung pada keadaan saat ini, dan tidak bergantung pada keadaan sebelumnya. Atau dapat dikatakan bahwa rantai Markov adalah kejadian dimana peluang bersyarat kejadian yang akan datang tergantung pada kejadian sekarang. Analisis rantai markov merupakan suatu bentuk khusus dari model probabilistik yang lebih umum dikenal dengan proses stokastik.

Proses stokastik merupakan suatu barisan kejadian yang memenuhi hukum-

hukum peluang. Jika dari pengamatan yang lalu, keadaan yang akan datang suatu barisan dapat diramalkan secara pasti, maka barisan kejadian itu dinamakan deterministik. Jika pengalaman yang lalu hanya dapat menyajikan struktur peluang keadaan yang akan datang, maka barisan kejadian yang demikian disebut stokastik. Berikut ini diberikan definisi proses stokastik dan proses stokastik bersifat Markov.

Definisi 2.7.2.1 Proses stokastik $X_n(t)$ merupakan serangkaian peubah acak yang berubah terhadap waktu pengamatan $t \in T$.

Definisi 2.7.2.2 Suatu proses stokastik $X_n(t)$ dikatakan memiliki sifat Markov jika $P(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = P(X_{n+1} = j | X_n = i)$ untuk waktu $n = 0, 1, \dots$ dan untuk setiap $j, i, i_{n-1}, \dots, i_1, i_0$.

Rantai Markov (*Markov Chain*) adalah proses stokastik yang bersifat Markov. $P(X_{n+1} = j)$ adalah notasi peluang dari $X_{n+1} = j$, notasi ini berarti bahwa proses berada pada keadaan ke- j dan pada waktu ke $n + 1$. Sifat rantai Markov ini menyatakan bahwa, peluang bersyarat dari kejadian mendatang $X_{n+1} = j$, hanya bergantung pada keadaan saat ini $X_n = i$.

Definisi 2.7.2.3 Peluang bersyarat $P(X_{n+1} = j | X_n = i)$ dikatakan peluang transisi satu Langkah jika untuk setiap i dan j

$$P(i, j) = P(X_{n+1} = j | X_n = i)$$

tidak bergantung pada waktu n .

Peluang transisi satu langkah yang tidak bergantung pada waktu n dikatakan stasioner. Oleh karena itu, peluang transisi stasioner menyiratkan bahwa peluang transisi tidak berubah seiring waktu, yang dinotasikan dengan $P(i, j)$. Per-

samaan pada definisi 2.7.2.3 menyatakan peluang dari keadaan j pada waktu $n + 1$, $X_{n+1} = j$ yang dinotasikan $P(X_{n+1} = j)$, jika diberikan pada peluang proses keadaan i pada waktu n , yang dinotasikan $P(X_n = i)$. Peluang transisi satu langkah yang bergantung pada waktu sekarang, dan waktu-waktu sebelumnya dinyatakan sebagai berikut:

$$P(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = P(i, j)$$

Peluang transisi satu langkah dapat dibuat ke dalam format matriks transisi. Matriks transisi dari rantai markov X_n , dengan $n = 0, 1, 2, \dots, N$ dengan ruang keadaan $(0, 1, 2, \dots, N)$. Berikut format matriks transisi dari peluang transisi :

$$\mathbf{P} = [p(i, j)] = \begin{bmatrix} p(0, 0) & p(0, 1) & \cdots & p(0, N) \\ p(1, 0) & p(1, 1) & \cdots & p(1, N) \\ \vdots & \vdots & \ddots & \vdots \\ p(N, 0) & p(N, 1) & \cdots & p(N, N) \end{bmatrix}$$

dengan sifat peluang transisi $p(i, j)$ sebagai berikut:

- $p(i, j) \geq 0, \forall i, j$
- $\sum_j p(i, j) = 1, \forall i, j$

2.8.3 Proses awal Pemodelan FTS

Proses pemodelan FTS terbagi dalam beberapa langkah, Langkah awal terdiri atas langkah pembagian interval dan Langkah pembentukan *Fuzzy Logical Relationship* (FLR). Pembagian interval pada penelitian ini adalah dengan metode *average based*, hal ini dikarenakan proses pembagian intervalnya yang

sangat rinci sehingga akan dapat menghasilkan model yang baik berdasarkan nilai keakuratan modelnya.

Langkah tengah dan akhir diproses berdasarkan Algoritma FTS masing-masing. Berikut Langkah awal pemodelan FTS:

Langkah Pertama Menentukan himpunan universal (U) data konkret melalui proses :

$$U = [D_{min} - D_1; D_{max} + D_2] \quad (2.8.1)$$

D_1, D_2 : sembarang bilangan positif.

Langkah kedua Menghitung banyaknya interval himpunan fuzzy dengan average based melalui proses:

1. Menentukan panjang interval U

$$R = [D_{max} + D_2 - D_{min} - D_1] \quad (2.8.2)$$

2. Hitung nilai selisih (*lag absolute*) dan rata-rata nilai selisih (*lag absolute*)

$$\text{lag absolute} = |(D_{t+1}) - D_t| \quad (2.8.3)$$

$$\text{mean lag absolute} = \frac{\sum_{t=1}^{N-1} |(D_{t+1}) - D_t|}{N - 1} \quad (2.8.4)$$

dengan D_t : data pada kondisi ke t dan N mewakili banyaknya data.

3. Menentukan basis interval

$$K = \frac{\text{mean}}{2} \quad (2.8.5)$$

Penentuan basis interval berdasarkan tabel berikut:

Tabel 2.8.3 : Basis Interval

Jangkauan	Basis
0.1 - 1	0.1
1.1 - 10	1
11 - 100	10
101 - 1000	100
1001 - 10000	1000

4. Menentukan banyaknya himpunan fuzzy melalui proses

$$n = \frac{R}{K} \quad (2.8.6)$$

5. Mencari median himpunan fuzzy

$$m_i = \frac{\text{Batas bawah } u_i + \text{Batas atas } u_i}{2}$$

Langkah ketiga Merumuskan tingkat kedudukan himpunan fuzzy terhadap A_i dan fuzzyfikasi. fuzzyfikasi pada data konkret yaitu proses mengonversikan karakter data input yang awalnya berupa numerik menjadi linguistik memanfaatkan tingkat kedudukan yang disimpan dalam basis pengetahuan fuzzy. Banyaknya variabel linguistik yang didapat boleh bebas, karena tidak adanya aturan baku. Perumusan himpunan fuzzy pada A_i adalah dengan tingkat kedudukan. Tingkat kedudukan dari himpunan fuzzy u_i dapat disederhanakan sebagai berikut:

$$\mu_{A_i}(u_i) = \begin{cases} 1, & \text{jika } i = i \\ 0.5, & \text{jika } i = i - 1 \text{ atau } i = i + 1 \\ 0, & \text{yang lainnya} \end{cases} \quad (2.8.7)$$

dengan $1 \leq i \leq n, n$: banyaknya himpunan fuzzy,

Tabel 2.8.4: Tingkat Kedudukan Himpunan Fuzzy

$\mu_{A_i}(u_i)$	A_1	A_2	A_3	...	A_{n-2}	A_{n-1}	A_n
A_1	1	0,5	0	...	0	0	0
A_2	0,5	1	0,5	...	0	0	0
A_3	0	0,5	1	...	0	0	0
...
A_{n-2}	0	0	0	...	1	0,5	0
A_{n-1}	0	0	0	...	0,5	1	0,5
A_n	0	0	0	...	0	0,5	1

Langkah keempat Menyusun *Fuzzy Logical Relationship* (FLR)

FLR bersandarkan pada data konkret. Langkah ini menetapkan relasi logika fuzzy yaitu $A_i \rightarrow A_j$. Himpunan A_i merupakan kondisi sekarang $D_{(t-1)}$ dan A_j adalah kondisi berikutnya pada waktu ke D_t . FLR menghubungkan relasi antara nilai linguistik yang ditentukan berdasarkan tabel fuzzyfikasi yang didapat sebelumnya. Langkah kelima dan seterusnya dibahas berdasarkan algoritma FTS masing-masing.

2.8.4 FTS Chen

FTS Chen diperkenalkan oleh Shyi Ming Chen pada 1996. Langkah FTS Chen lebih sederhana dibandingkan FTS lain pada penelitian ini. Berikut Langkah selanjutnya FTS Chen:

Langkah kelima Menyusun *Fuzzy Logical Relationship Group* (FLRG)

FLRG dikerjakan melalui proses kategorisasi *fuzzy* yang mempunyai kondisi sekarang yang serupa kemudian dikumpulkan menjadi satu kumpulan pada kondisi berikutnya. Setiap FLR dikumpulkan membentuk FLRG yang saling berkaitan.

Misal, ada 4 *Fuzzy Logical Relationship* (FLR) sama, $A_1 \rightarrow A_2, A_1 \rightarrow A_2, A_1 \rightarrow A_2, \text{ dan } A_1 \rightarrow A_3$, membentuk FLRG $A_1 \rightarrow A_2, A_3$ karena menurut Chen $A_1 \rightarrow A_2, A_1 \rightarrow A_2, A_1 \rightarrow A_2$, tidak mempengaruhi hasil model maka cukup dirasa satu FLRG.

Langkah keenam Pembentukan matriks peluang

$$P_{ij} = \frac{M_{ij}}{M_i}, i, j = 1, 2, 3, \dots, n$$

dengan

P_{ij} : peluang perubahan dari kondisi A_i ke A_j dengan satu tahap.

M_{ij} : waktu perubahan dari kondisi A_i ke A_j dengan satu tahap.

M_i : jumlah data dari kondisi A_i dengan mengikuti proses 5 diatas.

Mengikuti kaidah peluang perubahan pada kondisi tersebut, diperoleh matriks peluang perubahan R dengan dimensi $n \times n$, disajikan dengan rumus:

$$R = \begin{pmatrix} \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix} \end{pmatrix}$$

Langkah ketujuh *Defuzzyfikasi*

Defuzzyfikasi adalah mengonversi data keluaran *fuzzy* yang didapat dari susunan logika *fuzzy* menjadi data keluaran bersifat nilai tegas (numeris) sebagai nilai model menggunakan nilai kedudukan yang sesuai pada saat dilakukan *fuzzyfikasi*. Berikut *Defuzzyfikasi* FTS Chen :

- Jika FLRG dari A_i berubah ke himpunan kosong $A_i \rightarrow \theta$, maka model F_t bernilai m_i , yaitu median dari u_i dengan rumus :

$$F_t = m_i$$

- Jika FLRG dari A_i berubah satu ke satu ($A_i \rightarrow A_k$ dengan $P_{ij} = 0$ dan $P_{ik} = 1, j \neq k$) maka model F_t bernilai m_k yaitu median dari u_i dengan rumus :

$$F_t = m_k$$

- Jika FLRG dari A_i berubah satu ke banyak ($A_i \rightarrow A_1, A_2, \dots, A_p, i = 1, 2, \dots, p$), maka model F_t bernilai :

$$F_t = \frac{m_{1(t-1)} + m_{2(t-1)} + \dots + m_{p(t-1)}}{p}$$

2.8.5 FTS Lee

Pada 1993, Song dan Chissom mengembangkan FTS Chen menjadi FTS Song & Chissom. Kemudian FTS Lee ini merupakan perkembangan dari model FTS Song & Chissom, Cheng dan Chen. Berikut Langkah lanjutan dari FTS Lee :

Langkah kelima Menyusun *Fuzzy Logical Relationship Group* (FLRG)

Misal, ada 3 *Fuzzy Logical Relationship* (FLR) $A_1 \rightarrow A_2, A_1 \rightarrow A_2$, dan $A_1 \rightarrow A_3$, membentuk FLRG $A_1 \rightarrow A_2, A_2, A_3$. Karena kaidah Lee memaparkan $A_1 \rightarrow A_2, A_1 \rightarrow A_2$ dihitung lebih dari satu karena dapat mempengaruhi hasil model.

Langkah keenam Pembentukan matriks peluang

Langkah ini sama dengan Langkah keenam FTS Chen diatas

Langkah ketujuh *Defuzzyfikasi*

- Jika FLRG dari A_i berubah ke himpunan kosong $A_i \rightarrow ?$, maka model F_t bernilai m_i , yaitu median dari u_i dengan rumus :

$$F_t = m_i$$

- Jika FLRG dari A_i berubah satu ke satu ($A_i \rightarrow A_k$ dengan $P_{ij} = 0$ dan $P_{ik} = 1, j \neq k$) maka model F_t bernilai m_k yaitu median dari u_i dengan rumus :

$$F_t = m_k P_{ik} = m_k$$

- Jika FLRG dari A_i berubah satu ke banyak ($A_i \rightarrow A_1, A_2, \dots, A_n, i = 1, 2, \dots, n$) , maka model F_t bernilai :

$$F_t = m_1P_{i1} + m_2P_{i2} + \dots + m_{(i-1)}P_{i(i-1)} + m_iP_{ii} + \dots + m_nP_{in}$$

dengan $m_1, m_2, \dots, m_{i-1}, m_{i+1}, \dots, m_n$: median dari $u_1, u_2, \dots, u_{i-1}, u_{i+1}, \dots, u_n$.

2.8.6 FTS Tsaur

Tsaur mengembangkan model Fuzzy Time Series menjadi Fuzzy Time Series Markov Chain (FTSMC). Model ini mengombinasikan model fuzzy time series dengan rantai Markov. Berikut ini diuraikan langkah lanjutan untuk menyelesaikan model Fuzzy Time Series Tsaur:

Langkah kelima Menyusun *Fuzzy Logical Relationship Group* (FLRG)

Misal, ada 3 *Fuzzy Logical Relationship* (FLR) $A_1 \rightarrow A_2, A_1 \rightarrow A_2$, dan $A_1 \rightarrow A_3$, membentuk FLRG $A_1 \rightarrow A_2, A_2, A_3$. Karena Tsaur memaparkan $A_1 \rightarrow A_2, A_1 \rightarrow A_2$ dihitung lebih dari satu karena dapat mempengaruhi hasil model.

Langkah keenam Pembentukan matriks peluang

Langkah ini sama dengan Langkah keenam FTS Chen di atas

Langkah ketujuh Defuzzyfikasi

- Jika FLRG dari A_i berubah ke himpunan kosong $A_i \rightarrow \theta$, maka model F_t bernilai m_i , yaitu median dari u_i dengan rumus:

$$F_t = m_i$$

- Jika FLRG dari A_i berubah satu ke satu ($A_i \rightarrow A_k$ dengan $P_{ij} = 0$ dan $P_{ik} = 1, j \neq k$) maka model F_t bernilai m_k yaitu median dari u_i dengan rumus :

$$F_t = m_k P_{ik} = m_k$$

- Jika FLRG dari A_i berubah satu ke banyak ($A_i \rightarrow A_1, A_2, \dots, A_n, i = 1, 2, \dots, n$) dan kelompok data X_{t-1} ketika $t - 1$ ada pada kondisi A_i , maka model F_t bernilai :

$$F_t = m_1 P_{i1} + \dots + m_{(i-1)} P_{i(i-1)} + X_{t-1} P_{ii} + m_{(i+1)} P_{i(i+1)} + \dots + m_n P_{in}$$

Nilai m_j disubstitusikan oleh X_{t-1} agar didapatkan data dari kondisi A_j ketika $t - 1$.

Langkah kedelapan Perhitungan sinkronisasi model,

Perhitungan sinkronisasi model bertujuan memperbaiki error pemodelan yang disebabkan biasanya matriks peluang perubahan. Aturan sinkronisasi model (D_t) :

- Jika kondisi A_i ketika $t - 1$ sebagai $F_{t-1} = A_i$ ada pergerakan maju ke kondisi A_{i+s} ketika t , dimana $1 \leq s \leq n - 1$ maka sinkronisasi D_t :

$$D_t = \left(\frac{l}{2}\right)s, (1 \leq s \leq n - 1) \quad (2.8.8)$$

dengan s : banyak lompatan perubahan maju.

- Jika kondisi A_i ketika $t - 1$ sebagai $F_{t-1} = A_i$ ada pergerakan mundur ke kondisi A_{i-v} pada saat t , dengan $1 \leq v \leq i$ maka sinkronisasi D_t :

$$D_t = -\left(\frac{l}{2}\right)v, (1 \leq v \leq i) \quad (2.8.9)$$

dengan v : banyak lompatan perubahan mundur.

Langkah kesembilan Hasil akhir model

Perhitungan hasil akhir model FTS Tsaur diperoleh melalui proses menambah

hasil awal model dan sinkronisasi model. Bentuk umum hasil akhir model (FM_t) yaitu:

$$FM_t = F_t + D_t \quad (2.8.10)$$

FM_t : hasil akhir model pada saat ke- t

2.9 Keakuratan Model

Untuk mengetahui tingkat keakuratan model maka dihitung nilai error-nya. Model terbaik ialah model dengan error paling kecil. Diantara Metode perhitungan error adalah:

1. *Root Mean Squared Error* (RMSE)

RMSE adalah metode alternatif untuk mengukur tingkat akurasi hasil pemodelan suatu model.

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (X_t - F_t)^2} \quad (2.9.1)$$

X_t : data observasi pada waktu ke- t

2. *Mean Absolute Error* (MAE)

Untuk mengukur akurasi pemodelan dengan meratakan nilai absolut error pemodelan.

$$MAE = \frac{1}{n} \sum_{t=1}^n |X_t - F_t| \quad (2.9.2)$$

3. *Mean Absolute Percentage Error* (MAPE).

MAPE adalah rata-rata dari semua persentase kesalahan antara data asli

dan hasil pemodelan.

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{X_t - F_t}{X_t} \right| * 100\% \quad (2.9.3)$$

Berikut kategori penilaian MAPE :

- (a) Jika rate MAPE $< 10\%$ maka kriteria modelnya sangat bagus.
- (b) Jika rate MAPE 10% s/d 20% maka kriteria modelnya bagus
- (c) Jika rate MAPE 20% s/d 50% maka kriteria modelnya cukup bagus
- (d) Jika rate MAPE $> 50\%$ maka kriteria modelnya tidak bagus

BAB III

METODE PENELITIAN

3.1 Jenis dan Sumber Data

3.2 Teknik Pengolahan dan Analisis Data

DAFTAR PUSTAKA

- [1] Aldous, J. M. dan Robin J. W. 2000. *Graphs and Applications an Introductory Approach*, Springer.
- [2] Baskoro, E. T. dan Yulianti, L. 2011. On Ramsey minimal graphs for $2K_2$ versus P_n , *Adv. Appl. Discrete Math* **8**(2):83-90.
- [3] Baskoro, E. T. dan Wijaya, K. 2015. On Ramsey $(2K_2, C_4)$ – *Minimal Graphs*, Mathematics in the 21st Century: 6th World Conference, Lahore *Spinger Proceedings in Mathematics & Statistics* **98**:11-17.