

BAB I

PENDAHULUAN

1.1 Latar Belakang

Minyak mentah merupakan salah satu komoditas yang memegang peranan penting dalam perekonomian. Minyak mentah memiliki manfaat utama sebagai sumber tenaga untuk menggerakkan kendaraan. Selain itu, minyak mentah juga digunakan dalam berbagai industri seperti pembuatan pelarut, baja, sabun, detergen, plastik, dan bahan peledak. Banyaknya peranan minyak mentah, mengakibatkan tingginya permintaan minyak mentah di setiap negara. Jika pasokan minyak mentah dunia habis, maka harga minyak mentah akan meningkat. Fluktuasi harga minyak mentah tersebut berdampak pada kesehatan suatu ekonomi negara, terutama negara pengekspor minyak mentah. Hal ini dikarenakan, negara yang mengimpor minyak mentah lebih rentan terhadap perubahan harga, sebab minyak mentah dihargai dan dijual dalam dolar.

Salah satu jenis minyak mentah yang digunakan sebagai patokan dalam penentuan harga minyak mentah adalah *West Texas Intermediate* (WTI). Minyak WTI adalah sejenis minyak bumi yang memiliki belerang dan *gravity* rendah. Kandungan belerang berkisar 0.24% dan *gravity* 39.6°. Minyak WTI disebut sebagai minyak mentah manis dan ringan, serta dianggap memiliki

kualitas baik untuk diolah menjadi bensin. Penyulingan terutama dilakukan di daerah Teluk dan Amerika Serikat. Tingginya kualitas minyak WTI, menyebabkan harga minyak WTI dijadikan harga standar minyak dunia [24].

Pergerakan harga minyak WTI dapat dimodelkan dengan model deret waktu. Model deret waktu adalah model yang menggunakan data-data yang terurut berdasarkan waktu. Data deret waktu diyakini memiliki pola yang berulang, dimana periode pada masa lalu, akan terulang lagi dimasa sekarang atau masa depan. Analisis model deret waktu bertujuan untuk menemukan suatu pola yang dapat digunakan dalam memodelkan kejadian mendatang. Pola dari data deret waktu berupa pola jangka pendek (*short memory*) dan jangka panjang (*long memory*).

Data deret waktu berpola *short memory*, apabila korelasi antar data deret waktu lemah dalam rentang waktu yang kecil. Data deret waktu ini, dapat dimodelkan dengan model *Auto-regressive Integrated Moving Average* (ARIMA). Model ARIMA merupakan model ARMA untuk data deret waktu tidak stasioner. Data deret waktu akan distasionerkan dengan melakukan *differencing*, yang mana nilai *differencing* (d) berupa bilangan bulat non-negatif. Model ARMA merupakan gabungan dari model AR dan model MA. Model AR (*Auto-regressive*) yaitu model untuk melihat pergerakan suatu variabel melalui variabel itu sendiri. Model MA (*Moving Average*) yaitu model untuk mengetahui pergerakan suatu variabel dengan residunya di masa lalu. Model ARIMA menghasilkan peramalan jangka pendek yang akurat. Biasanya, nilai peramalan akan cenderung *flat* untuk periode yang panjang [36].

Adakalanya, data deret waktu memiliki korelasi yang kuat pada waktu pengamatan yang panjang *long memory*. Hal ini, terlihat dari nilai-nilai auto-korelasi pada plot *Autocorrelation Function* (ACF), yang turun secara lambat dalam jangka waktu yang panjang. Akibatnya, nilai *differencing* (d) berupa bilangan riil yang dapat diatasi dengan model *Auto-regressive Fractionally Integrated Moving Average* (ARFIMA) [3]. Pada tahun 1980, Granger dan Joyeux mengemukakan model ARFIMA yang merupakan pengembangan dari model ARIMA yang hanya dapat meramalkan data deret waktu berpola *short memory*. Namun, pada model ARFIMA dapat meramalkan suatu data deret waktu yang memiliki pola *short memory* dan *long memory* [15]. Kemudian pada tahun 1981, Hosking mengkaji sifat-sifat pola *long memory* dari model ARFIMA stasioner dan nonstasioner [18]. Model ARFIMA nonstasioner dapat dilakukan *differencing* agar menjadi stasioner. Salah satu metode yang digunakan untuk menaksir nilai *differencing* (d) yaitu metode *Geweke and Porter Hudak* (GPH). Metode GPH digunakan karena dapat menaksir parameter d secara langsung, tanpa mengetahui nilai orde AR yaitu p dan orde MA yaitu q terlebih dahulu [14].

Pemodelan data deret waktu secara klasik seperti ARIMA dan ARFIMA memerlukan beberapa asumsi yang harus dipenuhi. Hal ini, akan menjadi kendala bagi suatu data deret waktu yang tidak dapat memenuhi asumsi tersebut. Oleh sebab itu, salah satu pemodelan data deret waktu tanpa memerlukan asumsi yaitu model *Fuzzy Time Series* (FTS). Pada tahun 1960-an ditemukan suatu teori yang dikenal dengan logika *fuzzy*. Kelebihan dari logika

fuzzy yaitu logika *fuzzy* mudah dimengerti, memiliki logika yang fleksibel, dan logika *fuzzy* mampu memodelkan fungsi-fungsi nonlinear yang kompleks [5]. Kemudian pada tahun 1993, Song dan Chissom mengembangkan logika *fuzzy* menjadi suatu metode yang digunakan untuk melakukan pemodelan, metode tersebut dinamakan model *Fuzzy Time Series* (FTS) [29]. Model FTS merupakan pemodelan data deret waktu yang menggunakan prinsip-prinsip *fuzzy* sebagai dasarnya. Model FTS menggunakan pola dari data yang telah lalu kemudian data tersebut digunakan untuk memproyeksikan data yang akan datang [31].

Pada tahun 1996, metode FTS juga dikembangkan oleh Chen, dengan menggunakan operasi aritmatika sederhana untuk melakukan pemodelan di Universitas Alabama [9]. Kemudian pada tahun 2008, Cheng mengemukakan sebuah logika dari metode FTS, dimana digunakan pembobotan dan pemodelan adaptif pada metode FTS tersebut [10]. Pada tahun 2012, Tsaur pertama kali mengusulkan *Fuzzy Time Series Markov Chain* (FTSMC), yang merupakan konsep baru dalam penelitiannya untuk menganalisis keakuratan prediksi nilai tukar mata uang Taiwan dengan dolar US. Dalam penelitiannya, Tsaur menggabungkan metode *Fuzzy Time Series* dengan rantai markov. Hasil penelitian tersebut menunjukkan bahwa, metode *Fuzzy Time Series Markov Chain* memberikan hasil yang cukup baik, dibandingkan metode *Fuzzy Time Series* yang diusulkan oleh Song dan Chissom dan Chen [32]. Pada tahun 2020, Ramadani dkk, melakukan penelitian mengenai perbandingan tiga model FTS. Model FTS yang digunakan yaitu model FTS Chen, *FTS Segmented*

Chen dan FTSMC dalam meramalkan harga bitcoin. Pada penelitian tersebut, berdasarkan tingkat keakuratannya yaitu *Mean Absolute Percentage Error* (MAPE), model FTSMC memberikan hasil yang lebih baik dalam meramalkan data harga bitcoin [25]. Hal ini menguatkan bahwa, model FTSMC memberikan akurasi pemodelan yang cukup baik dari model FTS lainnya.

Pada tahun 2011, Saima dkk melakukan penelitian mengenai model *hybrid fuzzy time series*, yaitu mengembangkan model *fuzzy* tipe 2. Model tersebut menggunakan ARIMA yang di *hybrid* dengan model *fuzzy* tipe 2. Model *Fuzzy* tipe 2 ini dapat menangani ketidakpastian dalam data deret waktu [28]. Pada tahun 2018, Panigrahi dkk melakukan penelitian tentang model *hybrid* ARIMA dengan *Artificial Neural Network* (ANN), dalam meramalkan data volatilitas. Berdasarkan penelitian tersebut, diperoleh model *hybrid* memberikan peramalan yang baik karena residu data lebih kecil [23].

Pada penelitian ini, akan dibahas tentang pemodelan harga minyak *West Texas Intermediate* (WTI). Berdasarkan penelitian sebelumnya model ARIMA dan ARFIMA diyakini dapat memodelkan data deret waktu dengan baik. Data yang digunakan merupakan data deret waktu yang berpola *long memory*. Data tersebut telah memenuhi asumsi-asumsi residu pada model ARIMA dan ARFIMA. Pada model FTSMC, asumsi-asumsi tersebut tidak menjadi perhatian dalam pemodelan sehingga diduga model FTSMC lebih fleksibel dalam memodelkan data deret waktu. Kemudian, akan diajukan suatu alternatif model baru yang disebut sebagai model *hybrid* ARIMA-FTSMC. Model *hybrid* ARIMA-FTSMC menerapkan model ARIMA sebagai model

dasarnya. Pada penelitian sebelumnya, belum terdapat perbandingan antara model ARIMA, ARFIMA, FTSMC dan model *hybrid* ARIMA dengan FTSMC pada data *long memory*. Oleh karena itu, akan dibandingkan hasil pemodelan dari model deret waktu tersebut untuk menentukan model yang lebih baik. Model dikatakan lebih baik apabila memiliki nilai *Mean Absolute Error* (MAE), *Root Mean Square Error* (RMSE), dan *Mean Absolute Percentage Error* (MAPE) terkecil.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan maka permasalahan yang akan menjadi kajian dalam penelitian ini yaitu:

1. Bagaimana memodelkan harga minyak *West Texas Intermediate* (WTI) menggunakan model ARIMA, ARFIMA, *Fuzzy Time Series Markov Chain* (FTSMC) dan *hybrid* ARIMA-FTSMC?
2. Manakah model yang terbaik di antara model ARIMA, ARFIMA, *Fuzzy Time Series Markov Chain* (FTSMC) dan *hybrid* ARIMA-FTSMC, untuk pemodelan harga minyak WTI berdasarkan MAE, RMSE dan MAPE terkecil?

1.3 Tujuan Penelitian

Dari rumusan masalah, tujuan dari penelitian ini yaitu:

1. Memodelkan harga minyak *West Texas Intermediate* (WTI) dengan menggunakan model ARIMA, ARFIMA, *Fuzzy Time Series Markov Chain*

(FTSMC) dan *hybrid* ARIMA-FTSMC.

2. Menentukan model yang terbaik untuk pemodelan harga minyak WTI berdasarkan MAE, RMSE dan MAPE terkecil.

1.4 Sistematika Penulisan

Tugas akhir ini disusun dengan sistematika penulisan yaitu pada BAB I Pendahuluan akan dibahas mengenai latar belakang, rumusan masalah, tujuan penelitian dan sistematika penulisan. BAB II Landasan teori, menjelaskan teori-teori yang terkait dalam pembahasan dan mendukung masalah yang dibahas. BAB III Metode Penelitian, pada bab ini akan memaparkan tentang cara menyelesaikan masalah yang telah dirumuskan. BAB IV Pembahasan dibahas mengenai proses pemodelan harga minyak *West Texas Intermediate* (WTI) menggunakan model ARIMA, ARFIMA, *Fuzzy Time Series Markov Chain* (FTSMC) dan *hybrid* ARIMA-FTSMC dengan bantuan *software RStudio, Minitab* dan *Microsoft Excel*. BAB V Penutup, memberikan kesimpulan berdasarkan hasil yang diperoleh pada pembahasan dan juga disampaikan saran yang menjadi pedoman untuk peneliti selanjutnya.

BAB II

LANDASAN TEORI

Pada bab ini akan dibahas teori-teori yang terkait dengan pembahasan tentang model ARIMA, ARFIMA, *Fuzzy Time Series Markov Chain* (FTSMC) dan *hybrid* ARIMA-FTSMC dalam memodelkan harga minyak *West Texas Intermediate* (WTI).

2.1 Minyak Mentah

Minyak mentah merupakan bahan bakar yang terbentuk dari tumbuhan dan hewan fosil selama jutaan tahun. Minyak mentah menjadi salah satu faktor penggerak perekonomian dunia. Hal ini dikarenakan industri di seluruh dunia mengandalkan bahan bakar minyak sebagai bahan baku faktor produksi. Minyak mentah diklasifikasikan berdasarkan jenis kandungan benzena dan kepadatan minyak. Standar ini ditetapkan oleh *American Petroleum Institute* (API). Menurut API, satuan yang disebut *gravity* menandakan tingkat kepadatan minyak mentah. Minyak mentah dengan *gravity* lebih dari 40° dianggap sebagai minyak mentah ringan. Sementara, minyak mentah dengan *gravity* kurang dari 20° dianggap sebagai minyak mentah berat. Minyak mentah yang memiliki *gravity* yang berada diantara $20^\circ - 40^\circ$ dianggap sebagai minyak mentah sedang [24].

Ada 4 jenis minyak mentah yang menjadi standar dalam penentuan harga minyak mentah yaitu [24]:

1. *Brent Blend*

Nama minyak mentah *brent blend* berasal dari lokasi geografis tempat minyak ini diekstrak. Campuran minyak jenis ini berasal dari sumur yang berlokasi di laut utara Eropa. *Brent blend* dianggap sebagai minyak mentah manis dengan kandungan belerang 0.37% dan *gravity* 38.06°. Dengan demikian, minyak jenis ini dikategorikan sebagai minyak mentah ringan yang terutama digunakan untuk membuat bensin. Sebagian besar minyak *brent blend* diperdagangkan di Amerika Serikat dan negara-negara mediterania.

2. *West Texas Intermediate*

Minyak *West Texas Intermediate* (WTI) merupakan jenis minyak mentah yang memiliki belerang dan *gravity* rendah. Kandungan belerang 0.24% dan *gravity* 39.6°. Penyulingan pertama minyak WTI dilakukan di daerah teluk dan Amerika Serikat. Minyak WTI disebut sebagai minyak mentah manis dan ringan serta memiliki kualitas baik untuk diolah menjadi bensin. Tingginya kualitas minyak WTI sehingga harga minyak ini dijadikan harga standar minyak dunia.

3. *Russian Export Blend*

Minyak *russian export blend* menjadi acuan bagi harga minyak mentah Rusia. Minyak ini memiliki kandungan belerang 1.2% dan *gra-*

ivity 32°. Kandungan belerang yang tinggi membuat *russian export blend* digolongkan sebagai minyak asam serta memiliki *gravity* sedang. Harga minyak ini umumnya ditentukan dari dua lokasi utama pengiriman yaitu Italia dan Belanda.

4. *Dubai Crude*

Minyak *dubai crude* memiliki *gravity* rendah dengan kandungan belerang 2% dan *gravity* 31°. Minyak *Dubai crude* dijadikan acuan harga bagi minyak yang diekspor di Asia.

2.2 Faktor yang Mempengaruhi Harga Minyak Mentah

Faktor-faktor yang dapat mempengaruhi harga minyak mentah adalah sebagai berikut [4]:

1. *Organization of the Petroleum Exporting Countries* (OPEC) yaitu organisasi negara penghasil minyak. OPEC dapat mempengaruhi harga minyak mentah dengan meningkatkan, atau mengurangi produksi di antara negara-negara anggotanya.
2. Harga minyak mentah dihargai dalam dolar Amerika Serikat. Ketika nilai dolar AS jatuh, harga minyak mentah akan mengalami kenaikan dan berlaku juga sebaliknya.
3. Cuaca yang terjadi bukan pada musimnya, mempengaruhi harga minyak mentah. Kemajuan teknologi membuat terciptanya teknologi alternatif pengembangan minyak mentah. Misalkan pasir tar dan serpih minyak

yang lebih mudah diperoleh. Teknologi alternatif ini memiliki potensi untuk menjaga ketersediaan bahan bakar selama puluhan tahun. Setiap pengembangan yang ada mempunyai potensi untuk mempengaruhi harga minyak mentah.

2.3 Data Deret Waktu

Data deret waktu merupakan sekumpulan nilai suatu variabel yang dikumpulkan secara berkala pada waktu tertentu. Misalnya harian, mingguan, bulanan, tahunan, dan sebagainya. Data deret waktu menjelaskan perilaku data dari waktu ke waktu.

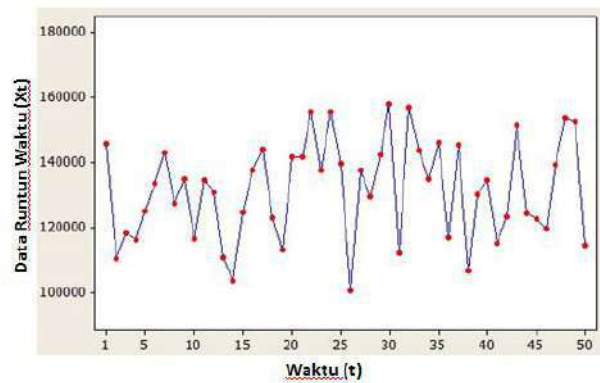
2.3.1 Pola Data

Tahap awal untuk menentukan suatu model deret waktu yaitu dengan melihat pola data. Berikut dijelaskan beberapa jenis pola data yaitu [21]:

1. Pola Data Horizontal

Pola data horizontal terjadi apabila data pengamatan berfluktuasi secara acak yang membentuk garis horizontal. Data berpola horizontal disebut juga dengan data stasioner. Bentuk pola data horizontal dapat dilihat pada Gambar 2.3.1 berikut:



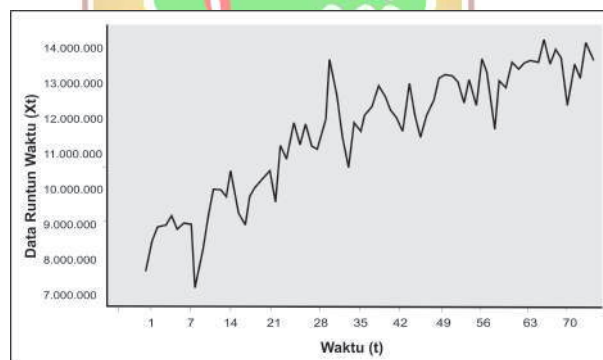


Gambar 2.3.1: Pola Data Horizontal

2. Pola Data *Trend*

Pola data *trend* terjadi apabila data pengamatan berfluktuasi selama periode yang lama. Pola data *trend* disebut dengan data tidak stasioner.

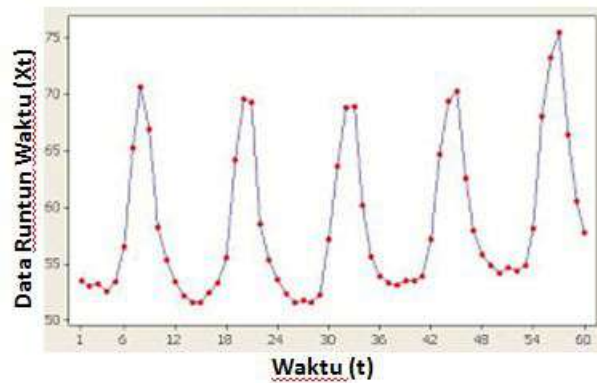
Bentuk pola data *trend* dapat dilihat pada gambar berikut:



Gambar 2.3.2: Pola Data *Trend*

3. Pola Data Musiman

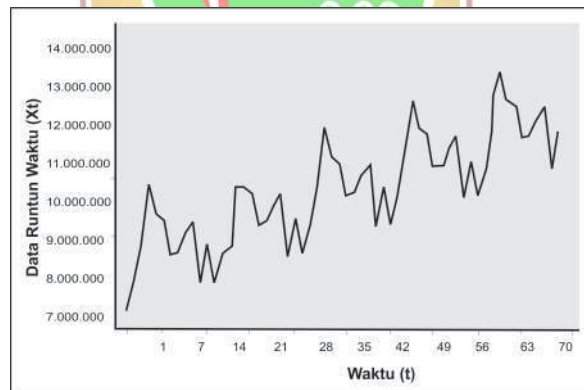
Pola data musiman terjadi apabila data pengamatan dipengaruhi oleh faktor-faktor musim yang berulang, dari satu periode ke periode berikutnya. Bentuk pola data musiman adalah sebagai berikut:



Gambar 2.3.3: Pola Data Musiman

4. Pola Data Siklis

Pola data siklis terjadi apabila data pengamatan berfluktuasi 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: Pola Data Siklis

2.3.2 Stasioneritas Data

Setelah pola data diidentifikasi, tahap selanjutnya memeriksa stasioneritas data. Data deret waktu dikatakan stasioner apabila data menunjukkan pola yang konstan dari waktu ke waktu. Dengan kata lain, tidak terdapat kenaikan atau penurunan pada data yang terlalu mencolok. Ada dua perilaku stasioneritas data, yaitu:

1. Stasioner terhadap Nilai Tengah

Data deret waktu dikatakan stasioner terhadap nilai tengah, apabila data berfluktuasi 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 $E(X_t) = \mu$, dimana nilai μ merupakan nilai tengah yang tidak bergantung terhadap t [21]. Data deret waktu yang tidak stasioner terhadap nilai tengah dapat distasionerkan dengan melakukan *differencing* [36].

Differencing adalah menghitung nilai selisih dari data pengamatan. Nilai selisih tersebut diperiksa apakah stasioner atau tidak. Jika belum stasioner, maka dilakukan *differencing* 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:

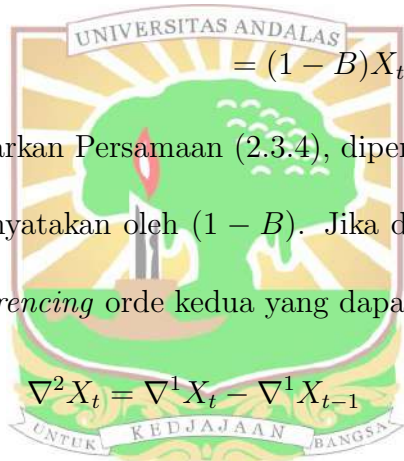
$$BX_t = X_{t-1}. \quad (2.3.2)$$

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:

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

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

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



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:

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

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 [21]. Jika

data tidak stasioner terhadap ragam, maka dapat dilakukan transformasi pada data asli agar stasioner terhadap ragam [12].

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 [35]:

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

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 [35]:

Tabel 2.3.1: Bentuk Transformasi

λ	Bentuk Transformasi
-0.5	$\frac{1}{\sqrt{X_t}}$
0	$\ln X_t$
0.5	$\sqrt{X_t}$
1	X_t (tidak ditransformasi)

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 disekitar 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 [16].

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 [16]:

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

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:

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

$H_1 : \delta \neq 0$ (data stasioner)

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

sebagai berikut:

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

dimana $SE(\hat{\delta})$ merupakan *standard error* untuk $\hat{\delta}$. Kriteria pengambilan keputusan yaitu:

- Jika nilai kritis tabel-t ADF \geq statistik uji ADF maka tolak H_0 , hal ini berarti data stasioner.
- Jika nilai kritis tabel-t ADF $<$ statistik uji ADF maka terima H_0 , hal ini berarti data tidak stasioner.

dimana tabel-t ADF dapat dilihat pada Lampiran 2.1.

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 stasioner.

Definisi 2.4.1. [7] Misalkan X_t adalah data deret waktu stasioner, maka

fungsi autokovarian (ACVF) dari X_t pada lag k adalah

$$\gamma_k = Cov(X_t, X_{t+k}) = E[(X_t - \mu)(X_{t+k} - \mu)]. \quad (2.4.11)$$

Fungsi autokorelasi (ACF) dari X_t adalah

$$\rho_k = Corr(X_t, X_{t+k}) = \frac{\gamma_k}{\gamma_0}. \quad (2.4.12)$$

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 [36]:

$$\hat{\gamma}_k = \frac{1}{n} \sum_{t=1}^{n-k} (X_t - \bar{X})(X_{t+k} - \bar{X}), \quad (2.4.13)$$

dan fungsi autokorelasi untuk sampel X_1, X_2, \dots, X_n yaitu [36]:

$$\hat{\rho}_k = \frac{\sum_{t=1}^{n-k} (X_t - \bar{X})(X_{t+k} - \bar{X})}{\sum_{t=1}^n (X_t - \bar{X})^2}, \quad (2.4.14)$$

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 [36]:

$$\phi_{kk} = Corr(X_t, X_{t+k} \mid X_{t+k-1}, X_{t+k-2}, \dots, X_{t+1}); k \geq 1. \quad (2.4.15)$$

Fungsi autokorelasi parsial biasanya tidak diketahui pada data populasi. Hal ini, dapat diduga dengan menggunakan fungsi autokorelasi parsial sampel yang dinyatakan sebagai berikut [36]:

$$\hat{\phi}_{k+1,k+1} = \frac{\hat{\rho}_{k+1} - \sum_{j=1}^k \hat{\phi}_{k,j} \hat{\rho}_{k+1-j}}{1 - \sum_{j=1}^k \hat{\phi}_{k,j} \hat{\rho}_j}, \quad (2.4.16)$$

dan

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

Fungsi autokorelasi parsial antara data X_t dan X_{t+k} pada Persamaan (2.4.15) diuraikan sebagai berikut [36]:

Misalkan data X_t adalah data yang stasioner dengan $E(X_t) = 0$.

Data X_{t+k} dapat dinyatakan sebagai model linier yaitu:

$$X_{t+k} = \phi_{k1} X_{t+k-1} + \phi_{k2} X_{t+k-2} + \dots + \phi_{kk} X_t + \varepsilon_{t+k}, \quad (2.4.17)$$

dengan ϕ_{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 (2.4.17), dengan X_{t+k-j} pada kedua ruas sehingga diperoleh

$$\begin{aligned} X_{t+k} X_{t+k-j} &= \phi_{k1} X_{t+k-1} X_{t+k-j} + \phi_{k2} X_{t+k-2} X_{t+k-j} + \dots \\ &+ \phi_{kk} X_t X_{t+k-j} + \varepsilon_{t+k} X_{t+k-j}. \end{aligned} \quad (2.4.18)$$

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

$$\begin{aligned} E(X_{t+k} X_{t+k-j}) &= \phi_{k1} E(X_{t+k-1} X_{t+k-j}) + \phi_{k2} E(X_{t+k-2} X_{t+k-j}) + \dots \\ &+ \phi_{kk} E(X_t X_{t+k-j}) + E(\varepsilon_{t+k} X_{t+k-j}), \end{aligned} \quad (2.4.19)$$


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

$$\gamma_j = \phi_{k1}\gamma_{j-1} + \phi_{k2}\gamma_{j-2} + \dots + \phi_{kk}\gamma_{j-k}. \quad (2.4.20)$$

Diketahui $\rho_j = \frac{\gamma_j}{\gamma_0}$ untuk $j = 0, 1, \dots, k$. Persamaan (2.4.20) 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:



$$\begin{aligned} \rho_1 &= \rho_0\phi_{k1} + \rho_1\phi_{k2} + \rho_2\phi_{k3} + \dots + \rho_{k-1}\phi_{kk} \\ \rho_2 &= \rho_1\phi_{k1} + \rho_0\phi_{k2} + \rho_1\phi_{k3} + \dots + \rho_{k-2}\phi_{kk} \\ \rho_3 &= \rho_2\phi_{k1} + \rho_1\phi_{k2} + \rho_0\phi_{k3} + \dots + \rho_{k-3}\phi_{kk} \\ &\vdots \\ \rho_k &= \rho_{k-1}\phi_{k1} + \rho_{k-2}\phi_{k2} + \rho_{k-3}\phi_{k3} + \dots + \rho_0\phi_{kk}. \end{aligned} \quad (2.4.21)$$

Diketahui $\rho_0 = 1$ sehingga Persamaan (2.4.21) dapat dituliskan dalam notasi matriks, yaitu

$$\begin{bmatrix} 1 & \rho_1 & \dots & \rho_{k-1} \\ \rho_1 & 1 & \dots & \rho_{k-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \dots & 1 \end{bmatrix} \begin{bmatrix} \phi_{k1} \\ \phi_{k2} \\ \vdots \\ \phi_{kk} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_k \end{bmatrix}. \quad (2.4.22)$$

Persamaan 2.4.22 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. [1] Jika $A\mathbf{x} = \mathbf{b}$ 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

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

Persamaan (2.4.22) dapat diselesaikan menggunakan Teorema 2.4.1

tentang aturan Cramer, dengan

$$A = \begin{bmatrix} 1 & \rho_1 & \dots & \rho_{k-1} & \phi_{k1} \\ \rho_1 & 1 & \dots & \rho_{k-2} & \phi_{k2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \dots & 1 & \phi_{kk} \end{bmatrix}, \mathbf{x} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_k \end{bmatrix}, \mathbf{b} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_k \end{bmatrix}, \quad (2.4.23)$$

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

$$\phi_{11} = \rho_1. \quad (2.4.24)$$

2. *Lag* kedua dengan $k = 2$.

Fungsi autokorelasi parsial pada *lag* kedua diperoleh sistem persamaan yaitu:

$$\rho_1 = \phi_{11}\rho_0 + \phi_{22}\rho_1 \quad (2.4.25)$$

$$\rho_2 = \phi_{11}\rho_1 + \phi_{22}\rho_0,$$

dan dapat ditulis dalam bentuk matriks, yaitu:

$$\begin{bmatrix} \rho_0 & \rho_1 \\ \rho_1 & \rho_0 \end{bmatrix} \begin{bmatrix} \phi_{11} \\ \phi_{22} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix},$$

dengan menggunakan aturan *Cramer* diperoleh,

$$\phi_{22} = \frac{\det(A_2)}{\det(A)} = \frac{\begin{vmatrix} 1 & \rho_1 \\ \rho_1 & \rho_2 \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{vmatrix}} = \frac{\rho_2 - \rho_1^2}{1 - \rho_1^2}. \quad (2.4.26)$$

3. *Lag* ketiga dengan $k = 3$.

Fungsi autokorelasi parsial pada *lag* ketiga diperoleh sistem persamaan yaitu:

$$\rho_1 = \phi_{11}\rho_0 + \phi_{22}\rho_1 + \phi_{33}\rho_2$$

$$\rho_2 = \phi_{11}\rho_1 + \phi_{22}\rho_0 + \phi_{33}\rho_1 \quad (2.4.27)$$

$$\rho_3 = \phi_{11}\rho_2 + \phi_{22}\rho_1 + \phi_{33}\rho_0,$$

dapat ditulis dalam bentuk matriks, yaitu:

$$\begin{bmatrix} \rho_0 & \rho_1 & \rho_2 \\ \rho_1 & \rho_0 & \rho_1 \\ \rho_2 & \rho_1 & \rho_0 \end{bmatrix} \begin{bmatrix} \phi_{11} \\ \phi_{22} \\ \phi_{33} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{bmatrix},$$

dengan menggunakan aturan *Cramer* diperoleh,

$$\begin{aligned}
\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}} \\
&= \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) - \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 - \rho_1\rho_2 + \left(\frac{\rho_2 - \rho_1^2}{1 - \rho_1^2}\right)\rho_1\rho_2 - \left(\frac{\rho_2 - \rho_1^2}{1 - \rho_1^2}\right)\rho_1}{1 - \rho_1^2 + \left(\frac{\rho_2 - \rho_1^2}{1 - \rho_1^2}\right)\rho_1^2 - \left(\frac{\rho_2 - \rho_1^2}{1 - \rho_1^2}\right)\rho_2} \\
&= \frac{\rho_3 - \left(\rho_1 - \left(\frac{\rho_2 - \rho_1^2}{1 - \rho_1^2}\right)\rho_1\right)\rho_2 - \left(\frac{\rho_2 - \rho_1^2}{1 - \rho_1^2}\right)\rho_1}{1 - \left(\rho_1 - \left(\frac{\rho_2 - \rho_1^2}{1 - \rho_1^2}\right)\rho_1\right)\rho_1 - \left(\frac{\rho_2 - \rho_1^2}{1 - \rho_1^2}\right)\rho_2} \\
&= \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_2} \\
&= \frac{\rho_3 - \phi_{21}\rho_2 - \phi_{22}\rho_1}{1 - \phi_{21}\rho_1 - \phi_{22}\rho_2}
\end{aligned}$$

$$= \frac{\rho_3 - \sum_{j=1}^2 \phi_{2,j} \rho_{3-j}}{1 - \sum_{j=1}^2 \phi_{2,j} \rho_j}$$

4. *Lag ke-k*, dengan menggunakan aturan *Cramer* diperoleh fungsi auto-korelasi parsial *lag ke-k* yaitu [36]:

$$\phi_{kk} = \frac{\begin{vmatrix} 1 & \rho_1 & \cdots & \rho_{k-2} & \rho_1 \\ \rho_1 & 1 & \cdots & \rho_{k-3} & \rho_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \cdots & \rho_1 & \rho_k \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 & \cdots & \rho_{k-2} & \rho_{k-1} \\ \rho_1 & 1 & \cdots & \rho_{k-3} & \rho_{k-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \cdots & \rho_1 & 1 \end{vmatrix}}, \quad (2.4.28)$$

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 [36].

2.5 Proses *White Noise*

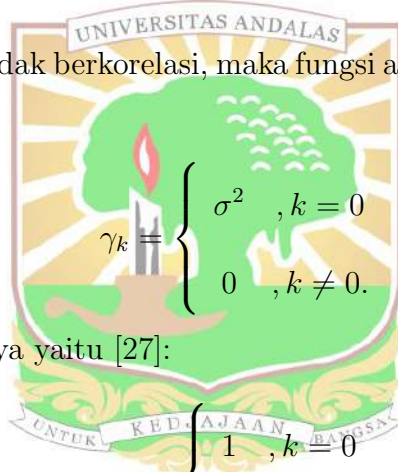
Proses *white noise* merupakan proses stasioner yang dianggap sebagai faktor pembangun bagi proses deret waktu lainnya. Proses *white noise* digunakan untuk menjelaskan perilaku data yang acak dan stasioner. Proses *white noise* dinotasikan sebagai berikut [27]:

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

Suatu data deret waktu $\{X_t\}$ disebut *white noise* apabila mempunyai sifat-sifat sebagai berikut [27]:

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 [27]:



$$\gamma_k = \begin{cases} \sigma^2, & k = 0 \\ 0, & k \neq 0. \end{cases} \quad (2.5.29)$$

Fungsi autokorelasinya yaitu [27]:

$$\rho_k = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0. \end{cases} \quad (2.5.30)$$

Fungsi autokorelasi parsialnya adalah sebagai berikut [27]:

$$\phi_{kk} = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0. \end{cases} \quad (2.5.31)$$

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 [33]:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + \varepsilon_t, \quad (2.6.32)$$

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 (2.6.32) dapat ditulis yaitu:

$$\begin{aligned} X_t &= \phi_1 BX_t + \phi_2 B^2 X_t + \dots + \phi_p B^p X_t + \varepsilon_t \\ X_t - \phi_1 BX_t - \phi_2 B^2 X_t - \dots - \phi_p B^p X_t &= \varepsilon_t \\ (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)X_t &= \varepsilon_t \end{aligned} \quad (2.6.33)$$

$$\phi_p(B)X_t = \varepsilon_t,$$

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 residu

pada orde q waktu sebelumnya. Bentuk umum model MA dengan orde q yang dinotasikan dengan $MA(q)$ sebagai berikut [33]:

$$X_t = \varepsilon_t - \theta_1\varepsilon_{t-1} - \theta_2\varepsilon_{t-2} - \dots - \theta_q\varepsilon_{t-q}, \quad (2.6.34)$$

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:

$$\begin{aligned} X_t &= \varepsilon_t - \theta_1 B\varepsilon_t - \theta_2 B^2\varepsilon_t - \dots - \theta_q B^q\varepsilon_t \\ &= (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q)\varepsilon_t \\ &= \theta_q(B)\varepsilon_t, \end{aligned} \quad (2.6.35)$$

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 [33]:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q}, \quad (2.6.36)$$

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 (2.6.36) dapat ditulis yaitu:

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)X_t = (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q)\varepsilon_t, \quad (2.6.37)$$

atau dapat ditulis dalam bentuk seperti dibawah ini:



$$\phi_p(B)X_t = \theta_q(B)\varepsilon_t. \quad (2.6.38)$$

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

Tabel 2.6.2: Pola Autokorelasi (ACF) dan Autokorelasi Parsial (PACF)

Model	ACF	PACF
ARMA $(p, 0)$	Menurun secara bertahap menuju nol	Menuju nol setelah <i>lag</i> ke- p
ARMA $(0, q)$	Menuju nol setelah <i>lag</i> 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 [33]:

$$(1 - B)^d X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} + \varepsilon_t, \quad (2.6.39)$$

atau dengan menggunakan operator *shift* mundur yaitu $BX_t = X_{t-1}$, Persamaan (2.6.39) dapat ditulis sebagai berikut:

$$\phi_p(B)(1 - B)^d X_t = \theta_q(B)\varepsilon_t, \quad (2.6.40)$$

dimana

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

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

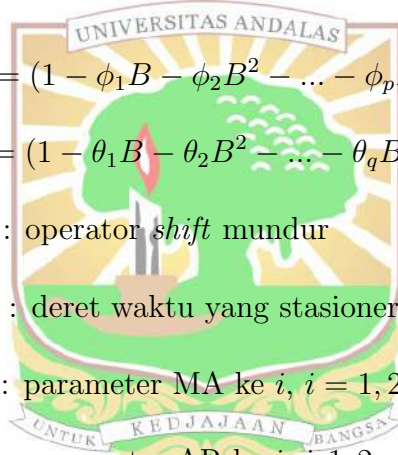
B : operator *shift* mundur

$(1 - B)^d X_t$: 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 *Auto-regressive Fractionally Integrated Moving Average (ARFIMA)*

Auto-regressive Fractionally Integrated Moving Average (ARFIMA) pertama kali dikemukakan oleh Grager dan Joyeux pada tahun 1980. Model ARFIMA merupakan pengembangan dari model ARIMA untuk memodelkan data jangka panjang (*long memory*). *Long memory* merupakan deret waktu

yang mempunyai korelasi tinggi diantara waktu pengamatan yang panjang. Hal ini terlihat dari, fungsi autokorelasi dimana lagnya turun secara lambat dalam periode yang panjang [7]. Model ARFIMA(p, d, q) dengan nilai p dan q adalah bilangan bulat non-negatif, serta nilai d berupa bilangan riil yang berada pada interval $0 < d < 0.5$ [18].

Bentuk umum dari model ARFIMA(p, d, q) sebagai berikut [36]:

$$(1 - B)^d X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} + \varepsilon_t. \quad (2.6.41)$$

Dengan menggunakan operator *shift* mundur yaitu $BX_t = X_{t-1}$, Persamaan (2.6.41) dapat ditulis sebagai berikut:

$$\phi_p(B)(1 - B)^d X_t = \theta_q(B)\varepsilon_t, \quad (2.6.42)$$

dimana

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

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

B : operator *shift* mundur

d : parameter *differencing* (bilangan riil)

$(1 - B)^d X_t$: 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 , $\varepsilon_t \sim WN(0, \sigma^2)$.

Operator *fractionally difference* $(1 - B)^d$ merupakan perluasan dari binomial yang didefinisikan sebagai berikut [18]:

$$(1 - B)^d = \sum_{k=1}^{\infty} \binom{d}{k} (-1)^k B^k, \quad (2.6.43)$$

dengan

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

sehingga

$$\begin{aligned} (1 - B)^d &= \binom{d}{0} (-1)^0 B^0 + \binom{d}{1} (-1)^1 B^1 + \binom{d}{2} (-1)^2 B^2 + \dots \\ (1 - B)^d &= \frac{d!}{0!(d-0)!} B^0 + \frac{d!}{1!(d-1)!} B^1 + \frac{d!}{2!(d-2)!} B^2 + \dots \\ &= 1 - dB - \frac{d(1-d)B^2}{2} - \frac{d(1-d)(2-d)B^3}{6} + \dots \end{aligned} \quad (2.6.44)$$

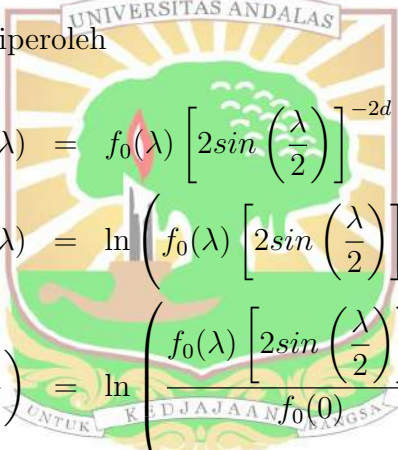
Estimasi parameter *differencing* (d) menggunakan metode *Geweke and Porter Hudak* (GPH). Metode GPH diusulkan oleh Geweke dan Porter-Hudak pada tahun 1983. Parameter *differencing* (d) diestimasi dengan metode regresi kuadrat terkecil, yang diperoleh dari penaksiran persamaan logaritma *spectral density*. Kelebihan metode GPH yaitu fleksibel dalam estimasi parameternya. Estimasi parameter *differencing* (d) dapat dilakukan tanpa mengetahui nilai orde p dan q terlebih dahulu.

Pada data deret waktu, terdapat dua domain yaitu domain waktu dan domain frekuensi. Domain waktu pada data deret waktu menjelaskan korelasi data, stasioneritas data, taksiran parameter hingga pemodelan data. Pada domain frekuensi terdapat periodisitas dari data deret waktu. Hal ini dapat ditentukan dengan analisis *spectral density*. Analisis *spectral density* merupakan suatu metode analisis dalam penaksiran domain frekuensi, untuk menelaah periodisitas tersembunyi, yang sulit ditemukan pada domain waktu. Pada analisis *spectral density* menggunakan periodogram data deret waktu,

yang dalam estimasinya melalui frekuensi *fourier*. *Spectral density* merupakan sebuah fungsi riil positif, dimana variabel frekuensinya dihubungkan dengan fungsi deterministik dari waktu. *Spectral density* didefinisikan adalah sebagai berikut [22, 35]:

$$\begin{aligned} f(\lambda) &= \frac{\sigma_\varepsilon^2 |\theta(e^{-i\lambda})|^2}{2\pi |\phi(e^{-i\lambda})|^2} \left[2\sin\left(\frac{\lambda}{2}\right) \right]^{-2d} \\ &= f_0(\lambda) \left[2\sin\left(\frac{\lambda}{2}\right) \right]^{-2d}, \end{aligned} \quad (2.6.45)$$

dengan $f(\lambda)$ adalah *spectral density*, $f_0(\lambda)$ adalah *spectral density* dari ε_t dan λ adalah frekuensi dari periodogram. Dengan melogaritmakan kedua sisi dari Persamaan (2.6.45) diperoleh



$$\begin{aligned} f(\lambda) &= f_0(\lambda) \left[2\sin\left(\frac{\lambda}{2}\right) \right]^{-2d} \\ \ln f(\lambda) &= \ln \left(f_0(\lambda) \left[2\sin\left(\frac{\lambda}{2}\right) \right]^{-2d} \right) \\ \ln \left(\frac{f_0(\lambda)}{f_0(0)} \right) &= \ln \left(\frac{f_0(\lambda) \left[2\sin\left(\frac{\lambda}{2}\right) \right]^{-2d}}{f_0(0)} \right) \\ \ln f(\lambda) - \ln f_0(0) &= \ln \left(\left[2\sin\left(\frac{\lambda}{2}\right) \right]^{-2d} \frac{f_0(\lambda)}{f_0(0)} \right) \\ \ln f(\lambda) &= \ln f_0(0) + \ln \left(\frac{f_0(\lambda)}{f_0(0)} \right) - d \ln \left[2\sin\left(\frac{\lambda}{2}\right) \right]^2 \\ \ln f(\lambda) &= \ln f_0(\lambda) - d \ln \left[2\sin\left(\frac{\lambda}{2}\right) \right]^2. \end{aligned} \quad (2.6.46)$$

Diketahui $I(\lambda_j)$ adalah periodogram dengan frekuensi *fourier* $\lambda_j = \frac{2\pi j}{T}$, dengan $j = 1, 2, 3, \dots, m$. Notasi T adalah jumlah pengamatan dan m adalah jumlah frekuensi *fourier*. Periodogram $I(\lambda_j)$ didefinisikan sebagai [35]:

$$I(\lambda_j) = \frac{1}{2\pi} \left\{ \gamma_0 + 2 \sum_{t=1}^{T-1} \gamma_t \cos(t.\lambda_j) \right\}. \quad (2.6.47)$$

Logaritma dari periodogram $I(\lambda_j)$ dapat ditulis sebagai:

$$\begin{aligned}\ln I(\lambda_j) &= \ln f(\lambda_j) + \ln \left[\frac{I(\lambda_j)}{f(\lambda_j)} \right] \\ \ln I(\lambda_j) &= \ln f_0(0) - d \ln \left[2 \sin \left(\frac{\lambda}{2} \right) \right]^2 + \ln \left[\frac{f_0(\lambda_j)}{f_0(0)} \right] + \ln \left[\frac{I(\lambda_j)}{f(\lambda_j)} \right] \\ \ln I(\lambda_j) &= \ln f_0(0) - d \ln \left[2 \sin \left(\frac{\lambda}{2} \right) \right]^2 + \ln \left[\frac{I(\lambda_j) [2 \sin \left(\frac{\lambda}{2} \right)]^{2d}}{f_0(0)} \right],\end{aligned}$$

dimana

$$\begin{aligned}y_j &= \ln I(\lambda_j), \quad \alpha = \ln f_0(0), \quad x_j = -\ln \left[2 \sin \left(\frac{\lambda_j}{2} \right) \right]^2, \\ \varepsilon_j &= \ln \left[\frac{I(\lambda_j) [2 \sin \left(\frac{\lambda}{2} \right)]^{2d}}{f_0(0)} \right],\end{aligned}$$

diperoleh persamaan regresi sebagai berikut:

$$y_j = \alpha + dx_j + \varepsilon_j. \quad (2.6.48)$$

Berdasarkan Persamaan (2.6.48), dengan menggunakan metode *Ordinary Least Square* (OLS), maka estimasi parameter d dengan metode *Geweke and Porter Hudak* disimbolkan dengan \hat{d}_{GPH} yaitu [35]:

$$\hat{d}_{GPH} = \frac{\sum_{j=1}^m (x_j - \bar{x})(y_j - \bar{y})}{\sum_{j=1}^m (x_j - \bar{x})^2}. \quad (2.6.49)$$

Fungsi kovarian dari model ARFIMA adalah sebagai berikut [35]:

$$\begin{aligned}\gamma_k &= \int_{-\pi}^{\pi} e^{ik\lambda} f(\lambda) d(\lambda) \\ &= (-1)^k \sigma_\varepsilon^2 \frac{\Gamma(1-2d)}{\Gamma(k-d+1)\Gamma(1-d-k)}.\end{aligned} \quad (2.6.50)$$

Fungsi *Autocorrelation Function* (ACF) model ARFIMA yaitu [35]:

$$\begin{aligned}\rho_k &= \frac{\gamma_k}{\gamma_0} \\ &= \frac{(-1)^k \Gamma(1-2d)}{\Gamma(k-d+1)\Gamma(1-d-k)} \frac{\Gamma(1-d)\Gamma(1-d)}{\Gamma(1-2d)},\end{aligned} \quad (2.6.51)$$

$$\begin{aligned}
\rho_k &= \frac{\Gamma(1-d)}{\Gamma(k-d+1)} \frac{(-1)^k(1-d-1)(1-d-2)\dots(1-d-k)\Gamma(1-d-k)}{\Gamma(1-d-k)} \\
&= \frac{\Gamma(1-d)}{\Gamma(k-d+1)} (-1)^k (-1)^k (d)(d+1)\dots(d+k-1) \\
&= \frac{\Gamma(1-d)}{\Gamma(d)} \frac{\Gamma(k+d)}{\Gamma(k-d+1)}. \tag{2.6.52}
\end{aligned}$$

Fungsi autokorelasi *Partial Autocorrelation Function* (PACF) dari model ARFIMA adalah sebagai berikut [22]:

$$\begin{aligned}
\phi_{kj} &= -\binom{k}{j} \frac{\Gamma(j-d)\Gamma(k-d-j+1)}{\Gamma(-d)\Gamma(k-d+1)} \\
\phi_{kk} &= -\binom{k}{k} \frac{\Gamma(k-d)\Gamma(k-d-k+1)}{\Gamma(-d)\Gamma(k-d+1)} \\
&= \frac{\Gamma(k-d)\Gamma(1-d)}{\Gamma(k-d+1)\Gamma(-d)} \\
&= \frac{\Gamma(k-d)}{(k-d)\Gamma(k-d)} \frac{\Gamma(1-d)}{\Gamma(-d)} \\
&= -\frac{1}{k-d} (-d) \frac{\Gamma(-d)}{\Gamma(-d)} \\
&= \frac{d}{(k-d)}, \tag{2.6.53}
\end{aligned}$$

dengan $\phi_{kk} \sim \frac{d}{k}$ untuk nilai k yang besar.

2.6.6 Pendugaan Parameter

Pendugaan parameter merupakan perhitungan yang dilakukan untuk mendapatkan nilai parameter suatu model. Pada penelitian ini digunakan metode *maximum likelihood estimation* untuk menduga parameter dari model. Berikut diberikan definisi fungsi kemungkinan (*likelihood*) dan nilai penduga kemungkinan maksimum.

Definisi 2.6.1. [2] Fungsi kepekatan peluang bersama dari peubah acak X_1, X_2, \dots, X_n yang dihitung pada nilai x_1, x_2, \dots, x_n yaitu $f(x_1, x_2, \dots, x_n, \theta)$,

disebut sebagai fungsi kemungkinan. Untuk x_1, x_2, \dots, x_n yang tetap, fungsi likelihood adalah fungsi dari θ dan dinotasikan dengan $L(\theta)$. Jika X_1, X_2, \dots, X_n merupakan sampel acak dari $f(x; \theta)$, maka

$$L(\theta) = f(x_1; \theta)f(x_2; \theta) \cdots f(x_n; \theta). \quad (2.6.54)$$

Definisi 2.6.2. [2] Misalkan $L(\theta) = f(x_1; \theta)f(x_2; \theta) \cdots f(x_n; \theta), \theta \in \Omega$ adalah fungsi kemungkinan. Suatu nilai pengamatan dari x_1, x_2, \dots, x_n , nilai dari $\hat{\theta}$ dalam Ω yang membuat $L(\theta)$ maksimum disebut dengan nilai penduga kemungkinan maksimum bagi θ yang memenuhi,

$$f(x_1, x_2, \dots, x_n, \hat{\theta}) = \max_{\theta \in \Omega}. \quad (2.6.55)$$

Jika x_1, x_2, \dots, x_n merupakan nilai-nilai pengamatan yang diambil dari $f(x; \theta)$, maka

$$\begin{aligned} L(\theta) &= f(x_1, x_2, \dots, x_n; \theta) \\ &= f(x_1; \theta)f(x_2; \theta) \cdots f(x_n; \theta) \\ &= \prod_{i=1}^n f(x_i; \theta). \end{aligned} \quad (2.6.56)$$

Nilai θ diperoleh dengan memaksimumkan fungsi kemungkinan, yaitu dengan menyelesaikan persamaan berikut:

$$\frac{dL(\theta)}{d\theta} = 0. \quad (2.6.57)$$

Kesulitan dalam menemukan turunan fungsi kemungkinan pada Persamaan (2.6.57), dapat diselesaikan melalui *logaritma natural*, yaitu:

$$\ln(L(\theta)) = \sum_{i=1}^n \ln(f(x_i; \theta)). \quad (2.6.58)$$

Maksimum fungsi $\ln(L(\theta))$ diperoleh dengan cara, menurunkan Persamaan (2.6.58) terhadap θ dan menyamakan dengan nol, yaitu:

$$\frac{d\ln(L(\theta))}{d\theta} = 0. \quad (2.6.59)$$

Perhatikan model ARMA(p,q) berikut:

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

dengan asumsi ε_t merupakan proses *white noise*, berdistribusi normal dengan rata-rata nol dan standar deviasi σ ($\varepsilon_t \sim N(0, \sigma^2)$) [11]. Fungsi kepekatan peluang untuk setiap ε_t adalah

$$f(\varepsilon_t; \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\varepsilon_t^2}{2\sigma^2}\right) \text{ untuk } -\infty < \varepsilon_t < \infty. \quad (2.6.60)$$

Perhatikan model ARIMA(p, d, q) berikut:

$$\phi_p(B) (1 - B)^d X_t = \theta_q(B) \varepsilon_t, \quad (2.6.61)$$

dengan parameter yang akan diestimasi diberikan dalam bentuk vektor yaitu $\omega = (\phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q)$. Fungsi *likelihood* untuk model ARIMA adalah [7]:

$$L(\omega) = \prod_{t=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2} \frac{\varepsilon_t^2}{\sigma^2}\right), \quad (2.6.62)$$

dengan $\varepsilon_t = \frac{\phi_p(B)(1 - B)^d X_t}{\theta_q(B)}$.

Kemudian, fungsi *log-likelihood* Persamaan (2.6.62) dapat dituliskan menjadi:

$$\begin{aligned} \ln(L(\omega)) &= \ln\left(\prod_{t=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2} \frac{\varepsilon_t^2}{\sigma^2}\right)\right) \\ &= \sum_{t=1}^n \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2} \frac{\varepsilon_t^2}{\sigma^2}\right)\right) \\ &= \sum_{t=1}^n \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) + \sum_{t=1}^n \ln\left(\exp\left(-\frac{1}{2} \frac{\varepsilon_t^2}{\sigma^2}\right)\right) \end{aligned} \quad (2.6.63)$$

$$\begin{aligned}
\ln(L(\boldsymbol{\omega})) &= -\frac{n}{2} \ln 2\pi - \frac{1}{2} \sum_{t=1}^n \ln \sigma^2 - \frac{1}{2} \sum_{t=1}^n \frac{\varepsilon_t^2}{\sigma^2} \\
&= -\frac{n}{2} \ln 2\pi - \frac{n}{2} \ln \sigma^2 - \frac{1}{2\sigma^2} \sum_{t=1}^n \varepsilon_t^2 \\
&= -\frac{n}{2} \ln 2\pi - \frac{n}{2} \ln \sigma^2 - \frac{1}{2\sigma^2} \sum_{t=1}^n \left(\frac{\phi_p(B)(1-B)^d X_t}{\theta_q(B)} \right)^2.
\end{aligned}$$

Parameter $\boldsymbol{\omega}$ diestimasi dengan memaksimalkan fungsi *likelihood* pada Persamaan (2.6.64), yaitu:

$$\frac{\partial \ln(L(\boldsymbol{\omega}))}{\partial \boldsymbol{\omega}} = \frac{\partial \left[-\frac{n}{2} \ln 2\pi - \frac{n}{2} \ln \sigma^2 - \frac{1}{2\sigma^2} \sum_{t=1}^n \left(\frac{\phi_p(B)(1-B)^d X_t}{\theta_q(B)} \right)^2 \right]}{\partial \boldsymbol{\omega}} = 0.$$

2.7 Uji Signifikansi Parameter

Model yang dapat digunakan adalah model yang memiliki parameter signifikan. Parameter model yang signifikan atau tidak dapat diketahui melalui uji signifikansi parameter [16]. Misalkan θ adalah parameter yang akan diuji, maka hipotesisnya sebagai berikut:

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

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

Statistik uji yang digunakan yaitu:

$$t_{hitung} = \frac{\hat{\theta}}{SE(\hat{\theta})}, \quad (2.7.64)$$

dengan

$\hat{\theta}$: estimasi parameter

$SE(\hat{\theta})$: standar *error* estimasi parameter.

Kesimpulan tolak H_0 , jika $|t_{hitung}| > t_{\frac{\alpha}{2}, n-k}$ atau $p\text{-value} < \alpha$ yang berarti parameter signifikan. Model sementara dapat digunakan untuk pemo-

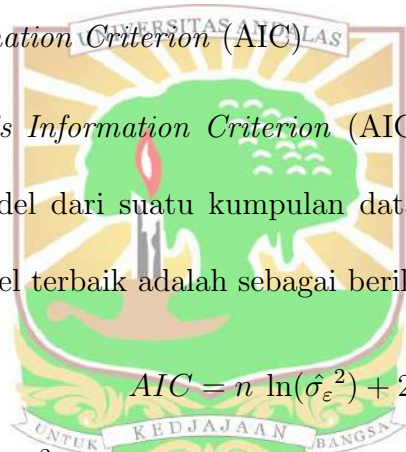
delan, dengan n adalah banyak pengamatan, dan k banyak parameter dalam model.

2.8 Pemilihan Model Terbaik

Pemilihan model terbaik dilakukan jika terdapat lebih dari satu jenis model yang diperoleh. Oleh karena itu, diperlukan kriteria untuk memilih model terbaik diantara beberapa jenis model tersebut. Kriteria pemilihan model terbaik diantaranya adalah sebagai berikut [36]:

1. *Akaike's Information Criterion* (AIC)

Akaike's Information Criterion (AIC) digunakan untuk mengidentifikasi model dari suatu kumpulan data. Persamaan AIC dalam pemilihan model terbaik adalah sebagai berikut:



$$AIC = n \ln(\hat{\sigma}_\varepsilon^2) + 2k, \quad (2.8.65)$$

dimana $\hat{\sigma}_\varepsilon^2$: Penduga *likelihood* maksimum dari σ_ε^2

k : jumlah parameter

n : banyak data pengamatan.

Nilai penduga *likelihood* maksimum juga dapat dihitung dengan

$$\hat{\sigma}_\varepsilon^2 = \sum_{t=k+1}^n \frac{\hat{\varepsilon}_t^2}{(n-k)},$$

dimana $\hat{\varepsilon}_t$: residu dugaan

k : jumlah parameter

n : banyak data pengamatan.

Dengan demikian, model dikatakan baik jika nilai AIC semakin kecil.

2. *Bayesian Information Criterion (BIC)*

Bayesian Information Criterion (BIC) merupakan metode yang menggunakan pendekatan *Penalized Maximum Likelihood* dalam pemilihan model terbaik. Persamaan BIC dalam pemilihan model adalah sebagai berikut:

$$BIC = n \ln(\hat{\sigma}_\varepsilon^2) + k \ln(n), \quad (2.8.66)$$

dimana $\hat{\sigma}_\varepsilon^2$: Penduga *likelihood* maksimum dari σ_ε^2

k : jumlah parameter

n : banyak data pengamatan.

Dapat disimpulkan bahwa model dikatakan baik jika nilai dari BIC semakin kecil.



2.9 Uji Asumsi Residu

Setelah melakukan pemilihan model terbaik, langkah selanjutnya dilakukan uji asumsi-asumsi residu pemodelan yaitu:

1. Uji Non-Autokorelasi

Uji non-autokorelasi dengan menggunakan uji *QLjung-Box* digunakan untuk melihat apakah terdapat autokorelasi atau tidak pada suatu data. Persamaan untuk menghitung uji *QLjung-Box* adalah sebagai

berikut:

$$Q_{LB} = n(n+2) \sum_{i=1}^k \frac{\rho_i^2}{n-i}, \quad (2.9.67)$$

dimana Q_{LB} : nilai *QLjung-Box*

n : banyak data pengamatan

k : banyak *lag* yang diperhatikan

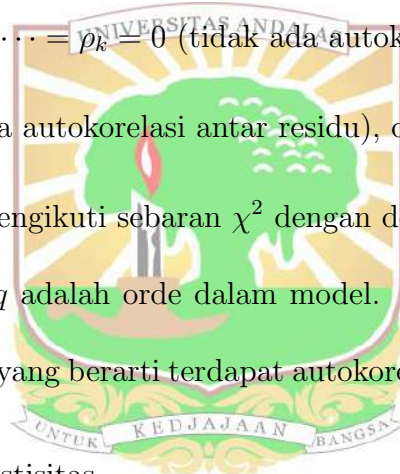
ρ_i : autokorelasi residu ke- i , $i = 1, 2, \dots, k$.

Hipotesis yang digunakan pada uji *QLjung-Box* ini adalah sebagai berikut:

$H_0 : \rho_1 = \rho_2 = \dots = \rho_k = 0$ (tidak ada autokorelasi antar residu).

$H_1 : \rho_i \neq 0$ (ada autokorelasi antar residu), dengan $i = 1, 2, \dots, k$,

dengan Q_{LB} mengikuti sebaran χ^2 dengan derajat kebebasan $k - p - q$, dengan p dan q adalah orde dalam model. Jika $Q_{LB} > \chi_{\alpha}^2(k - p - q)$, maka tolak H_0 yang berarti terdapat autokorelasi pada residu model [16].



2. Uji Homoskedastisitas

Uji *Lagrange Multiplier* merupakan uji yang digunakan untuk melihat terdapat homoskedastisitas atau tidak, pada suatu data deret waktu. Uji *Lagrange Multiplier* dilakukan dengan meregresikan residu kuadrat dengan model berikut:

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

Kemudian melakukan estimasi koefisien regresinya. Hipotesis dari uji ini adalah:

$H_0 : \alpha_1 = \alpha_2 = \dots = \alpha_k = 0$ (terdapat efek homoskedastisitas).

$H_1 : \alpha_i \neq 0$ (tidak terdapat efek homoskedastisitas) dengan $i = 1, 2, \dots, k$.

Persamaan untuk menghitung uji *Lagrange Multiplier* adalah sebagai berikut:

$$LM = nR^2, \quad (2.9.68)$$

dimana LM : nilai *Lagrange Multiplier*

n : banyak data pengamatan

R^2 : koefisien determinasi model regresi residu kuadrat.

Nilai *Lagrange Multiplier* mengikuti sebaran χ^2 , dengan derajat kebebasan sebesar 2. Jika $LM > \chi^2_{\alpha}(2)$, maka tolak H_0 berarti tidak terdapat efek homoskedastisitas pada residu model [16].

3. Uji Normalitas

Uji normalitas merupakan uji yang digunakan untuk menentukan data deret waktu berdistribusi normal atau tidak. Uji normalitas dapat menggunakan hipotesis *Jarque-Bera*. Persamaan untuk menghitung uji *Jarque-Bera* adalah sebagai berikut:

$$JB = \frac{n}{6} \left(S^2 + \frac{(K-3)^2}{4} \right), \quad (2.9.69)$$

dimana JB : nilai *Jarque-Bera*

n : banyak data pengamatan

$$S = \frac{\frac{1}{n} (\varepsilon_i - \bar{\varepsilon})^3}{\left(\frac{1}{n} \sum_{i=1}^n (\varepsilon_i - \bar{\varepsilon})^2 \right)^{\frac{3}{2}}}$$

$$K = \frac{\frac{1}{n} \sum_{i=1}^n (\varepsilon_i - \bar{\varepsilon})^4}{\left(\frac{1}{n} \sum_{i=1}^n (\varepsilon_i - \bar{\varepsilon})^2 \right)^2}$$

$$\bar{\varepsilon} = \frac{\sum_{i=1}^n \varepsilon_i}{n}.$$

Semakin besar nilai statistik JB , maka semakin kecil kemungkinan data berdistribusi normal. Hipotesis yang digunakan pada uji *Jarque-Bera* ini yaitu:

H_0 : data berdistribusi normal.

H_1 : data tidak berdistribusi normal,

dengan nilai *Jarque-Bera* mengikuti sebaran χ^2 dengan derajat kebebasan 2. Jika nilai $JB > \chi^2_{\alpha}(2)$, maka tolak H_0 berarti data tidak berdistribusi normal [16].



2.10 Himpunan *Fuzzy*

Himpunan *fuzzy* merupakan himpunan yang digunakan untuk mengatasi kelemahan dari himpunan tegas (*crisp*). Himpunan tegas merupakan himpunan yang nilai derajat keanggotaan dari elemen hanya mempunyai dua kemungkinan yaitu 0 dan 1. Nilai 0 menunjukkan bahwa elemen x tidak terdapat di dalam himpunan. Nilai 1 menunjukkan elemen x terdapat dalam himpunan. Jika pada himpunan tegas nilai derajat keanggotaan hanya 0 dan 1, maka pada himpunan *fuzzy* nilai derajat keanggotaan berada pada interval $[0,1]$. Dengan kata lain, 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 seperti 0.25, 0.5, 0.8, ... dst [20].

Beberapa hal yang berkaitan dengan *fuzzy* yaitu [20]:

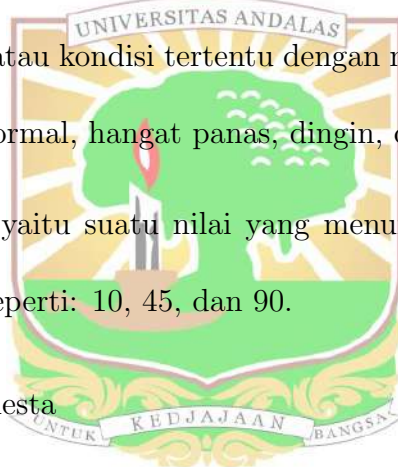
1. Variabel *Fuzzy*

Variabel *fuzzy* merupakan variabel yang akan dibahas dalam suatu sistem *fuzzy*.

2. Himpunan *Fuzzy*

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

- (a) Linguistik, yaitu penamaan suatu kelompok yang mewakili suatu keadaan, atau kondisi tertentu dengan menggunakan bahasa alami, seperti: normal, hangat panas, dingin, dan sejuk.
- (b) Numerik, yaitu suatu nilai yang menunjukkan ukuran dari suatu variabel seperti: 10, 45, dan 90.



3. Himpunan Semesta

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

4. Domain Himpunan *Fuzzy*

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

5. Fungsi Keanggotaan

Fungsi keanggotaan merupakan suatu kurva yang menunjukkan, pemetaan titik-titik data ke dalam nilai derajat keanggotaan, yang berada pada interval antara 0 sampai 1.

2.11 Rantai Markov (*Markov Chain*)

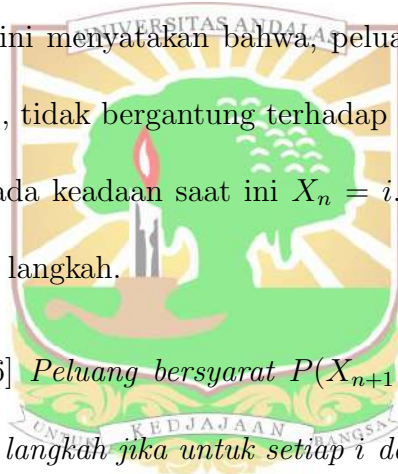
Rantai markov pertama kali diperkenalkan pada tahun 1906. Awalnya, penerapan rantai Markov digunakan untuk menganalisis dan memperkirakan perilaku partikel-partikel gas dalam suatu wadah tertutup. Selain itu, rantai Markov digunakan untuk meramalkan keadaan cuaca. Sifat dari rantai Markov, apabila suatu 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. Dengan kata lain, 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 2.11.1. [26] Proses stokastik $X_n(t)$ merupakan serangkaian peubah acak yang berubah terhadap waktu pengamatan $t \in T$.

Definisi 2.11.2. [26] Suatu proses stokastik $X_n(t)$ dikatakan memiliki sifat Markov jika $P(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = P(X_{n+1} = j \mid X_n = i)$ untuk waktu $n = 0, 1, \dots$ dan untuk setiap $j, i, i_{n-1}, \dots, i_1, i_0$.

Proses stokastik yang bersifat Markov disebut rantai Markov (*Markov Chain*). Peluang dari $X_{n+1} = j$ yang dinotasikan sebagai $P(X_{n+1} = j)$, mempunyai arti bahwa proses berada pada keadaan ke- j dan pada waktu ke $n + 1$. Sifat rantai Markov ini menyatakan bahwa, peluang bersyarat dari kejadian mendatang $X_{n+1} = j$, tidak bergantung terhadap kejadian di waktu lalu, dan hanya bergantung pada keadaan saat ini $X_n = i$. Berikut diberikan definisi peluang transisi satu langkah.



Definisi 2.11.3. [26] Peluang bersyarat $P(X_{n+1} = j \mid X_n = i)$ dikatakan peluang transisi satu langkah jika untuk setiap i dan j :

$$P(i, j) = P(X_{n+1} = j \mid X_n = i), \quad (2.11.70)$$

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 $P(i, j)$. Persamaan (2.11.70) 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:

$$P(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = p(i, j). \quad (2.11.71)$$

Peluang transisi satu langkah dapat dituliskan dalam bentuk matriks transisi. Matriks transisi dari rantai markov X_n , dengan $n = 0, 1, 2, \dots, N$ dengan ruang keadaan $(0, 1, 2, \dots, N)$, dan peluang transisi dinotasikan dengan,

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

Peluang transisi $p(i, j)$ harus memenuhi sifat sebagai berikut:

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



2.12 Fuzzy Time Series Markov Chain (FTSMC)

Fuzzy Time Series (FTS) merupakan pendekatan baru untuk menyelesaikan masalah data linguistik. Pendekatan ini menggabungkan variabel linguistik dengan analisis penerapan logika *fuzzy* ke dalam data deret waktu. Berikut ini diberikan definisi *fuzzy time series* yaitu:

Definisi 2.12.1. [32] Misalkan himpunan semesta Y_t ($t = \dots, 0, 1, 2, \dots, n, \dots$) adalah subset dari \mathfrak{R} yang didefinisikan dengan himpunan fuzzy A_i . Jika F_t

terdiri dari A_i ($i = 1, 2, \dots, n$), F_t didefinisikan sebagai FTS pada Y_t ($t = \dots, 0, 1, 2, \dots, n, \dots$).

Pada tahun 2012, Tsaur mengemukakan model *Fuzzy Time Series Markov Chain* (FTSMC). Model ini menggabungkan model *fuzzy time series* dengan rantai Markov. Berikut ini diuraikan langkah-langkah untuk menyelesaikan model *Fuzzy Time Series Markov Chain* [32]:

Langkah 1: Pendefinisian Himpunan Semesta U .

Himpunan semesta U diperoleh dengan menetapkan nilai minimum (D_{min}) dan maksimum (D_{max}) dari data pengamatan. Berikutnya, nilai D_1 dan D_2 ditentukan secara bebas, yang bertujuan untuk mempermudah dalam pembentukan interval. Berikut adalah rumus himpunan semesta U :

$$U = [D_{min} - D_1, D_{max} + D_2], \quad (2.12.72)$$

dimana D_{min} adalah nilai minimum, D_{max} adalah nilai maksimum, serta D_1 dan D_2 adalah bilangan riil yang sesuai.

Langkah 2: Pembentukan Interval.

Himpunan semesta U dipartisi menjadi beberapa bagian dengan interval (n) yang sama. Dengan menggunakan rumus *Sturges* yaitu [30]:

$$n = 1 + 3.322 \log N, \quad (2.12.73)$$

dengan N adalah banyaknya data pengamatan.

Langkah selanjutnya, menentukan panjang interval agar interval tersebut memiliki panjang yang sama. Berikut adalah rumus dalam penentuan

panjang interval:

$$l = \frac{[(D_{max} + D_2) - (D_{min} - D_1)]}{n}, \quad (2.12.74)$$

dimana l adalah panjang interval dan n adalah banyak interval.

Setiap interval dapat dihitung dengan cara sebagai berikut:

$$\begin{aligned} u_1 &= [D_{min} - D_1, D_{min} - D_1 + l] \\ u_2 &= [D_{min} - D_1 + l, D_{min} - D_1 + 2l] \\ &\vdots \\ u_n &= [D_{min} - D_1 + (n - 1)l, D_{min} - D_1 + nl]. \end{aligned} \quad (2.12.75)$$

Langkah 3: Pendefinisian himpunan *fuzzy* untuk masing-masing interval.

Penentuan himpunan *fuzzy* untuk masing-masing interval dapat dijelaskan dengan beberapa aturan sebagai berikut [5]:

1. Jika data pengamatan (Y_t) adalah u_i , maka derajat keanggotaan u_i adalah 1, u_{i+1} adalah 0.5 dan lainnya adalah 0.
2. Jika data pengamatan (Y_t) adalah u_i , $1 < i < n$, maka derajat keanggotaan u_i adalah 1, u_{i-1} dan u_{i+1} adalah 0.5 dan lainnya adalah 0.
3. Jika data pengamatan (Y_t) adalah u_n , maka derajat keanggotaan u_n adalah 1, u_{n-1} adalah 0.5 dan lainnya adalah 0.

Berdasarkan aturan-aturan tersebut, himpunan *fuzzy* A_1, A_2, \dots, A_n didefinisikan sebagai berikut [32]:

$$\begin{aligned} A_1 &= \left\{ \frac{1}{u_1}, \frac{0.5}{u_2}, \frac{0}{u_3}, \frac{0}{u_4}, \dots, \frac{0}{u_n} \right\} \\ A_2 &= \left\{ \frac{0.5}{u_1}, \frac{1}{u_2}, \frac{0.5}{u_3}, \frac{0}{u_4}, \dots, \frac{0}{u_n} \right\} \end{aligned}$$

$$A_3 = \left\{ \frac{0}{u_1}, \frac{0.5}{u_2}, \frac{1}{u_3}, \frac{0.5}{u_4}, \dots, \frac{0}{u_n} \right\}$$

$$\vdots$$
(2.12.76)

$$A_n = \left\{ \frac{0}{u_1}, \frac{0}{u_2}, \frac{0}{u_3}, \dots, \frac{0.5}{u_{n-1}}, \frac{1}{u_n} \right\}.$$

Langkah 4: *Fuzzyfikasi* terhadap data pengamatan.

Fuzzyfikasi merupakan proses mengidentifikasi variabel linguistik se-tiap data pengamatan. Jika suatu data pengamatan interval linguistiknya u_i , maka variabel linguistik dari data tersebut adalah A_i .

Langkah 5: Penentuan *Fuzzy Logical Relationship* (FLR) dan *Fuzzy Logical Relationship group* (FLRG) berdasarkan definisi berikut:

Definisi 2.12.2. [32] *Jika $F_t = A_i$ dan $F_{t-1} = A_j$ maka hubungan antara F_t dan F_{t-1} disebut sebagai Fuzzy Logical Relationship (FLR). Hubungan ini dapat dinyatakan dengan $A_i \rightarrow A_j$, dimana A_i disebut Left-Hand Side (LHS) dan A_j disebut Right-Hand Side (RHS) dari FLR. Jika terdapat dua FLR mempunyai himpunan fuzzy yang sama (LHS $A_i \rightarrow A_{j1}, A_i \rightarrow A_{j2}$), maka dapat dikelompokkan ke dalam fuzzy Logical Relationship Group (FLRG) $A_i \rightarrow A_{j1}, A_{j2}$.*

Langkah 6: Pembentukan matriks probabilitas transisi *Markov*.

Persamaan (2.11.71) diinterpretasikan ke dalam rantai markov sebagai peluang bersyarat dari *state* yang akan datang, yaitu X_{n+1} . Nilai X_{n+1} tidak bergantung pada *state* sebelumnya yaitu X_0, X_1, \dots, X_{n-1} , melainkan bergantung pada *state* yang sekarang yaitu X_n .

Jika *state* A_i bertransisi dengan *state* A_j serta melewati *state* lainnya A_k , $i, j, k = 1, 2, 3, \dots, n$, maka suatu FLRG terbentuk. Pembentukan FLRG

dengan cara mengelompokkan setiap perpindahan *state*, yaitu *state* saat ini (*current state*) dan *state* selanjutnya (*next state*). *Current state* merupakan nilai yang akan dihitung sebagai nilai pemodelan. *Next state* merupakan data yang akan diolah untuk memperoleh nilai pada *current state*. FLRG digunakan untuk memperoleh probabilitas transisi pada *state*. Probabilitas transisi untuk *state* tersebut dapat dituliskan sebagai berikut [32]:

$$P_{ij} = \frac{M_{ij}}{M_i}, \quad i, j = 1, 2, 3, \dots, n, \quad (2.12.77)$$

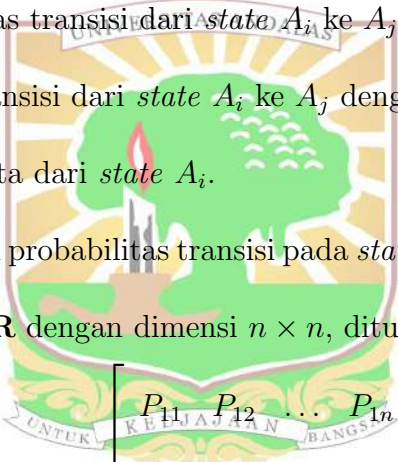
dengan

P_{ij} adalah probabilitas transisi dari *state* A_i ke A_j dengan satu langkah.

M_{ij} adalah waktu transisi dari *state* A_i ke A_j dengan satu langkah.

M_i adalah jumlah data dari *state* A_i .

Berdasarkan probabilitas transisi pada *state* tersebut, diperoleh matriks probabilitas transisi \mathbf{R} dengan dimensi $n \times n$, dituliskan sebagai berikut:



$$\mathbf{R} = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1n} \\ P_{21} & P_{22} & \dots & P_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1} & P_{n2} & \dots & P_{nn} \end{bmatrix}.$$

Langkah 7: Perhitungan hasil pemodelan awal.

Hasil pemodelan awal dihitung berdasarkan matriks probabilitas transisi \mathbf{R} , dengan aturan sebagai berikut:

1. Jika FLRG dari A_i bertransisi ke himpunan kosong ($A_i \rightarrow \emptyset$), maka hasil pemodelan F_t adalah m_i , yaitu nilai tengah dari u_i dengan persamaan:

$$F_t = m_i. \quad (2.12.78)$$

2. Jika FLRG dari A_i bertransisi satu ke satu ($A_i \rightarrow A_k$ dengan $P_{ij} = 0$ dan $P_{ik} = 1, j \neq k$), maka hasil pemodelan F_t adalah m_k yaitu nilai tengah dari u_k dengan persamaan:

$$F_t = m_k P_{ik} = m_k. \quad (2.12.79)$$

3. Jika FLRG dari A_j bertransisi satu ke banyak ($A_j \rightarrow A_1, A_2, \dots, A_n, j = 1, 2, \dots, n$), dan kumpulan data X_{t-1} pada saat $t - 1$ yang berada pada *state* A_j , maka hasil pemodelan F_t adalah sebagai berikut:

$$F_t = m_1 P_{j1} + m_2 P_{j2} + \dots + m_{j-1} P_{j(j-1)} + X_{t-1} P_{jj} + \quad (2.12.80)$$

$$m_{j+1} P_{j(j+1)} + \dots + m_n P_{jn},$$

dengan $m_1, m_2, \dots, m_{j-1}, m_{j+1}, \dots, m_n$ merupakan titik tengah dari $u_1, u_2, \dots, u_{j-1}, u_{j+1}, \dots, u_n$. Nilai m_j disubstitusikan oleh X_{t-1} agar diperoleh informasi dari *state* A_j saat $t - 1$.

Langkah 8: Perhitungan nilai penyesuaian pemodelan.

Perhitungan nilai penyesuaian pemodelan bertujuan untuk memperbaiki *error* pemodelan. Hal ini disebabkan oleh matriks probabilitas transisi yang bias. Oleh karena itu, nilai penyesuaian pemodelan (D_t) dihitung dengan mengikuti aturan-aturan berikut:

1. Jika *state* A_i pada saat $t - 1$ sebagai $F_{t-1} = A_i$, terjadi perpindahan lompatan transisi maju ke *state* A_{i+s} pada saat t , dengan $1 \leq s \leq n - i$, maka nilai penyesuaian D_t yaitu:

$$D_t = \left(\frac{l}{2}\right)s, \quad (1 \leq s \leq n - i), \quad (2.12.81)$$

dengan s adalah banyak lompatan perpindahan transisi maju.

2. Jika state A_i pada saat $t - 1$ sebagai $F_{t-1} = A_i$, terjadi perpindahan lompatan transisi mundur ke *state* A_{i-v} pada saat t , dengan $1 \leq v \leq i$, maka nilai penyesuaian D_t adalah

$$D_t = -\left(\frac{l}{2}\right)v, \quad (1 \leq v \leq i), \quad (2.12.82)$$

dengan v adalah banyak lompatan perpindahan transisi mundur.

Langkah 9: Penentuan hasil pemodelan akhir.

Hasil pemodelan akhir diperoleh dengan cara menjumlahkan hasil pemodelan awal dan nilai penyesuaian pemodelan. Bentuk umum hasil pemodelan akhir (FM_t) yaitu:



$$FM_t = F_t + D_t, \quad (2.12.83)$$

dimana

FM_t : hasil pemodelan akhir pada periode ke- t

F_t : hasil pemodelan awal pada periode ke- t

D_t : nilai penyesuaian pemodelan.

2.13 Tingkat Keakuratan Model

Perhitungan *error* merupakan suatu cara untuk mengetahui tingkat keakuratan model yang telah diperoleh dengan data observasinya. Penggunaan teknik pemodelan dengan tingkat *error* terkecil merupakan teknik pemodelan yang terbaik. Metode untuk menghitung besar *error* ini diantaranya adalah *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE), dan *Mean Absolute Percentage Error* (MAPE) [17].

1. *Mean Absolute Error* (MAE).

Simpangan rata-rata MAE mengukur akurasi pemodelan dengan meratakan nilai absolut *error* pemodelan. Perhitungannya adalah sebagai berikut:

$$MAE = \frac{1}{n} \sum_{t=1}^n |X_t - Z_t|, \quad (2.13.84)$$

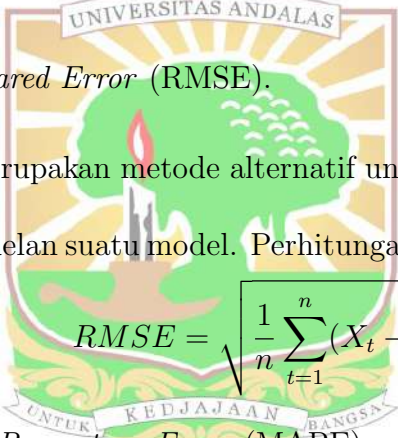
dengan X_t : data observasi pada waktu t

Z_t : data hasil pemodelan pada waktu t

n : banyak data observasi.

2. *Root Mean Squared Error* (RMSE).

RMSE merupakan metode alternatif untuk mengukur tingkat akurasi hasil pemodelan suatu model. Perhitungannya adalah sebagai berikut:


$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (X_t - Z_t)^2}. \quad (2.13.85)$$

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

MAPE merupakan rata-rata dari keseluruhan persentase kesalahan (selisih) antara data observasi dengan data hasil pemodelan [32]. MAPE dapat diperoleh dengan rumus sebagai berikut:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|X_t - Z_t|}{X_t} \times 100\%. \quad (2.13.86)$$

Kriteria keakuratan MAPE adalah sebagai berikut:

- (a) Ketepatan pemodelan sangat baik saat nilai MAPE < 10%.
- (b) Ketepatan pemodelan baik saat nilai MAPE 10% – 20%.

(c) Ketepatan pemodelan cukup saat nilai MAPE 20% – 50%.

(d) Ketepatan pemodelan tidak akurat saat nilai MAPE > 50%.

Semakin kecil nilai yang dihasilkan oleh ketiga metode tersebut, maka model yang digunakan akan semakin baik.

