Modul Praktikum Analisis Deret Waktu - AK2281 terdiri atas 5 buah modul sebagai berikut :

1. Modul 1 : Konsep Dasar Deret Waktu
2. Modul 2 : Model Data Tak Stasioner
3. Modul 3 : Model Musiman
4. Modul 4 : Model Heteroskedastik
5. Modul 5 : Pengayaan

Tim Praktikum AK2281

Koordinator Praktikum : Leonardo V. Kosasih, S.Aktr.

Tim Penyusun Modul Praktikum :

Ang Ditra Alif Pradana	10120046	Matthew Alfarazh	10820021
Feby Yolanda	10819028	Pamella Cathryn	10820033
Ferdinan Gratius Budisatya	10819041	Jeremy	10820034
Jevan Christopher Aryento	10820010	Aloysius Vincent	10820038
Shelly Delfiani	10820014	Kevin Christ Aditya	10820039
Binsar Gunadi Simbolon	10820017	Shafina Aulia Kusuma Putri	10820049

Desain sampul oleh : Matthew Alfarazh - 10820021

Daftar Isi

Modul 2

Tujuan:

1. Menganalisis data real deret waktu dan memodelkannya.
2. Memahami langkah-langkah iterasi Box-Jenkins,

Pada modul kali ini ada beberapa *library* yang akan digunakan sebagai berikut :

```
# Untuk membaca data dengan format Excel
library(readxl)

# Untuk mengubah data menjadi Time Series,
# membuat plot ACF, plot PACF, Model ARIMA dan ADF Test
library(tseries)

# Untuk membuat rolling means
library(zoo)

# Untuk memgolah data
library(dplyr)
library(tidyverse)

# Untuk melihat signifikansi koefisien dari parameter
library(lmtest)

# Untuk memprediksi data dari model
library(forecast)

# Untuk membuat grafik
library(ggplot2)
```

Sedangkan data yang akan diolah adalah data Total Barang Dalam Negri yang dimuat di Tanjung Priok. Data diperoleh dari bps.go.id pada bulan Juni tanggal 13 tahun 2020.

TEKAN UNTUK MENGUNDUH DATA

Alur Pemodelan

Data deret waktu yang baik adalah data yang memiliki ukuran setidaknya 50 observasi. Jika kurang dari 50 observasi maka hasil pengolahan data deret waktu dapat menjadi sangat bias. Selain jumlah data, perlu diperhatikan kualitas dari data juga perlu diperhatikan. Secara umum, alur pemodelan deret waktu dapat dibagi menjadi berikut :

1. Analisis Data atau EDA (*Exploratory Data Analysis*)
   Pada bagian ini akan dihitung statistik deskriptif dari data (nilai minimum, maksimum, rataan, median dan lainnya) dan dapat dilihat pola dari data. Kelengkapan data juga harus diperiksa. Umumnya, data deret waktu dilaporkan secara runtut (per bulan, per hari dan seterusnya).

2. Mempersiapkan data atau *Data Processing*
   Pada bagian ini dapat dilakukan *imputation* jika terdapat data yang kosong dan dapat dilakukan transformasi ataupun diferensiasi data. Dalam mempersiapkan data sebelum membangun model, ada baiknya pula untuk membagi data menjadi 2 bagian dengan rasio 80:20 dengan 80% data pertama akan disebut dengan data *training* dan 20% berikutnya akan disebut dengan data *validation/test*. Tujuan dari membagi data ini adalah untuk membagi data yang akan digunakan untuk **membangun** model dan untuk **menguji** model untuk menghindari *overfitting* dan juga untuk melihat performa model dalam melakukan prakiraan/*forecasting* pada data yang digunakan untuk **membangun** model dan dari data yang "belum diketahui". Hasil pengujian model dengan menggunakan data *training* disebut dengan *in-sample error* sedangkan jika menggunakan data *validation/test* disebut dengan *out-of-sample error*

3. Identifikasi Model
   Khusus untuk data deret waktu, akan ditinjau perilaku dari autokorelasi dan autokorelasi parsial dari data. Pada bagian ini akan dipilih beberapa kandidat model yang dirasa cocok untuk memodelkan data.

4. Estimasi Parameter
   Pada bagian ini akan dianalisis parameter yang telah ditaksir. Idealnya, parameter yang dipilih signifikan terhadap model dan juga memiliki jumlah parameter yang sedikit pula. Misalkan $\hat{\beta}$ adalah taksiran dari suatu parameter dan β adalah nilai parameter yang sebenarnya. Taksiran yang baik adalah taksiran yang memenuhi sifat-sifat berikut:

   (a) Tak bias: ekspektasi dari taksiran parameter adalah nilai parameter sebenarnya yang secara matematis ditulis sebagai berikut
   $$E\left[\hat{\beta}\right] = \beta \tag{0.1}$$

   (b) Konsisten: untuk data yang semakin banyak, maka peluang dari suatu taksiran parameter dengan nilai dari parameter sebenarnya akan mendekati 0 (sangat mirip). Secara matematis ditulis sebagai berikut
   $$\lim_{n\to\infty} \Pr\left(|\hat{\beta}_n - \beta| \leq \delta\right) = 1 \quad \forall \delta > 0 \tag{0.2}$$

   (c) Efisien: ukuran efisiensi yang umum digunakan adalah *Mean Squared Error* yang secara matematis ditulis sebagai berikut:
   $$E\left[\left(\hat{\beta} - \beta\right)^2\right] \tag{0.3}$$

5. Uji Diagnostik
   Pada bagian ini akan diperiksa perilaku galat yang dihasilkan oleh model. Ada 3 asumsi yang

harus dipenuhi oleh galat yaitu:

(a) Berdistribusi Normal dengan rataan 0 dan variansi σ_ε^2 atau $\varepsilon_t \sim N(0, \sigma_\varepsilon^2) \quad \forall t$

(b) Saling bebas

(c) Homoskedastis: variansi konstan

6. Penaksiran atau *Forecasting*
   Pada bagian ini akan dilakukan penaksiran titik dan taksiran selang dengan tingkat signifikansi tertentu.

Sedangkan pemodelan iterasi Box-Jenkins adalah

1. Identifikasi Model
2. Pembuatan Model (*Model Fitting*)
3. Uji Diagnostik Model

Dikatakan itereasi karena jika galat dari model tidak memenuhi sifat-sifat galat yang telah disampaikan di atas, maka perlu dilakukan kembali identifikasi model.
Catatan: penulisan angka ribuan akan digunakan tanda koma (,) sedangkan untuk menuliskan angka desimal akan digunakan tanda titik(.).

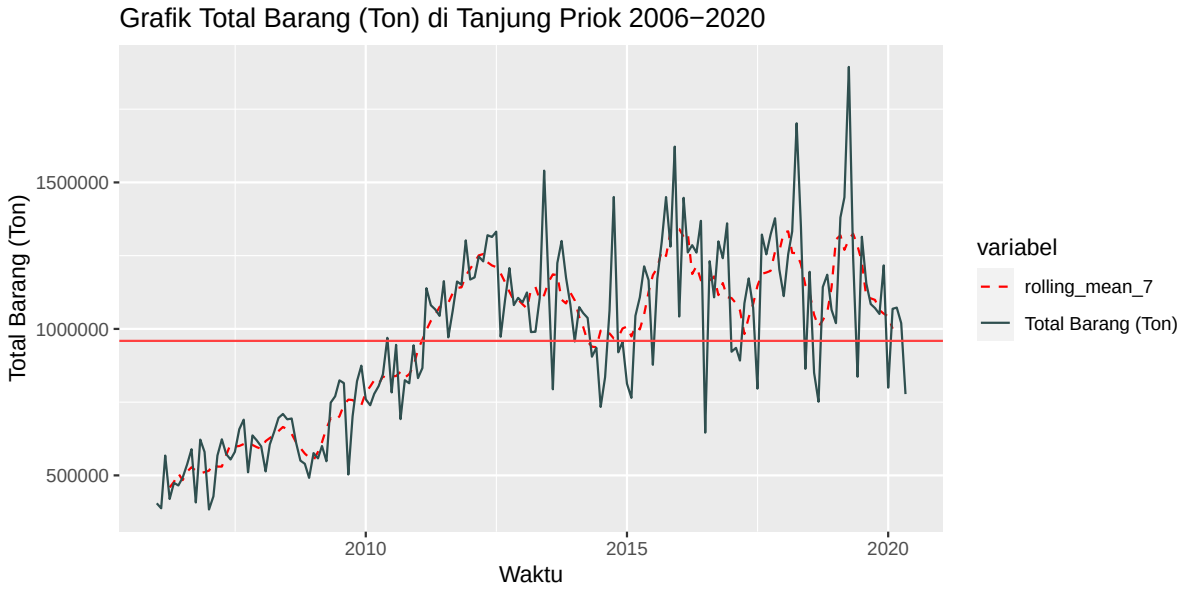
1 EDA

Pada bagian ini akan dilihat grafik data, pola yang dibangun dari *rolling mean*, dan juga stastik deskriptif dari data

```
# Memanggil Data
data <- read_excel("Data Total Barang.xlsx",
                   sheet = "Tanjung Priok",
                   col_types = c("date", "numeric"))
data <- data %>% mutate(
  rolling_mean_7 = rollmean(data$`Total Barang (Ton)`, k = 7, fill = NA))

# Mengelompokan data
df <- data %>%
  select(Waktu, `Total Barang (Ton)`,rolling_mean_7) %>%
  gather(key = "variabel", value = "value", -Waktu)

# Membuat grafik data
ggplot(df, aes(x = Waktu, y = value, group = variabel)) +
  geom_line(aes(linetype = variabel, color = variabel)) +
  geom_hline(yintercept = mean(data$`Total Barang (Ton)`), color="brown1")+
  scale_linetype_manual(values = c('dashed', 'solid', 'dashed'))+
  scale_color_manual(values = c('red','darkslategrey')) +
  labs(title = 'Grafik Total Barang (Ton) di Tanjung Priok 2006-2020',
       x = "Waktu", y = 'Total Barang (Ton)')
```




```
# Statistik Deskriptif
summary(data)
```

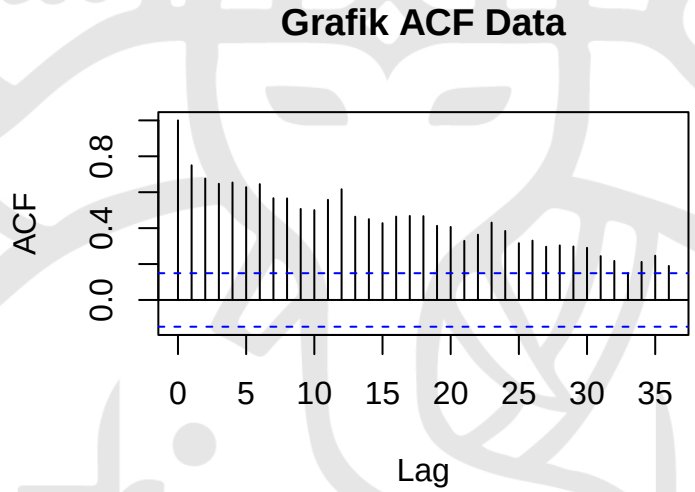
```
##      Waktu                         Total Barang (Ton) rolling_mean_7
##  Min.   :2006-01-01 00:00:00.00   Min.   : 383471    Min.   : 458872
##  1st Qu.:2009-08-01 00:00:00.00   1st Qu.: 709685    1st Qu.: 742811
##  Median :2013-03-01 00:00:00.00   Median : 989212    Median :1048126
```

```
##  Mean   :2013-03-01 18:35:22.53   Mean   : 959191    Mean   : 968516
##  3rd Qu.:2016-10-01 00:00:00.00   3rd Qu.:1174929    3rd Qu.:1165320
##  Max.   :2020-05-01 00:00:00.00   Max.   :1894263    Max.   :1343004
##                                                      NA's   :6
```

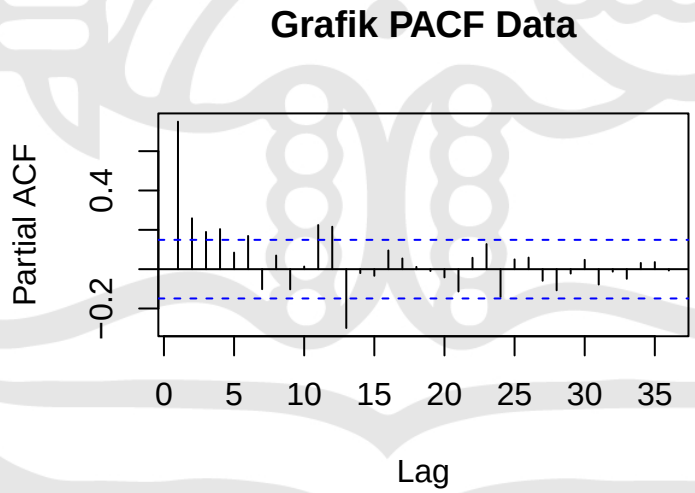
Dapat dilihat bahwa terdapat *trend* naik dari tahun 2006 hingga tahun 2013 kemudian data mulai berfluktuasi di sekitar rataan. Dapat dilihat juga bahwa rata-rata bergulir per 7 hari mengikuti pola dari data.

Akan dilihat pula grafik ACF dan PACF dari data

```
acf(data$`Total Barang (Ton)`, main ='Grafik ACF Data',lag.max = 36)
```




```
pacf(data$`Total Barang (Ton)`, main ='Grafik PACF Data',lag.max = 36)
```



Dapat dilihat pada grafik ACF nilai autokorelasi signifikan hingga lag ke-36 dan grafik PACF signifikan hingga lag ke-3. Nilai autokorelasi yang signifikan hingga suatu lag yang besar, menjadi salah satu indikasi bahwa data tidak stasioner.

2 Mempersiapkan Data

Berikutnya data akan dibagi menjadi data *train* dan data *validation*.

```
# Membuat data train dan validation dan mengubahnya
# menjadi data time series

# dibutuhkan untuk menghasilkan data prediksi
tpriok <- ts(data$`Total Barang (Ton)`)

# data yang akan digunakan untuk membuat model
tpriok_train <- ts(data$`Total Barang (Ton)`[1:138])

# data yang akan digunakan untuk memvalidasi model
tpriok_test <-ts(data$`Total Barang (Ton)`[139:173])
```

Untuk langkah-langkah berikutnya, data yang akan diolah adalah data *train*.

Sebelum memilih model yang akan digunakan untuk melakukan prakira, ada baiknya juga dipilih suatu model yang paling sederhana yang disebut dengan *base model*. Untuk memudahkan, akan dibuat model yang merupakan rataan dari data sebagai *base model* dan akan dilihat performa dari *base model* tersebut.

```
y_mean <- mean(tpriok_train)
rmse_train <- mean((tpriok_train - y_mean)^2)^0.5
rmse_train
```

```
## [1] 291227.7
```

```
y_mean <- mean(tpriok_train)
mape_train <- mean(abs((tpriok_train - y_mean))/tpriok_train)
mape_train
```

```
## [1] 0.3297599
```

Dapat dilihat bahwa RMSE (*Root Mean Square Error*) dari *in-sample* sebesar 291,227.7 dan MAPE (*Mean Absoulte Percentage Error*) sebesar 32.97 %, sehingga model yang dipilih nanti harus memiliki nilai yang lebih kecil dari nilai tersebut.

3 Identifikasi Model

Dalam mengindentifikasi model, langkah pertama yang perlu dilakukan adalah memeriksa kestasioneran data. Salah satu caranya adalah dengan melakukan uji **Augmented Dickey Fuller Test**, yang berikutnya akan disebut dengan uji ADF. Uji ADF ini akan menguji apakah **unit root** pada model deret waktu bernilai kecil dari 1 atau tidak. **Unit root** inilah yang menjadi salah satu indikator apakah suatu data deret waktu stasioner atau tidak. Jika H_0 ditolak maka dapat ditarik kesimpulan bahwa data stasioner. Selain uji ADF, dapat pula dilihat pola dari grafik ACF data. Pada modul ini akan digunakan $\alpha = 0.05$. Selengkapnya mengenai uji ADF, dapat dilihat pada Lampiran B.

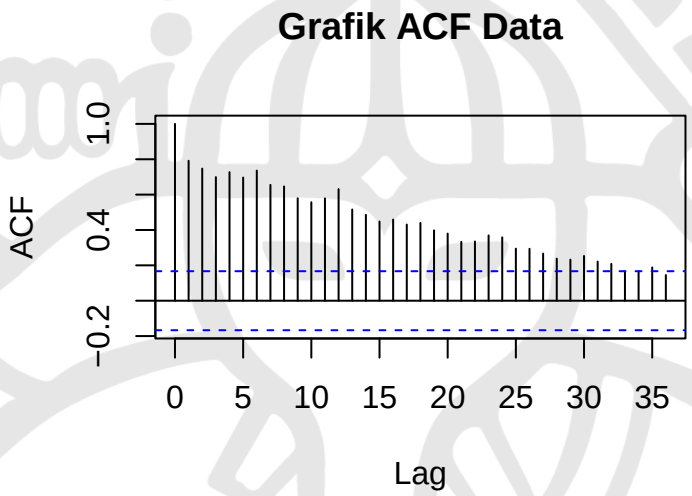
Akan dilakukan Uji ADF pada data

```
adf.test(tpriok_train)
```

```
##
```

```
##  Augmented Dickey-Fuller Test
##
## data:  tpriok_train
## Dickey-Fuller = -1.4635, Lag order = 5, p-value = 0.7997
## alternative hypothesis: stationary
```

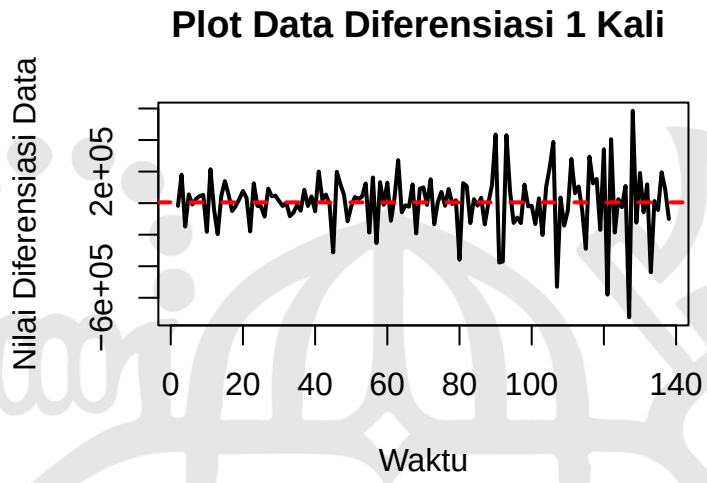
```
acf(tpriok_train, main = 'Grafik ACF Data',lag.max = 36)
```



Dapat dilihat bahwa nilai $p\text{-}value > \alpha$ sehingga dapat disimpulkan bahwa **data tidak stasioner**. Hal ini didukung pula dengan nilai ACF yang masih signifikan hingga lag ke-36.

Karena data tidak stasioner, maka salah satu cara untuk membuat data menjadi stasioner adalah dengan melakukan diferensiasi data.

```
tpriok_train_diff <- diff(tpriok_train)
plot(tpriok_train_diff,
     lwd = 2,
     main = 'Plot Data Diferensiasi 1 Kali ',
     xlab = "Waktu", ylab = "Nilai Diferensiasi Data")
abline(h=mean(tpriok_train_diff),
       lwd=2,lty = 2,
       col ='red')
```



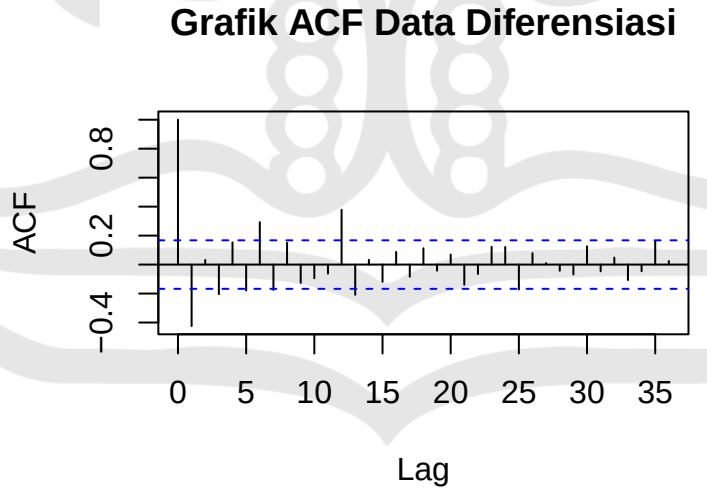
Dapat dilihat bahwa data yang sudah didiferensiasi jauh lebih terlihat stasioner dibandingkan dengan yang belum didiferensiasi. Untuk dapat mendukung argumen bahwa data sudah stasioner, akan dilakukan uji ADF dan akan dilihat pula grafik ACF-nya

```
adf.test(tpriok_train_diff)
```

```
## Warning in adf.test(tpriok_train_diff): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  tpriok_train_diff
## Dickey-Fuller = -6.8938, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```
acf(tpriok_train_diff, main ='Grafik ACF Data Diferensiasi',
    lag.max = 36)
```



Dapat dilihat bahwa *p-value* $< \alpha$ sehingga dapat disimpulkan bahwa model sudah stasioner. Terlihat

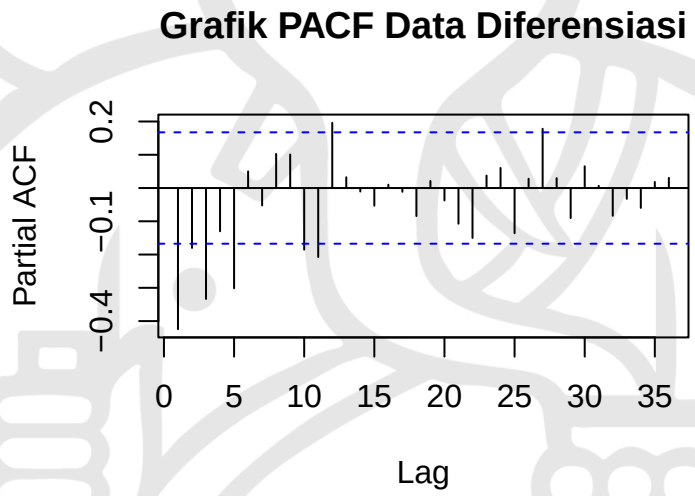
pula bahwa plot ACF memiliki pola sinusoidal dan *cut-off* pada lag pertama.

Jurnal Praktikum 1

Bandingkan hasil di atas dengan melakukan transformasi `logaritma natural` *atau dengan melakukan diferensiasi lagi. Menurut anda, dari 3 metode tersebut (logaritma natural, diferensiasi 2 kali dan diferensiasi 1 kali) metode mana yang lebih cocok untuk dilakukan. Apakah dari 3 metode tersebut menghasilkan data yang stasioner ? Adakah metode yang lebih baik diantara 3 metode tersebut ?*

Setelah data menjadi stasioner, berikutnya akan diidentifikasi model yang cocok pada data dengan melihat grafik ACF dan PACF data.

```
pacf(tpriok_train_diff, main ='Grafik PACF Data Diferensiasi',
     lag.max = 36)
```



Pada plot PACF dapat dilihat bahwa grafik *cut off* hingga lag ke-3.

Dari grafik ACF dan PACF di atas, dapat disimpulkan model-model yang cocok adalah sebagai berikut:

1. ARIMA (3,1,1)
2. ARIMA (3,1,0)
3. ARIMA (2,1,1)

Untuk interpretasi model dapat diperoleh hasil yang berbeda dengan model di atas asalkan alasan/argumen yang diberikan masih bisa diterima. Model yang baik adalah model yang memiliki sifat **parsimoni**, yang berarti jika ada 2 buah model yang memiliki performa yang hampir mirip, maka model yang baik adalah model yang paling sederhana atau model dengan parameter yang paling sedikit. Untuk itu perlu dilakukan analisis lebih lanjut seperti melihat signifikansi dari parameter model, nilai AIC (*Akaike's Information Criteria*) dari model dan uji diagnostik model pada ketiga kandidat model tersebut.

4 Estimasi Parameter

Dalam melakukan estimasi parameter dapat dilakukan dengan menggunakan autokorelasi parsial ataupun dengan menggunakan metode galat kuadrat terkecil. Pada bagian ini akan dibahas mengenai metode galat kuadrat terkecil untuk model AR(1), dan MA(1).

4.1 AR

Misalkan data Y_t dengan $t \in \{1, 2, \dots, n\}$mengikuti model AR(1). Maka persamaannya dapat ditulis sebagai berikut:

$$Y_t - \mu = \phi(Y_{t-1} - \mu) + \varepsilon_t, \quad t \in \{2, \dots n\} \tag{4.1}$$

Esimasi dengan kuadrat terkecil diperoleh dengan meminimumkan jumlah kuadrat galat. Perhatikan bahwa persamaan (4.1) dapat ditulis ulang menjadi

$$\varepsilon_t = Y_t - \mu - \phi(Y_{t-1} - \mu), \quad t\{2, \dots n\} \tag{4.2}$$

dengan μ adalah rataan dari data.
Misalkan total jumlah kuadrat galat adalah $S_c(\phi, \mu)$. Maka secara matematis dapat ditulis menjadi :

$$S_c(\phi, \mu) = \sum_{t=2}^{n} \varepsilon_t^2 = \sum_{t=2}^{n} ((Y_t - \mu) - \phi(Y_{t-1} - \mu))^2 \tag{4.3}$$

Maka untuk memperoleh nilai parameter ϕ dan μ yang meminimumkan jumlah kuadrat galat ini (jumlah kuadrat ini disebut juga dengan **conditional sum of square error**, akan dihitung turunan parsial dari persamaan (4.3) :

$$\begin{aligned}
\frac{\partial S_c(\phi, \mu)}{\partial \phi} = \frac{\partial \sum_{t=2}^{n} \varepsilon_t^2}{\partial \phi} &= \frac{\partial \sum_{t=2}^{n} ((Y_t - \mu) - \phi(Y_{t-1} - \mu))^2}{\partial \phi} \\
&= -2 \sum_{t=2}^{n} (Y_t - \mu)(Y_{t-1} - \mu) + 2 \sum_{t=2}^{n} \phi (Y_{t-1} - \mu)^2 = 0 \\
&\Leftrightarrow \hat{\phi} = \frac{\sum_{t=2}^{n} (Y_t - \bar{Y})(Y_{t-1} - \bar{Y})}{\sum_{t=2}^{n} (Y_{t-1} - \bar{Y})^2}
\end{aligned}$$

Sedangkan untuk nilai parameter μ yang meminimumkan galat adalah :

$$\begin{aligned}
\frac{\partial S_c(\phi, \mu)}{\partial \mu} = \frac{\partial \sum_{t=2}^{n} \varepsilon_t^2}{\partial \mu} &= \frac{\partial \sum_{t=2}^{n} ((Y_t - \mu) - \phi(Y_{t-1} - \mu))^2}{\partial \mu} \\
&= (\text{Diserahkan kepada pembaca sebagai latihan}) \\
&\Leftrightarrow \hat{\mu} \approx \frac{1}{1 - \hat{\phi}} (\bar{Y} - \hat{\phi} \bar{Y}), \quad \text{untuk } n \text{ yang besar}
\end{aligned}$$

4.2 MA

Misalkan data Y_t dengan $t = \{1, 2, \dots, n\}$mengikuti model MA(1). Maka persamaannya dapat ditulis sebagai berikut :

$$Y_t = \varepsilon_t - \theta\varepsilon_{t-1}, \quad t = 2, \dots n \tag{4.4}$$

Sehingga untuk **conditional sum of square error**-nya dapat ditulis dengan persamaan berikut :

$$S_c(\theta) = \sum_{t=1}^{n} \varepsilon_t = \sum_{t=1}^{n} (Y_t + \theta Y_{t-1} + \theta^2 Y_{t-2} + \dots)^2 \tag{4.5}$$

Perhatikan bahwa persamaan (4.5) memiliki parameter yang tak linear. Sehingga dalam penentuannya perlu dilakukan dengan metode numerik.

Salah satu alternatif lain adalah dengan menyusun ulang persaman (4.4) menjadi berikut :

$$\varepsilon_t = Y_t + \theta\varepsilon_{t-1}, \quad t \in \{1 \dots n\} \tag{4.6}$$

Idenya adalah dengan menaksir nilai θ yang dapat meminimumkan galat. Umumnya diasumsikan nilai dari $\varepsilon_0 = 0$ (Mengapa?) sehingga dapat dihitung secara rekursif nilai dari parameter θ.

Penjelasan umum mengenai metode *Maximum Likelihood* dapat dilihat pada Lampiran D

4.3 Bagian R

Berikut adalah kode untuk melakukan penaksiran parameter model

```
# Metode default dari arima adalah kombinasi antara maximum likelihood
# dan conditional sum of square

mod_1 <- arima(tpriok_train,order=c(3,1,1))
mod_1
```

```
##
## Call:
## arima(x = tpriok_train, order = c(3, 1, 1))
##
## Coefficients:
##          ar1      ar2      ar3      ma1
##      -0.0766  -0.1057  -0.2465  -0.5646
## s.e.  0.1446   0.1090   0.0966   0.1347
##
## sigma^2 estimated as 2.256e+10:  log likelihood = -1827.79,  aic = 3665.59
```

```
mod_2 <- arima(tpriok_train,order=c(3,1,0))
mod_2
```

```
##
## Call:
## arima(x = tpriok_train, order = c(3, 1, 0))
##
## Coefficients:
##          ar1      ar2      ar3
```

```
##       -0.5537  -0.3411  -0.3287
## s.e.   0.0804   0.0888   0.0805
##
## sigma^2 estimated as 2.372e+10:  log likelihood = -1831.15,  aic = 3670.29
```

```
mod_3 <- arima(tpriok_train,order=c(2,1,1))
mod_3
```

```
##
## Call:
## arima(x = tpriok_train, order = c(2, 1, 1))
##
## Coefficients:
##          ar1      ar2      ma1
##       0.1052  -0.0018  -0.7510
## s.e.  0.1207   0.1043   0.0863
##
## sigma^2 estimated as 2.351e+10:  log likelihood = -1830.55,  aic = 3669.11
```

```
mod_auto <- auto.arima(tpriok_train, max.p = 4,max.q = 4,
                       seasonal = FALSE, stationary = FALSE)
mod_auto
```

```
## Series: tpriok_train
## ARIMA(3,1,2)
##
## Coefficients:
##          ar1     ar2      ar3     ma1      ma2
##       -0.7712  0.0020  -0.1868  0.1925  -0.6072
## s.e.   0.1387  0.1714   0.1067  0.1251   0.1173
##
## sigma^2 = 2.258e+10:  log likelihood = -1825.47
## AIC=3662.94   AICc=3663.59   BIC=3680.46
```

Setelah melakukan penaksiran parameter, akan dipilih model yang paling cocok untuk memodelkan data. Ada 3 hal yang akan dipertimbangkan dalam memilih model yang paling cocok sebagai berikut:

1. Nilai AIC (*Akaike's Information Criteria*)
2. Signifikansi parameter
3. Parsimoni

Tentu saja selain 3 hal di atas, perbedaan antara performa model dengan *base model* dapat menjadi pertimbangan.
Nilai AIC didefinisikan dengan persamaan berikut:

$$AIC = -2\log(\textit{maximum likelihood}) + 2k \tag{4.7}$$

Dengan *maximum likelihood* diperoleh dari fungsi kepadatan peluang galat (umumnya dipilih distribusi normal dengan rataan 0 dan variansi σ_ε^2) dan k adalah banyak parameter. Secara umum fungsi ini akan memberikan *penalty* yang lebih besar pada model yang memiliki banyak parameter. Model yang baik

adalah model yang memiliki nilai AIC **terendah**.
Catatan: Alasan mengenai pemilihan model yang terbaik menggunakan AIC terendah karena pendefinisian persamaan AIC pada persamaan (4.7). Jika pendefinisian nilai AIC tidak sama dengan persamaan (4.7) maka model terbaik dapat berupa nilai AIC terbesar.

Dapat dilihat bahwa model dengan AIC terkecil dimiliki oleh model ARIMA (3,1,2). Namun, dapat dilihat pula pada model ARIMA(2,1,1) memiliki AIC yang tidak jauh berbeda dengan AIC pada model ARIMA(3,1,2). sehingga akan dipilih model ARIMA(2,1,1) untuk dianalisis lebih lanjut.

Jurnal Praktikum 2

Bagaimana dengan nilai AICc dan BIC dari 3 model tersebut? Apakah model dengan AIC terkecil juga memiliki nilai AICc dan nilai BIC terkecil juga? Jelaskan mengapa diperoleh hasil tersebut!

Berikutnya akan dibandingkan parameter model yang diperoleh dengan metode penaksiran parameter kuadrat galat terkecil

```
mod_1 <- arima(tpriok_train,order=c(2,1,1), method = 'CSS-ML')
mod_1
```

```
##
## Call:
## arima(x = tpriok_train, order = c(2, 1, 1), method = "CSS-ML")
##
## Coefficients:
##          ar1      ar2      ma1
##       0.1052  -0.0018  -0.7510
## s.e.  0.1207   0.1043   0.0863
##
## sigma^2 estimated as 2.351e+10:  log likelihood = -1830.55,  aic = 3669.11
```

```
mod_1_css <- arima(tpriok_train,order=c(2,1,1), method = 'CSS')
mod_1_css
```

```
##
## Call:
## arima(x = tpriok_train, order = c(2, 1, 1), method = "CSS")
##
## Coefficients:
##          ar1      ar2      ma1
##       0.0877  -0.0123  -0.7445
## s.e.  0.1175   0.1024   0.0857
##
## sigma^2 estimated as 2.382e+10:  part log likelihood = -1831.13
```

```
mod_1_ML <- arima(tpriok_train,order=c(2,1,1), method = 'ML')
mod_1_ML
```

```
##
## Call:
## arima(x = tpriok_train, order = c(2, 1, 1), method = "ML")
```

```
##
## Coefficients:
##          ar1      ar2      ma1
##       0.1051  -0.0018  -0.7509
## s.e.  0.1207   0.1043   0.0863
##
## sigma^2 estimated as 2.351e+10:  log likelihood = -1830.55,  aic = 3669.11
```

Dapat dilihat bahwa antara ketiga model memiliki parameter yang tidak jauh berbeda tetapi dapat dilihat bahwa σ^2 dari metode **conditional sum of square error** lebih besar dibandingkan metode yang lain. Sedangkan AIC model menggunakan penaksiran **maximum likelihood** ataupun kombinasi antara **maximum likelihood** dan **conditional sum of square error** tidak jauh berbeda. Sehingga untuk berikutnya akan dilakukan uji signifikansi dari model.

4.4 Signifikansi dari Koefisien Parameter

Akan diperiksa siginifikansi parameter model

```
coeftest(mod_1)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error z value Pr(>|z|)
## ar1  0.1051604  0.1207309  0.8710   0.3837
## ar2 -0.0017724  0.1043285 -0.0170   0.9864
## ma1 -0.7509917  0.0862586 -8.7063   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(mod_1_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.087714   0.117502  0.7465   0.4554
## ar2 -0.012293   0.102450 -0.1200   0.9045
## ma1 -0.744464   0.085673 -8.6896   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(mod_1_ML)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error z value Pr(>|z|)
## ar1  0.1051306  0.1207419  0.8707   0.3839
## ar2 -0.0018013  0.1043297 -0.0173   0.9862
## ma1 -0.7509279  0.0862867 -8.7027   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Dapat dilihat bahwa masing-masing model ada parameter yang **tidak signifikan**. Sehingga dapat disimpulkan bahwa model yang dipilih kurang tepat untuk memodelkan data. Idealnya semua parameter signifikan pada model. Untuk dapat memperoleh model yang signifikan dapat dilakukan salah satu ataupun kombinasi dari hal-hal berikut :

1. Mengubah rasio data untuk *training* dan *testing*
2. Melakukan diferensiasi data
3. Memilih model lain

Penulisan persamaan dari model adalah sebagai berikut :

$$\begin{aligned}
\phi_2(B)(1-B)^1 Y_t &= \theta_1(B)\varepsilon_t \\
(1-\phi_1 B-\phi_2 B^2)(1-B)Y_t &= (1-\theta_1 B)\varepsilon_t \\
(1-\phi_1 B-\phi_2 B^2-B+\phi_1 B^2+\phi_2 B^3)Y_t &= (1-\theta_1 B)\varepsilon_t \\
(1-(1+\phi_1)B-(\phi_2-\phi_1)B^2+\phi_2 B^3)Y_t &= \varepsilon_t-\theta_1\varepsilon_{t-1} \\
Y_t-(1+\phi_1)Y_{t-1}-(\phi_2-\phi_1)Y_{t-2}+\phi_2 Y_{t-3} &= \varepsilon_t-\theta_1\varepsilon_{t-1}
\end{aligned}$$

$$Y_t = (1+\phi_1)Y_{t-1}-(\phi_2-\phi_1)Y_{t-2}+\phi_2 Y_{t-3}+\varepsilon_t-\theta_1\varepsilon_{t-1} \tag{4.8}$$

(Perhatikan dokumentasi dari fungsi `arima`).
Perhatikan bahwa koefisien parameter dari ϕ_1, dan ϕ_2 tidak signifikan sehingga dalam penulisan persamaan modelnya hanya menuliskan koefisien parameter dari ϕ_2 dan θ_1. Sehingga persamaan (4.9) dapat ditulis sebagai berikut :

$$Y_t = Y_{t-1}+\varepsilon_t-0.7509917\varepsilon_{t-1} \tag{4.9}$$

Berikutnya akan dilihat performa model sebagai berikut

```
accuracy(mod_1)
```

```
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 17256.47 152764.4 112023.8 0.1114048 12.78606 0.8855094
##                       ACF1
## Training set -0.01749564
```

Definisi dari ukuran dari tiap performa model dapat dilihat pada Lampiran C. Dapat dilihat bahwa RMSE dari model sebesar 152,764 dan MAPE dari model sebesar 12.78% yang nilai tersebut jauh lebih kecil dibandingkan nilai RMSE dan MAPE dari *base model*.

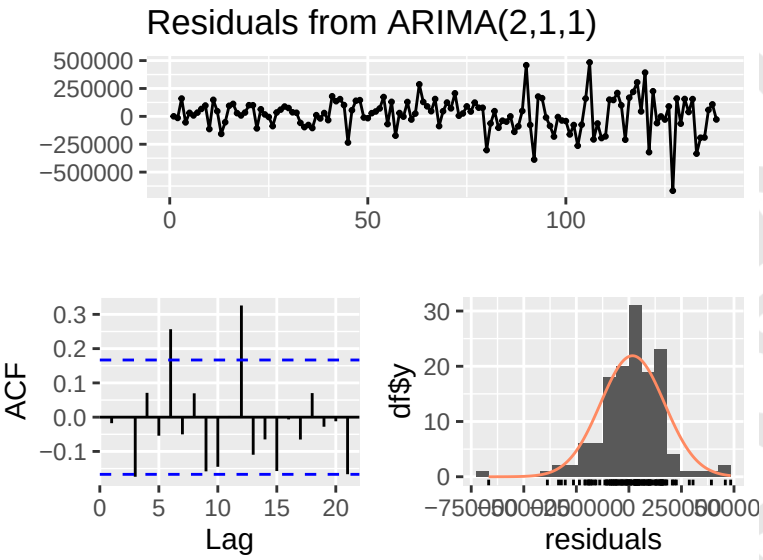
5 Uji Diagnostik

Model deret waktu dikatakan cocok jika galat memenuhi sifat-sifat berikut:

1. Berdistribusi Normal dengan rataan 0 dan variansi σ_ε^2 atau $\varepsilon_t \sim N(0, \sigma_\varepsilon^2) \quad \forall t$
2. Saling bebas
3. Homoskedastis: variansi konstan

Untuk melakukan uji diagnostik dapat dilakukan dengan bantuan grafik dan juga uji statistik seperti Ljung-Box seperti berikut :

```
checkresiduals(mod_1)
```



Gambar 1: Model Diagnostik dari Data Prediksi

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,1)
## Q* = 23.167, df = 7, p-value = 0.001594
##
## Model df: 3.   Total lags used: 10
```

Perhatikan bahwa *p-value* $< \alpha$ sehingga dapat disimpulkan bahwa data tidak saling bebas. Hal ini diperkuat dari plot ACF yang signifikan pada lag ke 6,19 dan juga 12. Dapat dilihat juga bahwa distribusi dari residual hampir menyerupai distribusi normal dan grafik dari galat dapat diasumsikan bahwa galat dari residual 0 dan variansinya konstan.

Jurnal Praktikum 3

Lakukanlah uji Kolmogorov-Smirnov, Anderson-Darling, Cramer von Mises, Jarque-Bera, dan Shapiro-Wilk pada data residual dengan menggunakan fungsi `residuals(mod_1)`*. Apakah dari ke-5 uji tersebut ada yang menolak bahwa data residual berdistribusi normal?*

6 Forecasting

Sebelum dilakukan **forecasting** akan dilihat terlebih dahulu performa model untuk memodelkan data **validation** terlebih dahulu.

```
validation <- forecast(tpriok_train, model = mod_1, h = length(tpriok_test))
actual <- as.vector(tpriok_test)
ape_validation <- abs((as.vector(validation$mean) -actual)/actual)
```

```
mape_validation <- mean(ape_validation)
mape_validation
```

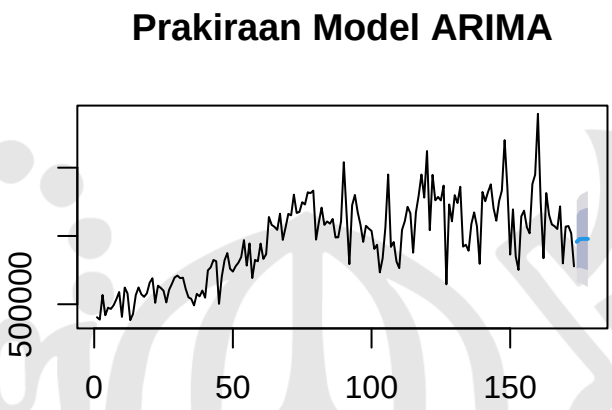
```
## [1] 0.1687372
```

Dapat dilihat bahwa MAPE dari model sebesar 16.87% yang lebih besar daripada MAPE yang diperoleh sebelumnya. Untuk memprediksi data di depan dapat digunakan fungsi dari library forecast yakni `forecast`. Perlu diperhatikan data apa yang akan diprediksi dan model yang akan digunakan.

```
fc <- forecast(tpriok, model = mod_1, h = 5)
summary(fc)
```

```
##
## Forecast method: ARIMA(2,1,1)
##
## Model Information:
## Series: object
## ARIMA(2,1,1)
##
## Coefficients:
##          ar1      ar2     ma1
##       0.1052  -0.0018  -0.751
## s.e.  0.0000   0.0000   0.000
##
## sigma^2 = 2.351e+10:  log likelihood = -2323.54
## AIC=4649.07   AICc=4649.1   BIC=4652.22
##
## Error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
## Training set 11574.6 177216.5 129109.5 -0.983319 13.87568 0.8805997 0.03903638
##
## Forecasts:
##     Point Forecast    Lo 80   Hi 80    Lo 95   Hi 95
## 174       957377.3 760888.7 1153866 656873.9 1257881
## 175       976721.6 768273.6 1185170 657927.9 1295515
## 176       978437.0 762624.5 1194250 648380.2 1308494
## 177       978583.1 755950.7 1201216 638096.2 1319070
## 178       978595.5 749370.6 1207820 628026.3 1329165
```

```
plot(fc, , main = 'Prakiraan Model ARIMA')
```



Dapat dilihat bahwa hasil prediksi model tidak terlalu mengikuti data yang ada.

7 Kesimpulan

Model yang dipilih untuk memodelkan data Total Barang di Pelabuhan Tanjung Priok adalah ARIMA $(2, 1, 1)$ dan memiliki MAPE sebesar $\approx 12\%$. Meskipun MAPE model bernilai kecil, namum model tidak mengikuti data, sehingga model ini kurang cocok digunakan untuk memprediksi data. Akan lebih baik jika dipilih model lain.

Jurnal Praktikum 4

Lakukan pemodelan deret waktu menggunakan data yang terdapat pada sheet Tanjung Perak dan juga semua tugas jurnal praktikum yang sebelumnya !

8 Daftar Pustaka

Cryer, J., & Chan, K. (2011). Time series analysis. New York: Springer. Guidolin, M., & Pedio, M. (2018). Essentials of Time Series for Financial Applications. Academic Press. Klugman, S. A., Panjer, H. H., & Willmot, G. E. (2019). In Loss models: From data to decisions (pp. 203–218). essay, John Wiley & Sons, Inc. Wei, W. W. S. (1990). Time series analysis: Univariate and multivariate methods. Redwood City, Calif: Addison-Wesley Pub.

Lampiran

A *Backshift Operator*

Pada analisis deret waktu ada suatu notasi yang sering digunakan untuk mempermudah merumuskan persamaan model yakni **Backshift Operator**. Perumusannya sangat sederhana yakni sebagai berikut :

$$BY_t = Y_{t-1}$$

Secara sederhana, B berlaku sebagai operator. Sehingga pada model AR(p) dapat ditulis seperti berikut :

$$\phi(B)Y_t = \varepsilon_t \tag{A.1}$$

dengan $\phi(B) = (1 - \phi_1 B - \cdots - \phi_p B^p)$.
Sedangkan untuk model MA(q) dapat ditulis sebagai berikut :

$$Y_t = \theta(B)\varepsilon_t \tag{A.2}$$

dengan $\theta(B) = (1 - \theta_1 B - \cdots - \theta_q B^q)$.
Secara umum untuk model ARIMA(p, d, q) dapat ditulis dengan persamaan berikut:

$$\phi(B)(1 - B)^d Y_t = \theta(B)\varepsilon_t \tag{A.3}$$

B Uji ADF

Didefinisikan suatu persamaan deret waktu yang mengikuti model AR(p) dengan *drift* sebagai berikut:

$$y_t = \mu + \sum_{i=1}^{p} \phi_i y_{t-i} + \varepsilon_t \tag{B.1}$$

Jika diasumsikan $p = 1$ maka, uji Augmented Dicky Fuller akan sama dengan uji Dicky Fuller. Sehingga uji hipotesis yang dilakukan adalah sebagai berikut

$$H_0 : \phi_1 = \cdots = \phi_k = 1$$
$$H_1 : \phi_1 \text{ hingga } \phi_k \text{ tidak sama dengan } 1$$

Jika H_0 ditolak, maka data dapat dikatakan stasioner (karena tidak ada efek dari *unit root*). Untuk nilai statistik hitung, pembaca disarankan untuk membaca *Probability & Statistics for Engineers & Scientists* karangan Walpole, Mayers, Ye Subbab 12.6 terkhusus pada bagian *Partial F-Tests on Subsets of Coefficients*. (Hal ini tidak dibahas lebih lanjut karna akan dipelajari pada kuliah AK3182 Model Linear).

C *Error Metrics*

Salah satu cara untuk menentukan model yang paling baik untuk memodelkan data adalah dengan menggunakan *error metric*. *Error metric* adalah cara untuk mengkuantifikasi ketepatan hasil prakiraan model terhadap data asli. Nilai *error metric* yang kecil menandakan model yang digunakan baik dalam memodelkan data. Pada modul ini, *error metric* yang akan dibahas adalah **MAE**, **MAPE**, **MSE**, dan **RMSE**. Pada bagian ini, akan digunakan beberapa notasi sebagai berikut:

1. n: ukuran data
2. Y_t: data ke-t
3. $\hat{Y}_t$: data prakiraan model

C.1 MAE

MAE (*Mean Absolute Error*) menghitung rataan dari nilai mutlak dari selisih nilai prakira model dan data asli pada suatu waktu tertentu. Nilai MAE diperoleh dengan persamaan:

$$MAE = \frac{\sum_{t=1}^{n} |\hat{Y}_t - Y_t|}{n} \tag{C.1}$$

Kelemahan dari parameter MAE adalah ketidakmampuan untuk menentukan apakah *error* dari model besar atau kecil karena *error* yang kecil dari nilai data yang besar akan menghasilkan nilai MAE yang besar. MAE juga sulit mendeteksi *error* yang besar dengan frekuensi sedikit, sehingga pengambilan keputusan yang hanya berdasarkan MAE sulit dilakukan. Lebih jauh, MAE tidak dapat mendeteksi apakah prakiraan dari model lebih rendah atau lebih besar dari nilai data yang sebenarnya. Pengembangan dari MAE adalah MAPE.

C.2 MAPE

MAPE (*Mean Absolute Percentage Error*) menghitung rataan dari persentase nilai mutlak dari selisih nilai prakira model dan data asli terhadap data asli pada suatu waktu tertentu. Nilai MAPE diperoleh dengan persamaan:

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \frac{|\hat{Y}_t - Y_t|}{Y_t} \times 100\% \tag{C.2}$$

MAPE memberikan gambaran keakuratan model yang lebih baik daripada MAE dengan membandingkan *error* model terhadap data asli. MAPE tidak cocok digunakan ketika terdapat nilai 0 atau nilai ekstrim pada data. Data dengan nilai 0 yang terlalu banyak akan menyebabkan nilai MAPE yang terlalu besar sehingga tidak baik untuk menguji keakuratan data. Selain itu, seperti MAE, MAPE kurang baik dalam mendeteksi pencilan.

C.3 MSE

MSE (*Mean Squared Error*) menghitung rataan dari kuadrat selisih nilai prakira model dan data asli pada suatu waktu tertentu. Nilai MSE diperoleh dengan persamaan:

$$MSE = \frac{\sum_{t=1}^{n} (\hat{Y}_t - Y_t)^2}{n} \tag{C.3}$$

Penggunaan kuadrat pada MSE menyebabkan data pencilan akan lebih berpengaruh pada nilai MSE. MSE memiliki kelemahan yang sama dengan MAE, yaitu sulit menentukan apakah *error* dari data besar atau kecil berdasarkan nilai MSE, namun tidak terpengaruh nilai data 0 yang merupakan kelemahan MAPE. Selain itu, penambahan kuadrat pada persamaan MSE menyebabkan satuan data MSE berbeda dengan satuan data asli.

C.4 RMSE

RMSE (*Root Mean Squared Error*) adalah akar dari MSE. Persamaan RMSE didefinisikan sebagai berikut:

$$RMSE = \sqrt{MSE} \tag{C.4}$$

RMSE menghilangkan efek kuadrat dari MSE sehingga satuan dari RMSE menjadi sama dengan satuan data. RMSE dapat dibandingkan dengan MAE untuk mengetahui efek pencilan jika terdapat pencilan pada data.

D *Maximum Likelihood Estimator*

Salah satu metode penaksiran parameter dari suatu distribusi yang sudah diasumsikan terlebih dahulu adalah metode *Maximum Likelihood Estimator* (MLE). Ide dari teknik ini adalah memaksimumkan fungsi *likelihood*, yaitu fungsi yang menyatakan kebolehjadian (*likelihood*) sampel acak terobservasi.

Misalkan sampel acak $\mathbf{y} = (y_1, y_2, \cdots, y_n)$ berasal dari suatu distribusi tertentu dengan parameter dari fungsi peluang tersebut adalah $\beta = (\beta_1, \beta_2, \cdots, \beta_n)^T$, yaitu $f(\mathbf{y}; \beta)$. Maka dapat diperoleh fungsi peluang bersama dari sampel acak tersebut adalah:

$$f_n(\mathbf{y}, \beta) = \prod_{k=1}^{n} f(y_k; \beta) \tag{D.1}$$

Dari sampel acak yang diperoleh, metode ini akan mencari nilai parameter yang mungkin sedemikian sehingga fungsi *likelihood* berikut bernilai maksimum. Fungsi *likelihood* didefinisikan sebagai berikut

$$\mathcal{L}(\beta; \mathbf{y}) = \prod_{k=1}^{n} f(\beta; y_k) \tag{D.2}$$

Nilai β yang memaksimumkan $\mathcal{L}$, notasikan $\widehat{\beta}$, menjadi hasil estimasi dari teknik ini.

Jika $\mathcal{L}$ dapat diturunkan, maka uji turunan dapat dilakukan untuk mencari nilai maksimum. Kita juga dapat bekerja dengan fungsi *log-likelihood*,

$$\ell(\beta; \mathbf{y}) = \ln \mathcal{L}(\beta; y). \tag{D.3}$$

Turunan pertama fungsi *log-likelihood* relatif lebih mudah dicari dibanding fungsi $\mathcal{L}$.

Estimator yang diperoleh dari metode estimasi *likelihood* maksimum akan bersifat konsisten, efisien, takbias secara asimtotik, dan invarian.

Pada modul ini, fungsi `arima` yang Anda gunakan akan melakukan estimasi parameter dengan teknik estimasi *likelihood* maksimum jika Anda menambahkan argumen `method = 'ML'`. Untuk mempelajari lebih lanjut mengenai estimasi parameter ARMA(p, q) dengan metode MLE, silakan baca pada situs berikut

—- Selesai —-