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 3

## Tujuan:

1. Mengetahui perbedaan ARIMA dan *Seasonal* ARIMA
2. Menganalisis data real deret waktu dengan efek musiman dan memodelkannya,

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

```
# Untuk membaca data dengan format csv
library(readr)

# Untuk membersihkan data
library(tidyr)

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

# Untuk melihat signifikansi koefisien dari parameter
library(lmtest)

# Untuk memprediksi data dari model
library(forecast)
```

Sedangkan data yang akan diolah adalah data *Sharing Bike in Washington D.C.* yang diperoleh dari Kaggle dengan modifikasi.

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 $\beta$ 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 $\sigma_\varepsilon^2$ 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

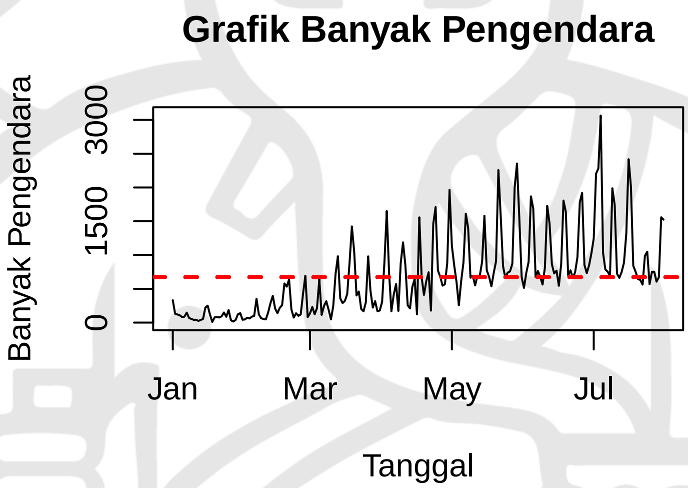
1. Identifikasi Model
2. Pembuatan Model (*Model Fitting*)
3. Uji Diagnostik Model

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, dan juga stastik deskriptif dari data

```
# Memanggil Data
data<- read_csv("day1.csv",
    col_types = cols(dteday = col_date(format = "%m/%d/%Y")))

# Membuat grafik data
plot(data$dteday,data$casual, type = 'l',
     ylab = "Banyak Pengendara", xlab = "Tanggal", main = "Grafik Banyak Pengendara")
# Membuat garis rataan
abline(h=mean(data$casual),lwd=2,
       lty = 2, col ='red')
```

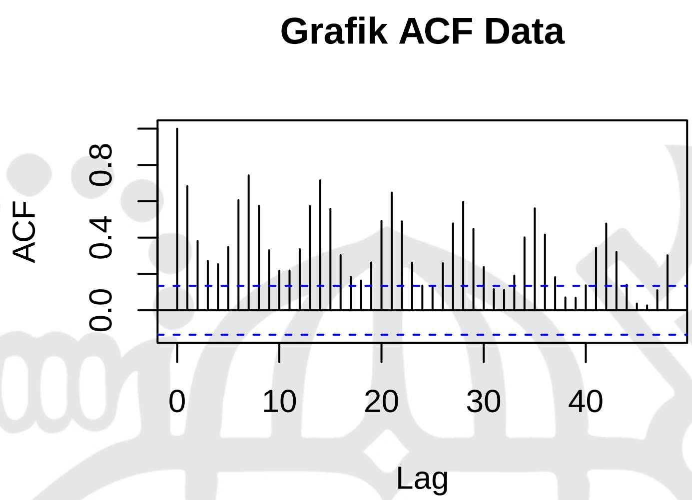



```
# Statistik Deskriptif
summary(data$casual)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##      9.0    178.5    623.0    672.6    865.0   3065.0
```

Dapat dilihat secara umum terdapat trend naik dari bulan Januari hingga Juli. Dapat pula dilihat seperti ada pola musiman pada data. Untuk membantu melihat pola musiman pada data, dapat dilihat melalui plot ACF data.

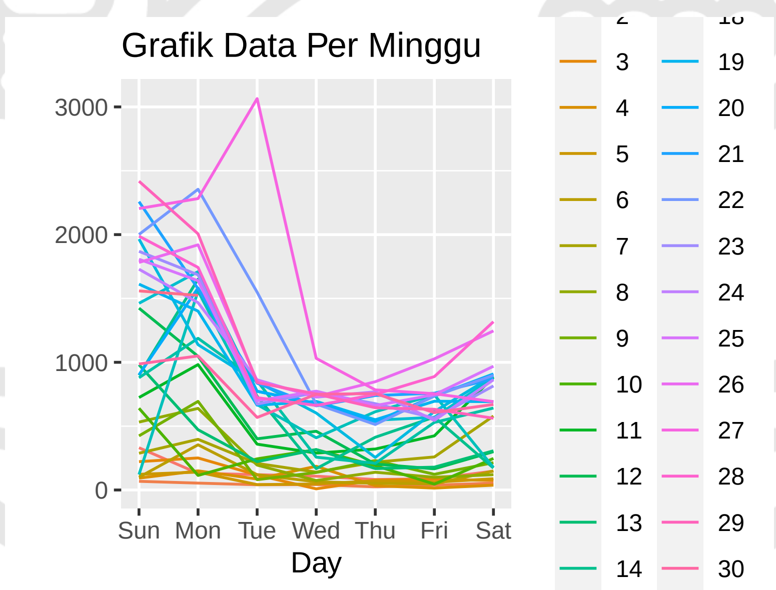
```
# Plot ACF
acf(data$casual, main ='Grafik ACF Data',lag.max = 48)
```



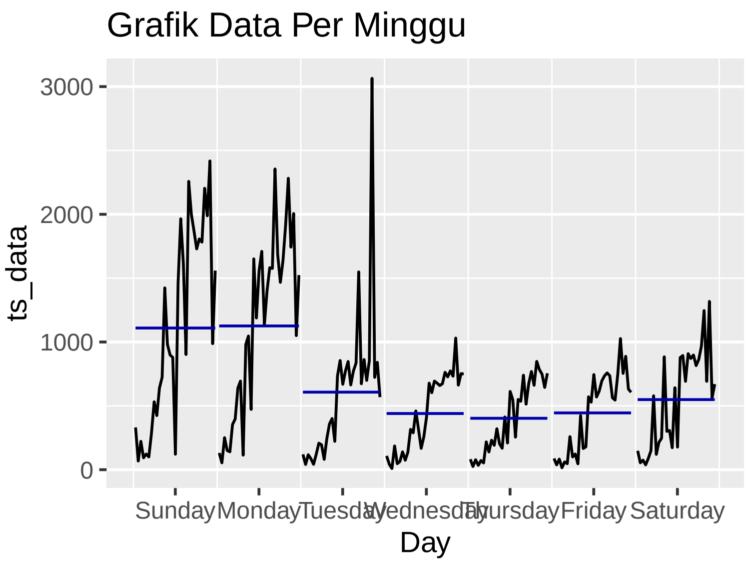

Pada grafik ACF dapat dilihat bahwa pada tiap lag kelipatan 7 cenderung lebih tinggi dari yang lain. Sehingga akan diasumsikan bahwa data memiliki pola seasonal dengan $s = 7$. Apabila terdapat data yang sulit diindentifikasi periode musimnya, dapat dilakukan *spectral analysis* yang terdapat pada lampiran A.

## 1.1 Lebih Jauh Mengenai Musiman

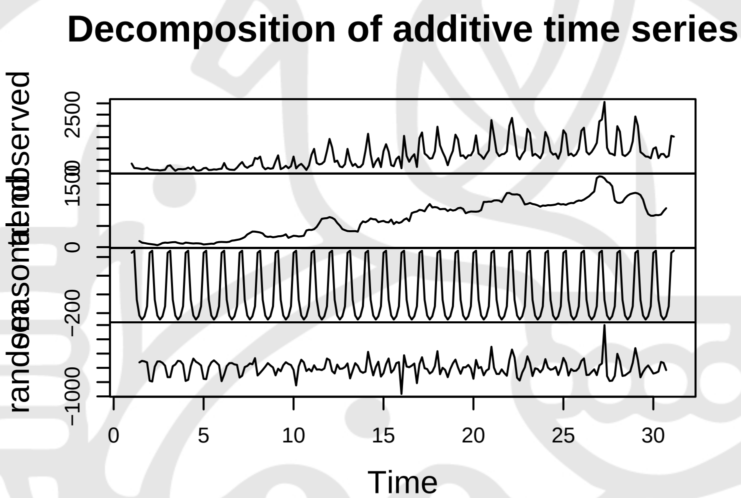
```
# Membuat data menjadi time series
ts_data <- ts(data$casual, frequency = 7)

#Plot data secara seasonal
ggseasonplot(ts_data, season.labels = 'Weeks', main = 'Grafik Data Per Minggu')
```




```
#Plot per subseries, misalnya perminggu
ggsubseriesplot(ts_data, main ='Grafik Data Per Minggu')
```




```
# Plot data yang telah didekomposisi
plot(decompose(ts_data))
```




Dari fungsi `ggsubseriesplot` dapat dilihat lebih jelas bahwa memang terdapat pola mingguan pada data. Sedangkan dari fungsi `decompose`dapat dilihat lebih jelas unsur-unsur dari masing-masing **trend** dan musiman.

## 2 Mempersiapkan Data

Untuk membagi data menjadi data **train** dan **validation** dilakukan dengan cara yang sedikit berbeda dengan yang sebelumnya. Data akan dibagi berdasarkan banyak minggunya. Pada data terdapat 212 observasi yang berarti terdapat sekitar 30 minggu. Sehingga jumlah observasi yang akan menjadi data **train** sebanyak $0.8 \times 30$ minggu $=$ 24 minggu $=$ 168 observasi dan sisa datanya akan menjadi data **validation**

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

```r
# dibutuhkan untuk menghasilkan data prediksi
casual <- ts(data$casual,frequency = 7)

# data yang akan digunakan untuk membuat model
casual_train <- ts(data$casual[1:168],frequency = 7)

# data yang akan digunakan untuk memvalidasi model
casual_validation <-ts(data$casual[169:212],frequency = 7)
```

**Jurnal Praktikum 1**

*Buatlah suatu base model yang sesuai!*

# 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 $\alpha = 0.05$.

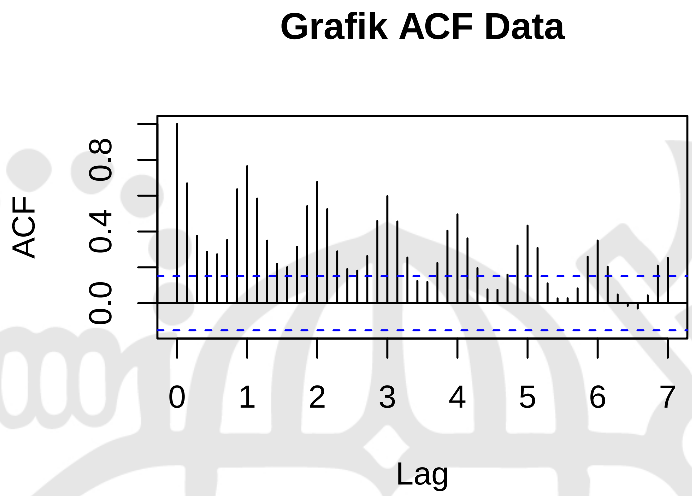
Akan dilakukan Uji ADF pada data

```r
adf.test(casual_train)
```

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

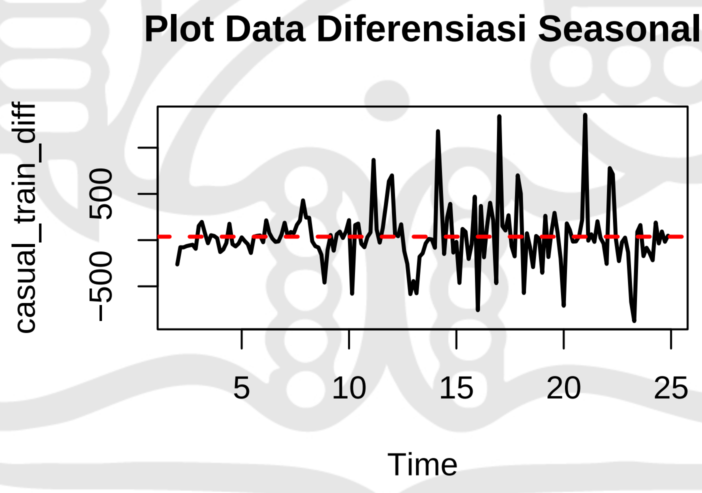
```
##
##  Augmented Dickey-Fuller Test
##
## data:  casual_train
## Dickey-Fuller = -4.7585, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```r
acf(casual_train, main = 'Grafik ACF Data',lag.max = 49)
```




Dapat dilihat bahwa nilai *p-value* $< \alpha$ sehingga dapat disimpulkan bahwa data sudah stasioner dalam rataan. Sehingga akan dilakukan diferensiasi musiman.

```
casual_train_diff <- diff(casual_train, lag = 7)
plot(casual_train_diff,
     lwd = 2,
     main = 'Plot Data Diferensiasi Seasonal')
abline(h=mean(casual_train_diff),
       lwd=2,lty = 2,
       col ='red')
```



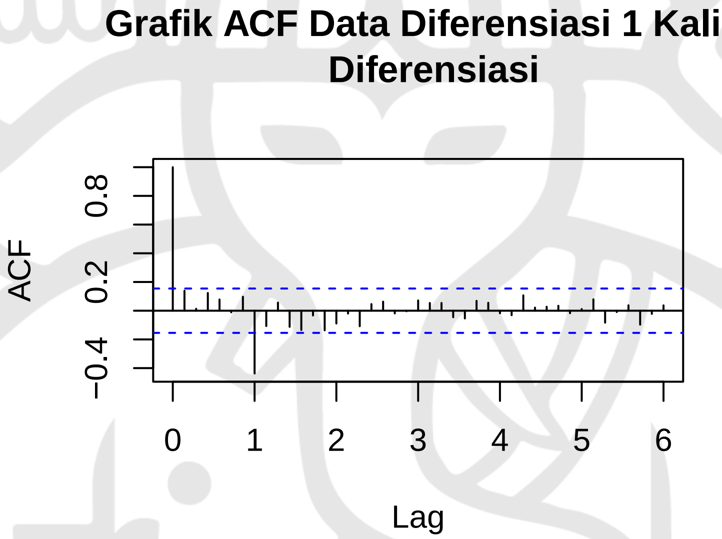

Dapat dilihat bahwa data yang sudah didiferensiasi musiman jauh lebih stasioner dibandingkan dengan yang belum didiferensiasi. Untuk lebih pasti akan dilakukan uji ADF dan akan dilihat pula grafik ACFnya

```
adf.test(casual_train_diff, k = 1)
```

```
## Warning in adf.test(casual_train_diff, k = 1): p-value smaller than printed
## p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  casual_train_diff
## Dickey-Fuller = -8.2423, Lag order = 1, p-value = 0.01
## alternative hypothesis: stationary
```

```
acf(casual_train_diff,
    main ='Grafik ACF Data Diferensiasi 1 Kali
    Diferensiasi ',
    lag.max = 42)
```



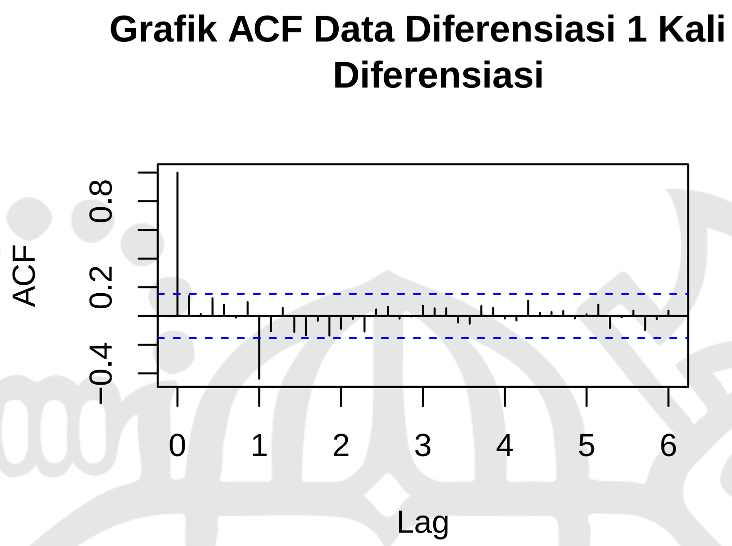

Dapat dilihat bahwa p-value $< \alpha$ sehingga dapat disimpulkan bahwa model sudah stasioner.

**Jurnal Praktikum 2**

*Diferensiasi mana yang perlu dilakukan terlebih dahulu ? Apakah diferensiasi musiman atau diferensiasi rataan ?*

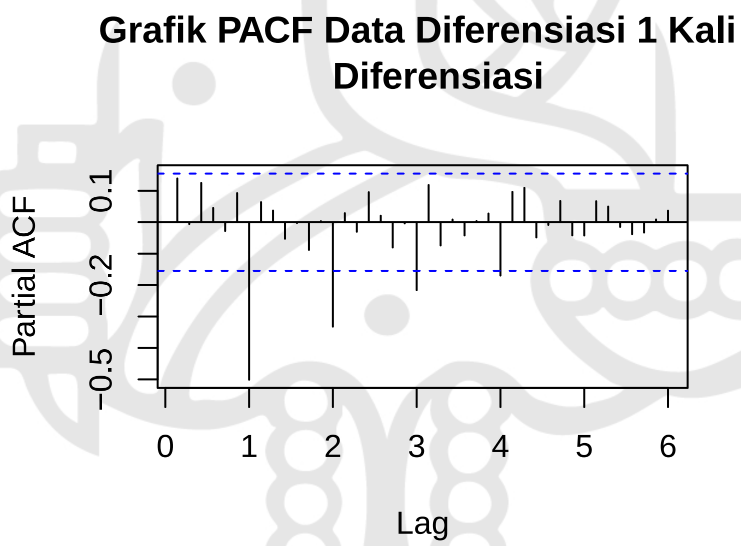
Setelah data menjadi stasioner, berikutnya akan diidentifikasi model yang cocok pada data dengan melihat grafik ACF dan PACF data

```
acf(casual_train_diff,
main ='Grafik ACF Data Diferensiasi 1 Kali
    Diferensiasi ', lag.max = 42)
```




Dapat dilihat bahwa grafik ACF cut off pada lag pertama musiman. Namun dapat dilihat juga bahwa plot ACF memiliki *tails off*. Hal ini akan disimpulkan kemudian.

```
pacf(casual_train_diff,
main ='Grafik PACF Data Diferensiasi 1 Kali
    Diferensiasi ', lag.max = 42)
```




Pada plot PACF dapat dilihat bahwa grafik *cuts off* musiman hingga lag ke-4.

Dari grafik-grafik tersebut dapat disimpulkan model-model yang akan dicoba adalah sebagai berikut :

1. $SARIMA(0,0,0) \times (4,1,1)_7$
2. $SARIMA(1,0,1) \times (4,1,1)_7$

Untuk interpretasi model dapat diperoleh model yang berbeda dengan yang di atas asalkan alasan yang diberikan masih bisa diterima. Model yang baik adalah model yang memiliki sifat parsimoni, yakni model yang dapat menjelaskan data dengan baik dengan parameter yang sedikit. Sehingga untuk itu perlu dilakukan analisis lebih lanjut mengenai ketiga kandidat model tersebut.

# 4 Penaksiran Parameter

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(casual_train, order = c(0,0,0),
               seasonal = list(order = c(4,1,1), period = 7),
               method = 'ML')
mod_1
```

```
##
## Call:
## arima(x = casual_train, order = c(0, 0, 0), seasonal = list(order = c(4, 1,
##     1), period = 7), method = "ML")
##
## Coefficients:
##         sar1    sar2    sar3    sar4     sma1
##       0.3169  0.1781  0.2921  0.2114  -0.9864
## s.e.  0.0825  0.0761  0.0793  0.0833   0.0314
##
## sigma^2 estimated as 67966:  log likelihood = -1130.66,  aic = 2273.33
```

```
mod_2 <- arima(casual_train, order = c(1,0,1),
               seasonal = list(order = c(4,1,1),
                               period = 7),
               method = 'ML')
mod_2
```

```
##
## Call:
## arima(x = casual_train, order = c(1, 0, 1), seasonal = list(order = c(4, 1,
##     1), period = 7), method = "ML")
##
## Coefficients:
##          ar1      ma1     sar1     sar2     sar3     sar4     sma1
##       0.9781  -0.8690  -0.3287  -0.2976  -0.1166  -0.0862  -0.4397
## s.e.  0.0323   0.1094   0.2756   0.2167   0.1716   0.1123   0.2722
##
## sigma^2 estimated as 64598:  log likelihood = -1122.78,  aic = 2261.55
```

```
mod_auto <- auto.arima(casual_train, max.p=4,max.q=4, max.P = 4, max.Q = 4,
                       seasonal =TRUE, stationary = FALSE, allowdrift = FALSE)
mod_auto
```

```
## Series: casual_train
## ARIMA(4,0,3)(0,1,3)[7]
##
## Coefficients:
```

```
##           ar1     ar2     ar3     ar4     ma1      ma2      ma3     sma1
##       -0.3894  0.7174  0.5837  0.0533  0.6983  -0.6655  -0.7786  -0.7143
## s.e.   0.1688  0.1381  0.1633  0.1310  0.1634   0.1281   0.1921   0.1036
##          sma2    sma3
##       -0.0897  0.1378
## s.e.   0.0940  0.0843
##
## sigma^2 = 61403:  log likelihood = -1114.58
## AIC=2251.15   AICc=2252.93   BIC=2285.05
```

(Catatan : Pada bagian ini metode CSS tidak dapat digunakan, sehingga digunakan metode ML)
Dapat dilihat bahwa model dengan AIC terkecil dimiliki oleh model $SARIMA(4,0,3)\times(0,1,3)_7$. Namun, dapat dilihat pula pada model $SARIMA(0,0,0)\times(4,1,1)_7$ memiliki AIC yang tidak jauh berbeda dengan AIC pada model $SARIMA(1,0,1)\times(4,1,1)_7$. sehingga akan dipilih model $SARIMA(0,0,0)\times(4,1,1)_7$ untuk dianalisis lebih lanjut.

Berikutnya akan dilakukan uji signifikansi dari model

# 5 Signifikansi dari Koefisien Parameter dan Pembuatan Model

Akan diperiksa siginifikansi parameter model

```
coeftest(mod_1)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## sar1  0.316942   0.082450   3.8440 0.0001210 ***
## sar2  0.178136   0.076085   2.3413 0.0192182 *
## sar3  0.292121   0.079271   3.6851 0.0002286 ***
## sar4  0.211437   0.083283   2.5388 0.0111244 *
## sma1 -0.986412   0.031359 -31.4554 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Dapat dilihat bahwa semua parameter signifikan. Sehingga dapat disimpulkan bahwa model yang dipilih dapat digunakan untuk memodelkan data.
Penulisan persamaan dari model adalah sebagai berikut :

$$\Phi_4(B)(1-B^7)Y_t = \Theta_1(B)\varepsilon_t$$
$$\vdots$$

(Diserahkan kepada pembaca sebagai latihan)

$$Y_t = \text{... (Diserahkan kepada pembaca sebagai latihan)} \tag{5.1}$$

Berikutnya akan dilihat performa model seperti berikut

```
accuracy(mod_1)
```

```
##                    ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set 48.7357 255.214 163.0103 -19.06527 52.99769 0.5902329 0.1728057
```

Dapat dilihat bahwa MAPE dari model sebesar $\approx 53\%$, namun ME dari model sebesar 49 yang relatif kecil dibandingkan dengan rataan data. Sehingga dapat diambil kesimpulan bahwa model ini cocok digunakan untuk memodelkan data yang ada.

**Jurnal Praktikum 3**

*Bandingkanlah performa model SARIMA yang dipilih dengan base model yang sebelumnya telah dibuat!*

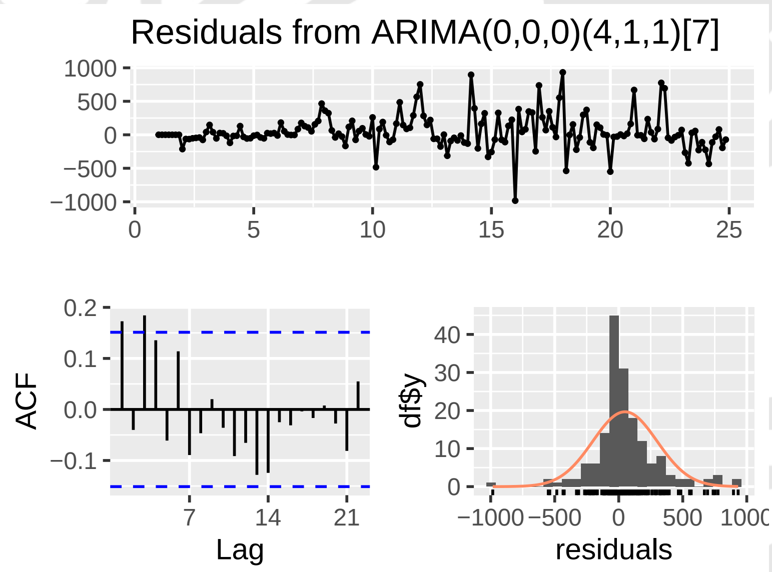
## 6 Uji Diagnostik

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

1. Berdistribusi Normal dengan rataan 0 dan variansi $\sigma_\varepsilon^2$ 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)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,0)(4,1,1)[7]
## Q* = 27.704, df = 9, p-value = 0.001069
```

```
##
## Model df: 5.   Total lags used: 14
```

Perhatikan bahwa p-value $< \alpha$ sehingga dapat disimpulkan bahwa data tidak saling bebas. Hal ini diperkuat dari plot ACF yang signifikan pada lag ke 1 dan 3. 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 4**

*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?*

## 7 Forecasting

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

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

```
## [1] 0.2940466
```

```
me_validation <- mean((validation$mean) -actual)
me_validation
```

```
## [1] 116.4079
```

Dapat dilihat bahwa MAPE dari model sebesar 29.4% yang lebih kecil daripada MAPE yang diperoleh sebelumnya sedangkan ME-nya sedikit lebih besar dibandingkan dengan yang sebelumnya namun tetap lebih kecil dibandingkan statistik deskriptif data. Sehingga model yang dipilih dapat digunakan untuk memprediksi data.

.

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.

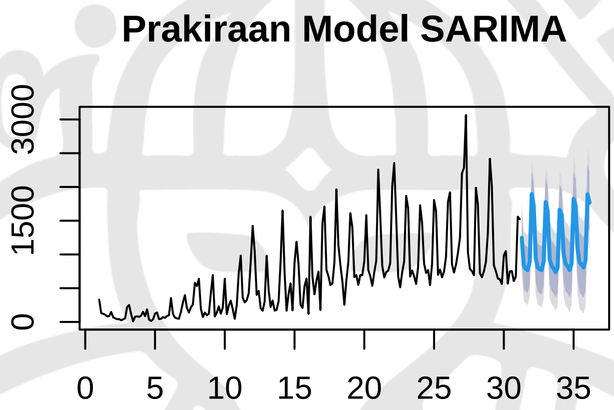
```
fc <- forecast(casual,
               model=mod_1,
               h=35)
summary(fc)
```

```
##
## Forecast method: ARIMA(0,0,0)(4,1,1)[7]
##
## Model Information:
```

```
## Series: object
## ARIMA(0,0,0)(4,1,1)[7]
##
## Coefficients:
##         sar1    sar2    sar3    sar4     sma1
##       0.3169  0.1781  0.2921  0.2114  -0.9864
## s.e.  0.0000  0.0000  0.0000  0.0000   0.0000
##
## sigma^2 = 67966:  log likelihood = -1489.27
## AIC=2980.54   AICc=2980.56   BIC=2983.86
##
## Error measures:
##                    ME     RMSE      MAE       MPE    MAPE      MASE      ACF1
## Training set 15.85028 328.6412 197.1135 -19.44382 48.7727 0.8159332 0.3862319
##
## Forecasts:
##          Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## 31.28571      1247.0291  909.8851 1584.173  731.4118 1762.646
## 31.42857       832.8130  495.6690 1169.957  317.1957 1348.430
## 31.57143       786.8936  449.7496 1124.038  271.2763 1302.511
## 31.71429       768.3337  431.1897 1105.478  252.7164 1283.951
## 31.85714       892.8899  555.7459 1230.034  377.2726 1408.507
## 32.00000      1894.8442 1557.8347 2231.854 1379.4327 2410.256
## 32.14286      1720.1048 1383.0953 2057.114 1204.6933 2235.516
## 32.28571       952.8962  596.0216 1309.771  407.1037 1498.689
## 32.42857       806.9743  450.0998 1163.849  261.1818 1352.767
## 32.57143       778.7828  421.9083 1135.657  232.9903 1324.575
## 32.71429       768.7022  411.8277 1125.577  222.9097 1314.495
## 32.85714       894.2881  537.4136 1251.163  348.4956 1440.081
## 33.00000      1776.0745 1419.4204 2132.729 1230.6191 2321.530
## 33.14286      1643.0358 1286.3817 1999.690 1097.5804 2188.491
## 33.28571       925.8331  553.2381 1298.428  355.9983 1495.668
## 33.42857       831.9820  459.3871 1204.577  262.1472 1401.817
## 33.57143       792.3295  419.7346 1164.924  222.4947 1362.164
## 33.71429       736.0135  363.4186 1108.608  166.1787 1305.848
## 33.85714       806.1387  433.5438 1178.734  236.3039 1375.974
## 34.00000      1662.7841 1290.4991 2035.069 1093.4233 2232.145
## 34.14286      1589.8741 1217.5892 1962.159 1020.5134 2159.235
## 34.28571      1005.4961  598.6391 1412.353  383.2621 1627.730
## 34.42857       859.0737  452.2167 1265.931  236.8397 1481.308
## 34.57143       827.9645  421.1075 1234.821  205.7304 1450.199
## 34.71429       767.6424  360.7854 1174.499  145.4083 1389.876
## 34.85714       866.3954  459.5384 1273.252  244.1614 1488.629
## 35.00000      1824.7693 1418.3730 2231.166 1203.2398 2446.299
## 35.14286      1716.8035 1310.4072 2123.200 1095.2740 2338.333
## 35.28571      1083.5733  636.0690 1531.078  399.1746 1767.972
## 35.42857       882.0767  434.5724 1329.581  197.6780 1566.475
```

```
## 35.57143       846.0460  398.5418 1293.550  161.6473 1530.445
## 35.71429       806.2749  358.7706 1253.779  121.8761 1490.674
## 35.85714       917.3264  469.8221 1364.831  232.9276 1601.725
## 36.00000      1892.2429 1445.3850 2339.101 1208.8327 2575.653
## 36.14286      1766.5132 1319.6552 2213.371 1083.1029 2449.923
```

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




Dapat dilihat bahwa hasil **forecast** untuk 5 periode mengikuti data yang ada meskipun pada data terjadi beberapa lonjakkan yang besar di puncak periodenya. Sehingga dapat disimpoulkan bahwa model dapat digunakan untuk memprediksi data.

# 8 Kesimpulan

Model yang dipilih untuk memodelkan data banyak orang yang melakukan **bike sharing** di Washington D.C. pada periode Januari 2018-Juli 2018 adalah $SARIMA(0,0,0) \times (4,1,1)_7$ dan memiliki ME sebesar $\approx$ 49% sehingga model ini cocok digunakan untuk memprediksi data.

**Jurnal Praktikum 5**

*Lakukan pemodelan musiman untuk data berikut*

*TEKAN UNTUK MENGUNDUH DATA*

# 9 Daftar Pustaka

Cryer, J., & Chan, K. (2011). Time series analysis. New York: Springer.
Wei, W. W. S. (1990). Time series analysis: Univariate and multivariate methods. Redwood City, Calif: Addison-Wesley Pub.

# Lampiran

## A *Spectral Analysis*

*Spectral analysis* adalah suatu metode yang digunakan untuk menemukan suatu pola “tersembunyi” yang terdapat pada data. Idenya adalah dengan mencocokkan data dengan suatu fungsi *cosinus* sebagai berikut:

$$y_t := R\cos\left(2\pi ft + \Phi\right) \tag{A.1}$$

dengan $y_t$ adalah data ke-$t$, $R$ adalah amplitudo, $f$ adalah frekuensi dan $\Phi$ adalah fasa.
Penaksiran parameter $R$ dan $\Phi$ dengan menggunakan identitas trigonometri sehingga persamaan (A.1) dapat ditulis sebagai berikut

$$R\cos\left(2\pi ft + \Phi\right) = A\cos\left(2\pi ft\right) + B\sin\left(2\pi ft\right) \tag{A.2}$$

Sehingga dapat diperoleh:

$$R = \sqrt{A^2 + B^2} \qquad \text{dan} \qquad \Phi = \tan^{-1}\left(-\frac{B}{A}\right) \tag{A.3}$$

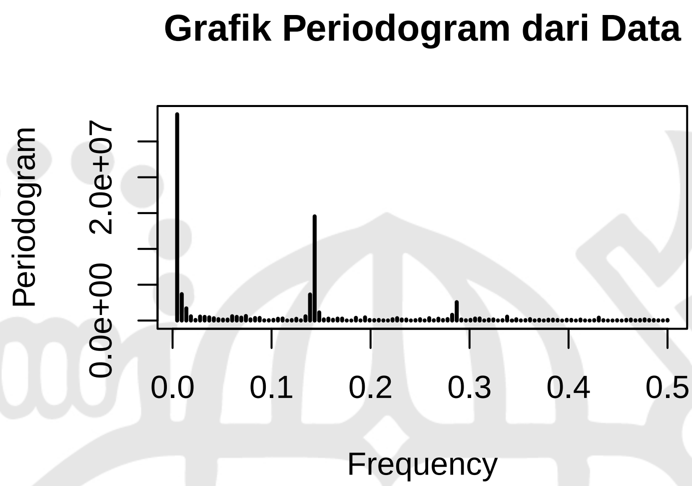
Nilai $A$ dan $B$ dapat ditaksir dengan menggunakan *Ordinary Least Square Regression.*
Nilai $y_t$ dapat pula ditulis dalam bentuk kombinasi linear dari $m$ fungsi kosinus dengan nilai amplitudo, frekuensi dan fasa tertentu seperti persamaan berikut:

$$y_t := A_0 + \sum_{j=1}^{m}\left[A_j\cos\left(2\pi f_j t\right) + B_j\sin\left(2\pi f_j t\right)\right] \tag{A.4}$$

Penaksiran nilai parameter dari persamaan (A.4) dapat menggunakan *Ordinary Least Square Regression* namun dapat pula dilakukan dengan menggunakan algoritma *Fast Fourier Transform* (FFT). Mengenai algoritma FFT tidak akan dibahas lebih lanjut pada modul ini.
Setelah melakukan transformasi Fourier pada data, akan dilihat nilai periodogram dari data. Nilai periodogram ini yang akan digunakan untuk mengindentifikasi frekuensi dominan dari data deret waktu. Sehingga akan dicari frekuensi dengan nilai periodogram yang terbesar.

Akan diberikan contoh untuk menentukan periode suatu data dengan menggunakan algoritma FFT.

```r
# Library yang digunakan adalah TSA
library(TSA)

# Masih menggunakan Data Casual Bike pada contoh sebelumnya
data <- read_csv("day1.csv",
    col_types = cols(dteday = col_date(format = "%m/%d/%Y")))
ts_data <- ts(data$casual)
y_t <- periodogram(ts_data, main = 'Grafik Periodogram dari Data')
```




```
freq <- y_t$freq
periodogram <- y_t$spec
```

Dapat dilihat dari grafik bahwa terdapat 2 nilai periodogram yang signifikan. Terdapat pula suatu hubungan antara Periode, $P$, dengan frekeunsi yaitu:

$$P := \frac{1}{f} \tag{A.5}$$

Sehingga setelah memperoleh frekuensi yang dominan, yaitu frekuensi dengan nilai periodogram yang terbesar, dapat diperoleh periode yang dominan pula.

```
# Memperoleh frekuensi yang dominan
dominan <- freq[which(periodogram %in% sort(periodogram, decreasing = T)[1:2])]
periode <- 1/dominan
periode
```

```
## [1] 216.000000   6.967742
```

Diperoleh Periode sebesar 216 dan 6.97. Dapat dilihat bahwa ukuran data sebesar 212, sehingga lebih masuk akal jika dipilih periode 6.97 sebagai periode yang lebih dominan. Karena data berupa bilangan bulat, maka 6.97 akan dibulatkan menjadi 7.

Perhatikan bahwa periode ini juga diperoleh dengan menggunakan grafik ACF pada bagian sebelumnya.

**CATATAN :** Bagian ini menjelaskan *spectral analysis* secara praktis dan ada beberapa konsep yang dilewati agar tidak memperumit pembahasan. Apabila ingin memahami lebih jauh mengenai *spectral analysis*, dianjurkan pada Daftar Pustaka pertama pada Bab 13 *Introduction to Spectral Analysis* dan 14 *Estimating the Spectrum*.

—- Selesai —-